

28/40/44-Pin 8-Bit Flash Microcontrollers with XLP Technology

High-Performance RISC CPU:

- C Compiler Optimized Architecture
- Only 49 Instructions
- Up to 14 Kbytes Self-Write/Read Flash Program Memory Addressing
- · Up to 256 Bytes Data Memory Addressing
- · Operating Speed:
 - DC 20 MHz clock input @ 3.6V
 - DC 16 MHz clock input @ 1.8V
 - DC 200 ns instruction cycle
- Interrupt Capability with Automatic Context Saving
- 16-Level Deep Hardware Stack with Optional Overflow/Underflow Reset
- Direct, Indirect and Relative Addressing modes:
 - Two full 16-bit File Select Registers (FSRs)
 - FSRs can read program and data memory

Memory

- Up to 14 Kbytes Self-Write/Read Flash Program Memory Addressing
- · Up to 256 Bytes Data Memory Addressing
- High-Endurance Flash Data Memory (HEF)
 - 128B of nonvolatile data storage
 - 100K erase/write cycles

Flexible Oscillator Structure:

- 16 MHz Internal Oscillator Block:
 - Accuracy to ± 3%, typical
 - Software selectable frequency range from 16 MHz to 31.25 kHz
- · 31 kHz Low-Power Internal Oscillator
- · Three External Clock modes up to 20 MHz
- Two-Speed Oscillator Start-up
- Low-Power RTC Implementation via LPT1OSC

Special Microcontroller Features:

- Operating Voltage Range:
- 1.8V-3.6V
- Self-Programmable under Software Control
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Low-Power Brown-Out Reset (LPBOR)

- Extended Watchdog Timer (WDT)
- In-Circuit Serial Programming[™] (ICSP[™]) via Two Pins
- · In-Circuit Debug (ICD) via Two Pins
- Enhanced Low-Voltage Programming (LVP)
- Programmable Code Protection
- · Power-Saving Sleep mode

eXtreme Low-Power (XLP) Features (PIC16LF1904/6/7):

- Sleep Current:
 - 30 nA @ 1.8V, typical
- Watchdog Timer Current:
 - 300 nA @ 1.8V, typical
- · Secondary Oscillator:
 - 500 nA @ 32 kHz, 1.8V, typical

Analog Features:

- Analog-to-Digital Converter (ADC):
 - 10-bit resolution, up to 14 channels
 - Conversion available during Sleep
 - Dedicated ADC RC oscillator
 - Fixed Voltage Reference (FVR) as channel
- Integrated Temperature Indicator
- · Voltage Reference module:
 - Fixed Voltage Reference (FVR) with 1.024V and 2.048V output levels

Peripheral Highlights:

- Up to 36 I/O Pins and 1 Input-only Pin:
 - High current 25 mA sink/source
 - Individually programmable weak pull-ups
 - Individually programmable interrupt-onchange (IOC) pins
- Integrated LCD Controller:
 - At least 19 segment pins and as many as 116 total segments
 - Variable clock input
 - Contrast control
- Internal voltage reference selections
- Timer0: 8-Bit Timer/Counter with 8-Bit Programmable Prescaler

- Enhanced Timer1:
 - 16-bit timer/counter with prescaler
 - External Gate Input mode
 - Dedicated low-power 32 kHz oscillator driver
- Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART):
 - RS-232, RS-485 and LIN compatible
 - Auto-Baud Detect
 - Auto-wake-up on start

PIC16LF190X Family Types

	~	~		ash						LCD			
Device	Data Sheet Inde)	Program Memor Flash (words)	Data SRAM (bytes)	High-Endurance Fl (bytes)	ا/O's ⁽²⁾	10-bit ADC (ch)	Timers (8/16-bit)	EUSART	Common Pins	Segment Pins	Total Segments	Debug ⁽¹⁾	ХГР
PIC16LF1902	(1)	2048	128	128	25	11	1/1		4	19	72 ⁽³⁾	Н	Y
PIC16LF1903	(1)	4096	256	128	25	11	1/1		4	19	72 ⁽³⁾	Н	Y
PIC16LF1904	(2)	4096	256	128	36	14	1/1	1	4	29	116	I/H	Y
PIC16LF1906	(2)	8192	512	128	25	11	1/1	1	4	19	72 ⁽³⁾	I/H	Y
PIC16LF1907	(2)	8192	512	128	36	14	1/1	1	4	29	116	I/H	Y

Note 1: Debugging Methods: (I) – Integrated on Chip; (H) – using Debug Header; (E) – using Emulation Header.
2: One pin is input-only.

3: COM3 and SEG15 share a pin, so the total segments are limited to 72 for 28-pin devices.

Data Sheet Index: (Unshaded devices are described in this document.)

- 1: DS40001455 PIC16LF1902/3 Data Sheet, 28-Pin Flash, 8-bit Microcontrollers.
- 2: DS40001569 PIC16LF1904/6/7 Data Sheet, 28/40/44-Pin Flash, 8-bit Microcontrollers.

Note: For other small form-factor package availability and marking information, please visit http://www.microchip.com/packaging or contact your local sales office.





FIGURE 3: 40-F	PIN PDIP PACKAGE DIA	AGRAM	FOR PIC16LF1904/7	
FIGURE 3: 40-F	VPP/MCLR/RE3 1 SEG12/AN0/RA0 2 SEG7/AN1/RA1 3 COM2/AN2/RA2 4 G15/VREF+/AN3/RA3 5 SEG4/T0CKI/RA4 6 SEG2/AN4/RA5 7 SEG21/AN5/RE0 8 SEG2/AN4/RA5 10 VDD 11 VSS 12 SEG2/AN6/RE1 9 SEG22/AN6/RE1 9 SEG22/AN6/RE1 10 VDD 11 VSS 12 SEG2/CLKIN/RA7 13 SEG1/CLKOUT/RA6 14 T1CKI/T10SO/RC0 15 T10SI/RC1 16 SEG3/RC2 17 SEG6/RC3 18 COM3/RD0 19 SEG27/RD1 20	AGRAM	40 RB7/ICSPDAT/ICDDAT/SEG13 39 RB6/ICSPCLK/ICDCLK/SEG14 38 RB5/AN13/COM1 37 RB4/AN11/COM0 36 RB3/AN9/SEG26/VLCD3 35 RB2/AN8/SEG25/VLCD2 34 RB1/AN10/SEG24/VLCD1 33 RB0/AN12/INT/SEG0 34 RD7/SEG20 35 RD7/SEG20 36 RD5/SEG18 37 RD4/SEG17 38 RC5/SEG10 39 RC5/SEG10 30 RC4/T1G/SEG11 31 RS2 32 RD4/SEG17	



FIGURE 5:	40-PIN UQFN (5X5) I	PACKAGE DIAGRAM FOR PIC16	6LF1904/7
		RC6/TX/CK/SEG9 RC5/SEG10 RC4/T16/SEG11 RD3/SEG16 RD2/SEG28 RD1/SEG27 RD0/COM3 RC3/SEG6 RC3/SEG6 RC2/SEG3 RC1/T1OS1	
	SEG8/DT/RX/RC7	1 337 337 337 337 338 333 337 337 337 337	
	SEG17/RD4 SEG18/RD5 SEG19/RD6 SEG20/RD7 VSS VDD SEG0/INT/AN12/RB0 VLCD1/SEG24/AN10/RB1 VLCD2/SEG25/AN8/RB2	2 30 F 3 29 F 4 28 F 5 27 V 6 PIC16LF1904/7 26 V 7 25 F 8 24 F 9 23 F 10 22 F 10 22 F 10 22 F	RC0/T1OSO/T1CKI RA6/CLKOUT/SEG1 RA7/CLKIN/SEG2 /ss /pd RE2/AN7/SEG23 RE1/AN6/SEG22 RE0/AN5/SEG21 RA5/AN4/SEG5 RA4/T0CKI/SEG4
		VLCD3/SEG26/AN9/RB3 COM0/AN11/RB4 COM11/RB4 COM1AN13/RB5 SEG14/ICDCLK/ICSPCLK/RB6 SEG13/ICDDAT/ICSPDAT/RB7 VPP/MCLR/RE3 SEG12/AN0/RA0 SEG12/AN1/RA1 COM2/AN1/RA1 COM2/AN1/RA1	

IADLL		20/4	0/44-11									
0/	28-Pin PDIP/ SOIC/SSOP	28-Pin UQFN	40-Pin PDIP	44-Pin TQFP	40-Pin UQFN	٩/Þ	Timers	EUSART	ГСD	Interrupt	Pull-up	Basic
RA0	2	27	2	19	17	AN0		_	SEG12	—		—
RA1	3	28	3	20	18	AN1	_	_	SEG7	_	_	_
RA2	4	1	4	21	19	AN2		_	COM2			_
RA3	5	2	5	22	20	AN3/ VREF+	—	_	SEG15/ COM3 ⁽²⁾	-	-	—
RA4	6	3	6	23	21	_	T0CKI	_	SEG4	_		
RA5	7	4	7	24	22	AN4		_	SEG5			_
RA6	10	7	14	31	29	_	_	_	SEG1	_		CLKOUT
RA7	9	6	13	30	28	_		_	SEG2	_	_	CI KIN
RB0	21	18	33	8	8	AN12	—	—	SEG0	INT/	Y	_
DB1	22	10	34	0	0	AN10				100	v	
IXD1	22	19	54	5	5	ANTO			SEG24	100	'	
RB2	23	20	35	10	10	AN8	—	—	VLCD2/ SEG25	IOC	Y	—
RB3	24	21	36	11	11	AN9	—	—	VLCD3/ SEG26	IOC	Y	—
RB4	25	22	37	14	12	AN11	—	_	COM0	IOC	Y	—
RB5	26	23	38	15	13	AN13	—		COM1	IOC	Y	
RB6	27	24	39	16	14	_	—	_	SEG14	IOC	Y	ICSPCLK/ ICDCLK
RB7	28	25	40	17	15		—		SEG13	IOC	Y	ICSPDAT/ ICDDAT
RC0	11	8	15	32	30	_	T1OSO/ T1CKI	_	—	-	—	—
RC1	12	9	16	35	31		T10SI	_		_	—	
RC2	13	10	17	36	32	_	—	-	SEG3	—	_	
RC3	14	11	18	37	33	_	—		SEG6	—	—	
RC4	15	12	23	42	38	_	T1G	_	SEG11	_	_	_
RC5	16	13	24	43	39	_	_	_	SEG10	_	_	_
RC6	17	14	25	44	40	_	_	TX/CK	SEG9	_		_
RC7	18	15	26	1	1	_	_	RX/DT	SEG8	_	_	_
RD0	_	_	19	38	34	_	_	_	COM3 ⁽³⁾	_	_	_
RD1	_	_	20	39	35	_	_	_	SEG27	_	_	_
RD2		_	21	40	36	_	_	_	SEG28	_		_
RD3	_	_	22	41	37	_	_	_	SEG16	_	_	_
RD4	_	_	27	2	2	_	_	_	SEG17	_	_	
RD5		_	28	3	3	_		_	SEG18			_
RD6			29	4	4	_		_	SEG19			_
RD7		_	30	5	5	_	_	_	SEG20	_	_	_
RE0			8	25	23	AN5 ⁽⁴⁾	_	_	SEG21	_		_
RF1	_	_	9	26	24	AN6 ⁽⁴⁾		_	SEG22	_	_	_
RF2			10	27	25	AN7(4)			SEG23			
	4	26	10	10	10	7.1.17			01020		V(1)	
KE3	1	20 17	11.00	10	7.00	_	_	_	_	_	1	
VDD	20	F 46	10.04	1,20	7,20	_	_	_	_	_		Vice
V SS	0,19	5,10	12,31	12.12	0, 27	_		_				V 55
NC	_	_	_	33 34	_	_	_	_	_	_	_	VUU

TABLE 1: 28/40/44-PIN ALLOCATION TABLE (PIC16LF1904/6/7)

Note 1: Weak pull-up always enabled when MCLR is enabled, otherwise the pull-up is under user control.

2: 28-pin only pin location (PIC16LF1906). Location different on 40/44-pin device.

3: 40/44-pin only pin location (PIC16LF1904/7). Location different on 28-pin device.

4: ADC channel is reserved on the PIC16LF1906 28-pin devices.

Table of Contents

Device Overview	10
Enhanced Mid-range CPU	15
Memory Organization	16
Device Configuration	38
Resets	43
Oscillator Module	51
Interrupts	60
Power-Down Mode (Sleep)	71
Watchdog Timer	73
Flash Program Memory Control	77
I/O Ports	93
Interrupt-On-Change	109
Fixed Voltage Reference (FVR)	112
Temperature Indicator Module	114
Analog-to-Digital Converter (ADC) Module	115
Timer0 Module	128
Timer1 Module with Gate Control	130
Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	142
Liquid Crystal Display (LCD) Driver Module	171
In-Circuit Serial Programming™ (ICSP™)	205
Instruction Set Summary	207
Electrical Specifications	221
DC and AC Characteristics Graphs and Charts	238
Development Support	239
Packaging Information	243
ndix A: Data Sheet Revision History	261
Aicrochip WebSite	262
omer Change Notification Service	262
omer Support	262
uct Identification System	263
	Device Overview

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@microchip.com. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

Microchip's Worldwide Website; http://www.microchip.com

• Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our website at www.microchip.com to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

The PIC16LF1904/6/7 are described within this data sheet. They are available in 28, 40 and 44-pin packages. Figure 1-1 shows a block diagram of the PIC16LF1904/6/7 devices. Table 1-2 shows the pinout descriptions.

Reference Table 1-1 for peripherals available per device.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC16LF1906	PIC16LF1904/7
ADC		٠	•
EUSART		•	•
Fixed Voltage Reference	•	•	
LCD	٠	•	
Temperature Indicator	•	•	
Timers			
	Timer0	•	•
	Timer1	٠	•



PIC16LF1904/6/7 PINOUT DESCRIPTION **TABLE 1-2:**

Name	Function	Input Type	Output Type	Description
RA0/AN0/SEG12	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN		A/D Channel 0 input.
	SEG12		AN	LCD Analog output.
RA1/AN1/SEG7	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN		A/D Channel 1 input.
	SEG7	—	AN	LCD Analog output.
RA2/AN2/COM2	RA2	TTL	CMOS	General purpose I/O.
	AN2	AN		A/D Channel 2 input.
	COM2	_	AN	LCD Analog output.
RA3/AN3/VREF+/COM3 ⁽²⁾ /	RA3	TTL	CMOS	General purpose I/O.
SEG15	AN3	AN	—	A/D Channel 3 input.
	VREF+	AN	—	A/D Voltage Reference input.
	COM3		AN	LCD Analog output.
	SEG15	_	AN	LCD Analog output.
RA4/T0CKI/SEG4	RA4	TTL	CMOS	General purpose I/O.
	TOCKI	ST		Timer0 clock input.
	SEG4		AN	LCD Analog output.
RA5/AN4/SEG5	RA5	TTL	CMOS	General purpose I/O.
	AN4	AN		A/D Channel 4 input.
	SEG5		AN	LCD Analog output.
RA6/CLKOUT/SEG1	RA6	TTL	CMOS	General purpose I/O.
	CLKOUT		CMOS	Fosc/4 output.
	SEG1	_	AN	LCD Analog output.
RA7/CLKIN/SEG2	RA7	TTL	CMOS	General purpose I/O.
	CLKIN	CMOS	—	External clock input (EC mode).
	SEG2	—	AN	LCD Analog output.
RB0/AN12/INT/SEG0	RB0	TTL	CMOS	General purpose I/O.
	AN12	AN	—	A/D Channel 12 input.
	INT	ST	_	External interrupt.
	SEG0	—	AN	LCD Analog output.
RB1 ⁽¹⁾ /AN10/SEG24/VLCD1	RB1	TTL	CMOS	General purpose I/O.
	AN10	AN	_	A/D Channel 10 input.
	SEG24	—	AN	LCD Analog output.
	VLCD1	AN	—	LCD analog input.
RB2 ⁽¹⁾ /AN8/SEG25/VLCD2	RB2	TTL	CMOS	General purpose I/O.
	AN8	AN	—	A/D Channel 8 input.
	SEG25		AN	LCD Analog output.
	VLCD2	AN	—	LCD analog input.

 Legend:
 AN = Analog input or output
 CMOS = CMOS compatible input or output
 OD = Open-Drain

 TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I²C = Schmitt Trigger input with I²C

levels

XTAL = Crystal H١

Note1:These pins have interrupt-on-change functionality.2:PIC16LF1906/7 only.

Name	Function	Input Type	Output Type	Description
RB3 ⁽¹⁾ /AN9/SEG26/VLCD3	RB3	TTL	CMOS	General purpose I/O.
	AN9	AN	_	A/D Channel 9 input.
	· SEG26	—	AN	LCD Analog output.
	VLCD3	AN	—	LCD analog input.
RB4 ⁽¹⁾ /AN11/COM0	RB4	TTL	CMOS	General purpose I/O.
	AN11	AN	_	A/D Channel 11 input.
	COM0	—	AN	LCD Analog output.
RB5 ⁽¹⁾ /AN13/COM1	RB5	TTL	CMOS	General purpose I/O.
	AN13	AN	—	A/D Channel 13 input.
	COM1		AN	LCD Analog output.
RB6 ⁽¹⁾ /ICSPCLK/ICDCLK/	RB6	TTL	CMOS	General purpose I/O.
SEG14	ICSPCLK	ST	_	Serial Programming Clock.
	ICDCLK	ST	_	In-Circuit Debug Clock.
	SEG14	—	AN	LCD Analog output.
RB7 ⁽¹⁾ /ICSPDAT/ICDDAT/	RB7	TTL	CMOS	General purpose I/O.
SEG13	ICSPDAT	ST	_	Serial Programming Clock.
	ICDDAT	ST	CMOS	In-Circuit Data I/O.
	SEG13	—	AN	LCD Analog output.
RC0/T1OSO/T1CKI	RC0	TTL	CMOS	General purpose I/O.
	T10SO	XTAL	XTAL	Timer1 oscillator connection.
	T1CKI	ST	_	Timer1 clock input.
RC1/T1OSI	RC1	TTL	CMOS	General purpose I/O.
	T10SI	XTAL	XTAL	Timer1 oscillator connection.
RC2/SEG3	RC2	TTL	CMOS	General purpose I/O.
	SEG3	_	AN	LCD Analog output.
RC3/SEG6	RC3	TTL	CMOS	General purpose I/O.
	SEG6	_	AN	LCD Analog output.
RC4/T1G/SEG11	RC4	TTL	CMOS	General purpose I/O.
	T1G	XTAL	XTAL	Timer1 oscillator connection.
	SEG11	_	AN	LCD Analog output.
RC5/SEG10	RC5	TTL	CMOS	General purpose I/O.
	SEG10		AN	LCD Analog output.
RC6/TX/CK/SEG9	RC6	TTL	CMOS	General purpose I/O.
	TX		CMOS	USART asynchronous transmit.
	СК	ST	CMOS	USART synchronous clock.
	SEG9	_	AN	LCD Analog output.
RC7/RX/DT/SEG8	RC7	TTL	CMOS	General purpose I/O.
	RX	ST	_	USART asynchronous input.
	DT	ST	CMOS	USART synchronous data.
	SEG8	_	AN	LCD Analog output.
Legend: AN = Analog input or	output CMC	S= CM	DS compa	atible input or output OD = Open-Drain

PIC16LF1904/6/7 PINOUT DESCRIPTION (CONTINUED) **TABLE 1-2:**

XTAL = Crystal HV = High Voltage

TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C levels

Note 1: These pins have interrupt-on-change functionality.

2: PIC16LF1906/7 only.

TABLE 1-2: PIC16LF1904/6/7 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RD0 ⁽²⁾ /COM3	RD0	TTL	CMOS	General purpose I/O.
	COM3	—	AN	LCD Analog output.
RD1 ⁽²⁾ /SEG27	RD1	TTL	CMOS	General purpose I/O.
	SEG27	_	AN	LCD Analog output.
RD2 ⁽²⁾ /SEG28	RD2	TTL	CMOS	General purpose I/O.
	SEG28	_	AN	LCD Analog output.
RD3 ⁽²⁾ /SEG16	RD3	TTL	CMOS	General purpose I/O.
	SEG16	—	AN	LCD Analog output.
RD4 ⁽²⁾ /SEG17	RD4	TTL	CMOS	General purpose I/O.
	SEG17	_	AN	LCD Analog output.
RD5 ⁽²⁾ /SEG18	RD5	TTL	CMOS	General purpose I/O.
	SEG18	_	AN	LCD Analog output.
RD6 ⁽²⁾ /SEG19	RD6	TTL	CMOS	General purpose I/O.
	SEG19	—	AN	LCD Analog output.
RD7 ⁽²⁾ /SEG20	RD7	TTL	CMOS	General purpose I/O.
	SEG20	_	AN	LCD Analog output.
RE0 ⁽²⁾ /AN5/SEG21	RE0	TTL	CMOS	General purpose I/O.
	AN5	AN	_	A/D Channel 5 input.
	SEG21	—	AN	LCD Analog output.
RE1 ⁽²⁾ /AN6/SEG22	RE1	TTL	CMOS	General purpose I/O.
	AN6	AN	—	A/D Channel 6 input.
	SEG22	—	AN	LCD Analog output.
RE2 ⁽²⁾ /AN7/SEG23	RE2	TTL	CMOS	General purpose I/O.
	AN7	AN	—	A/D Channel 7 input.
	SEG23	—	AN	LCD Analog output.
RE3/MCLR/VPP	RE3	TTL	CMOS	General purpose I/O.
	MCLR	ST	—	Master Clear with internal pull-up.
	Vpp	HV		Programming voltage.
Vdd	Vdd	Power		Positive supply.
Vss	Vss	Power	—	Ground reference.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output

TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C

HV = High Voltage XTAL = Crystal

OD = Open-Drain

= Schmitt Trigger input with I²C levels

Note 1: These pins have interrupt-on-change functionality.

2: PIC16LF1906/7 only.

DS40001569D-page 14

2.0 ENHANCED MID-RANGE CPU

This family of devices contain an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16 levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

- · Automatic Interrupt Context Saving
- 16-level Stack with Overflow and Underflow
- File Select Registers
- Instruction Set

2.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 7.5 "Automatic Context Saving"**, for more information.

2.2 16-Level Stack with Overflow and Underflow

These devices have an external stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and if enabled, will cause a software Reset. See **Section 3.4 "Stack"** for more details.

2.3 File Select Registers

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers and program memory, which allows one Data Pointer for all memory. When an FSR points to program memory, there is one additional instruction cycle in instructions using INDF to allow the data to be fetched. General purpose memory can now also be addressed linearly, providing the ability to access contiguous data larger than 80 bytes. There are also new instructions to support the FSRs. See **Section 3.5 "Indirect Addressing"** for more details.

2.4 Instruction Set

There are 49 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 21.0 "Instruction Set Summary"** for more details.





3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- · Program Memory
 - Configuration Words
 - Device ID
 - User ID
 - Flash Program Memory
- Data Memory
 - Core Registers
 - Special Function Registers
 - General Purpose RAM
 - Common RAM

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range ⁽¹⁾		
PIC16LF1904	4,096	0FFFh	0F80h-0FFFh		
PIC16LF1906/7	8,192	1FFFh	1F80h-1FFFh		

Note 1: High-endurance Flash applies to low byte of each address in the range.

3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16LF1904/6/7 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1, and 3-2).



3.1.1 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.

3.1.1.1 RETLW Instruction

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.

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_INI	DEX
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.

3.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH directive will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

constants				
RETLW	DATA0	;Index0	data	
RETLW	DATA1	;Index1	data	
RETLW	DATA2			
RETLW	DATA3			
my_functi	on			
; LO	IS OF CODE.			
MOVLW	LOW cons	tants		
MOVWF	FSR1L			
MOVLW	HIGH con	stants		
MOVWF	FSR1H			
MOVIW	0[FSR1]			
; THE PROG	RAM MEMORY	IS IN W		

3.2 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-3):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See Section 3.5 "Indirect Addressing" for more information.

Data memory uses a 12-bit address. The upper seven bits of the address define the Bank Address and the lower five bits select the registers/RAM in that bank.

3.2.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-4.

Addresses	BANKx
x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

3.2.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- the arithmetic status of the ALU
- · the Reset status

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, CLRF STATUS will clear the upper three bits and set the Z bit. This leaves the STATUS register as '000u uluu' (where u = unchanged).

It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (Refer to Section 21.0 "Instruction Set Summary").

Note:	The C	and DC	bits	opera	te as Borrow a	ind
	Digit	Borrow	out	bits,	respectively,	in
	subtra	action.				

REGISTER 3-1: STATUS: STATUS REGISTER

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
	_	_	TO	PD	Z	DC ⁽¹⁾	C ⁽¹⁾
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	q = Value depends on condition

bit 7-5	Unimplemented: Read as '0'
bit 4	TO: Time-out bit
	1 = After power-up, CLRWDT instruction or SLEEP instruction
	0 = A WDT time-out occurred
bit 3	PD: Power-Down bit
	1 = After power-up or by the CLRWDT instruction
	0 = By execution of the SLEEP instruction
bit 2	Z: Zero bit
	1 = The result of an arithmetic or logic operation is zero
	0 = The result of an arithmetic or logic operation is not zero
bit 1	DC: Digit Carry/Digit Borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) ⁽¹⁾
	1 = A carry-out from the 4th low-order bit of the result occurred
	0 = No carry-out from the 4th low-order bit of the result
bit 0	C: Carry/Borrow bit ⁽¹⁾ (ADDWF, ADDLW, SUBLW, SUBWF instructions) ⁽¹⁾
	1 = A carry-out from the Most Significant bit of the result occurred
	0 = No carry-out from the Most Significant bit of the result occurred
Note 1:	For Borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the
	second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high-order or low-order

bit of the source register.

3.2.2 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh). The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

3.2.3 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh).

3.2.3.1 Linear Access to GPR

The general purpose RAM can be accessed in a nonbanked method via the FSRs. This can simplify access to large memory structures. See **Section 3.5.2 "Linear Data Memory"** for more information.

3.2.4 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

FIGURE 3-3: BANKED MEMORY PARTITIONING



3.2.5 DEVICE MEMORY MAPS

The memory maps for PIC16LF1904/6/7 are as shown in Table 3-3.

TABLE 3-3: PIC16LF1904/6/7 MEMORY MAP

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7
000h	Core Registers (Table 3-2)	080h	Core Registers (Table 3-2)	100h	Core Registers (Table 3-2)	180h	Core Registers (Table 3-2)	200h	Core Registers (Table 3-2)	280h	Core Registers (Table 3-2)	300h	Core Registers (Table 3-2)	380h	Core Registers (Table 3-2)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	_	28Ch	_	30Ch	_	38Ch	_
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	_	38Dh	_
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	_	20Eh	—	28Eh	—	30Eh	_	38Eh	_
00Fh	PORTD ⁽¹⁾	08Fh	TRISD ⁽¹⁾	10Fh	LATD ⁽¹⁾	18Fh	_	20Fh	_	28Fh	_	30Fh	-	38Fh	_
010h	PORTE	090h	TRISE ⁽¹⁾	110h	LATE ⁽¹⁾	190h	ANSELE ⁽¹⁾	210h	WPUE	290h	—	310h	-	390h	_
011h	PIR1	091h	PIE1	111h	—	191h	PMADRL	211h	—	291h	—	311h	_	391h	_
012h	PIR2	092h	PIE2	112h	—	192h	PMADRH	212h	—	292h	—	312h	_	392h	—
013h	—	093h	_	113h	—	193h	PMDATL	213h	_	293h	_	313h	_	393h	—
014h	—	094h	_	114h	—	194h	PMDATH	214h	_	294h	_	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	—	195h	PMCON1	215h	—	295h	_	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	_	296h		316h	_	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	—	217h	_	297h		317h	_	397h	—
018h	T1CON	098h	_	118h	_	198h	—	218h	—	298h	_	318h	—	398h	_
019h	T1GCON	099h	OSCCON	119h	_	199h	RCREG	219h	_	299h	_	319h	_	399h	_
01Ah	_	09Ah	OSCSTAT	11Ah	_	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	_	39Ah	_
01Bh	—	09Bh	ADRESL	11Bh	_	19Bh	SPBRG	21Bh	_	29Bh	_	31Bh	_	39Bh	_
01Ch	—	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	_	39Ch	_
01Dh	—	09Dh	ADCON0	11Dh	—	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	_	39Dh	_
01Eh	—	09Eh	ADCON1	11Eh	_	19Eh	TXSTA	21Eh	—	29Eh		31Eh	—	39Eh	—
01Fh	—	09Fh	—	11Fh	_	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	_	39Fh	_
020h	General Purpose	0A0h	General Purpose Register 80 Bytes	120h 13Fh 140h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes ⁽²⁾	220h	General Purpose Register 80 Bytes ⁽²⁾	2A0h	General Purpose Register 80 Bytes ⁽²⁾	320h	General Purpose Register 32 Bytes ⁽²⁾ Unimplemented Read as '0'	3A0h	Unimplemented Read as '0'
06Fh	96 Bytes	0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h	,	0F0h	Accesses 70h – 7Fh	170h	Accesses 70h – 7Fh	1F0h	Accesses 70h – 7Fh	270h	Accesses 70h – 7Fh	2F0h	Accesses 70h – 7Fh	370h	Accesses 70h – 7Fh	3F0h	Accesses 70h – 7Fh
						1660		21511		2661		31 51		SEEU	

Legend: = Unimplemented data memory locations, read as '0'.

Note 1: PIC16LF1904/7 only.

2: PIC16LF1906/7 only.

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		
400h	Core Registers (Table 3-2)	480h	Core Registers (Table 3-2)	500h	Core Registers (Table 3-2)	580h	Core Registers (Table 3-2)	600h	Core Registers (Table 3-2)	680h	Core Registers (Table 3-2)	700h	Core Registers (Table 3-2)		
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh			
40Ch	Unimplemented Read as '0'	48Ch	Unimplemented Read as '0'	50Ch	Unimplemented Read as '0'	58Ch	Unimplemented Read as '0'	60Ch	Unimplemented Read as '0'	68Ch	Unimplemented Read as '0'	70Ch	Unimplemented Read as '0'		
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh			
470h	Common RAM	4F0h	Common RAM	570h	Common RAM	5F0h	Common RAM	670h	Common RAM	6F0h	Common RAM	770h	Common RAM		
47Fh	(Accesses 70h – 7Fh)	4FFh	(Accesses 70h – 7Fh)	57Fh	(Accesses 70h – 7Fh)	5EEh	(Accesses 70h – 7Fh)	67Fh	(Accesses 70h – 7Fh)	6FFh	(Accesses 70h – 7Fh)	77Fh	(Accesses 70h – 7Fh)		
4/111				1		Jorrin] 0/111		01111		,,,,,,,		1	
	BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23
800h	Core Registers	880h	Core Registers	900h	Core Registers	980h	Core Registers	A00h	Core Registers	A80h	Core Registers	B00h	Core Registers	B80h	Core Registers
80Ph	(Table 3-2)	88Dh	(Table 3-2)	OORh	(Table 3-2)	09Ph	(Table 3-2)	AORh	(Table 3-2)	AgDh	(Table 3-2)	POPh	(Table 3-2)	DOD	(Table 3-2)
80Ch		88Ch		90Ch		98Ch		A0Ch		A8Ch		BOCh		B8Ch	
	Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh		BEFh	
870h	Common RAM (Accesses	8F0h	Common RAM (Accesses	970h	Common RAM (Accesses	9F0h	Common RAM (Accesses	A70h	Common RAM (Accesses	AF0h	Common RAM (Accesses	B70h	Common RAM (Accesses	BF0h	Common RAM (Accesses
87Fh	7011 – 7FII)	8FFh	7011 – 7FII)	97Fh	7011 – 7FII)	9FFh	7011 – 711)	A7Fh	7011 – 7FII)	AFFh	7011 – 7FII)	B7Fh	7011 – 7FII)	BFFh	7011 – 7 FTI)
-														-	
	BANK 24		DANK 25		BANK 26		DANK 27		DANK 20		BANK 20		DVNK 30		

	BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30
C00h	Core Registers (Table 3-2)	C80h	Core Registers (Table 3-2)	D00h	Core Registers (Table 3-2)	D80h	Core Registers (Table 3-2)	E00h	Core Registers (Table 3-2)	E80h	Core Registers (Table 3-2)	F00h	Core Registers (Table 3-2)
C0Bh		C8Bh		D0Bh		D8Bh		E0Bh		E8Bh		F0Bh	
C0Ch		C8Ch		D0Ch		D8Ch		E0Ch		E8Ch		F0Ch	
	Unimplemented Read as '0'												
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh	
C70h C7Fh	Common RAM (Accesses 70h – 7Fh)	CF0h CFFh	Common RAM (Accesses 70h – 7Fh)	D70h D7Fh	Common RAM (Accesses 70h – 7Fh)	DF0h DFFh	Common RAM (Accesses 70h – 7Fh)	E70h E7Fh	Common RAM (Accesses 70h – 7Fh)	EF0h EFFh	Common RAM (Accesses 70h – 7Fh)	F70h F7Fh	Common RAM (Accesses 70h – 7Fh)



PIC16LF1904/6/7 MEMORY MAP (CONTINUED)

TABLE 3-3:

TABLE 3-3: PIC16LF1904/6/7 MEMORY MAP (CONTINUED)

BANK 15

Core Registers (Table 3-2)

Unimplemented Read as '0'

LCDCON

LCDPS

LCDREF

LCDCST LCDRL

LCDSE0

LCDSE1

LCDSE2(1)

LCDSE3

Unimplemented Read as '0'

LCDDATA0

LCDDATA1 LCDDATA2⁽¹⁾

LCDDATA3

LCDDATA4 LCDDATA5⁽¹⁾

LCDDATA6

LCDDATA7 LCDDATA8⁽¹⁾

LCDDATA9

LCDDATA10 LCDDATA11⁽¹⁾

LCDDATA12

—

LCDDATA15

LCDDATA18

LCDDATA21

Unimplemented Read as '0'

780h

78Bh 78Ch

790h

791h

792h

793h 794h

795h 796h 797h

798h

799h

79Ah

79Bh 79Ch

79Fh 7A0h

7A1h

7A2h 7A3h

7A4h

7A5h 7A6h

7A7h

7A8h 7A9h

7AAh

7ABh 7ACh

7ADh

7AEh 7AFh

7B0h 7B1h 7B2h

7B3h 7B4h 7B5h

7B6h 7B7h 7B8h

7EFh

	BANK 31
F80h	Core Registers (Table 3-2)
F8Bn	
F8Ch	Unimplemented Read as '0'
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	_
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH
FF0h	Common RAM (Accesses 70h – 7Fh)
FFFh	

00	٥r	٠h٠
Ley	61	ıu.

= Unimplemented data memory locations, read as '0'.

Note 1: PIC16LF1904/7 only.

3.2.6 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-4 can be addressed from any Bank.

TABLE 3-4:	CORE FUNCTION REGISTERS SUMMARY

Addr	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	0-31										
x00h or x80h	INDF0	Addressing (not a phys	this locatior ical register)	n uses conte	nts of FSR0H	/FSR0L to a	ddress data	memory		XXXX XXXX	uuuu uuuu
x01h or x81h	INDF1	Addressing (not a phys	this locatior ical register)	n uses conte	nts of FSR1H	/FSR1L to a	ddress data	memory		xxxx xxxx	uuuu uuuu
x02h or x82h	PCL	Program C	ounter (PC)	Least Signifi	cant Byte					0000 0000	0000 0000
x03h or x83h	STATUS	—	_	_	ТО	PD	Z	DC	С	1 1000	q quuu
x04h or x84h	FSR0L	Indirect Da	ta Memory A	ddress 0 Lo	w Pointer					0000 0000	uuuu uuuu
x05h or x85h	FSR0H	Indirect Da	ta Memory A	ddress 0 Hi	gh Pointer					0000 0000	0000 0000
x06h or x86h	FSR1L	Indirect Da	ta Memory A	ddress 1 Lo	w Pointer					0000 0000	uuuu uuuu
x07h or x87h	FSR1H	Indirect Da	ta Memory A	ddress 1 Hig	gh Pointer					0000 0000	0000 0000
x08h or x88h	BSR	_	_	_	BSR4	BSR3	BSR2	BSR1	BSR0	0 0000	0 0000
x09h or x89h	WREG	Working Re	egister							0000 0000	uuuu uuuu
x0Ahor x8Ah	PCLATH	_	Write Buffer	r for the upp	er 7 bits of the	e Program C	ounter			-000 0000	-000 0000
x0Bhor x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000

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

Addr	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
Ban	k 0										
00Ch	PORTA	PORTA Dat	ta Latch whe	en written: Po	ORTA pins wh	nen read				xxxx xxxx	uuuu uuuu
00Dh	PORTB	PORTB Dat	ta Latch whe	en written: P	ORTB pins w	hen read				xxxx xxxx	uuuu uuuu
00Eh	PORTC	PORTC Da	ta Latch whe	en written: P	ORTC pins w	hen read				xxxx xxxx	uuuu uuuu
00Fh	PORTD ⁽³⁾	PORTD Da	ta Latch whe	en written: P	ORTD pins w	hen read				xxxx xxxx	uuuu uuuu
010h	PORTE	—		—	—	RE3	RE2 ⁽²⁾	RE1 ⁽²⁾	RE0 ⁽²⁾	xxxx	uuuu
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF		—	—	TMR1IF	00000	00000
012h	PIR2	_	_	—	—	_	LCDIF	_	—	0 -0	0 -0
013h	—	Unimpleme	nted							_	_
014h	—	Unimpleme	nted							_	_
015h	TMR0	Timer0 Mod	dule Registe	r						xxxx xxxx	uuuu uuuu
016h	TMR1L	Holding Re	gister for the	Least Signi	ficant Byte of	the 16-bit TM	VR1 Registe	r		xxxx xxxx	uuuu uuuu
017h	TMR1H	Holding Re	gister for the	Most Signif	icant Byte of	the 16-bit TN	IR1 Register			xxxx xxxx	uuuu uuuu
018h	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR10N	0000 00-0	uuuu uu-u
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T <u>1GGO</u> / DONE	T1GVAL	T1GSS1	T1GSS0	0000 0x00	uuuu uxuu
01Ah to 01Fh	_	Unimpleme	nted							_	_
Ban	k 1	1									
08Ch	TRISA	PORTA Dat	ta Direction I	Register						1111 1111	1111 1111
08Dh	TRISB	PORTB Da	ta Direction	Register						1111 1111	1111 1111
08Eh	TRISC	PORTC Da	ta Direction	Register						1111 1111	1111 1111
08Fh	TRISD ⁽³⁾	PORTD Da	ta Direction	Register		(0)	(0)	(0)	(0)	1111 1111	1111 1111
090h	TRISE	—	—	—	—	(2)	TRISE2 ⁽³⁾	TRISE1 ⁽³⁾	TRISE0 ⁽³⁾	1111	1111
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	—	TMR1IE	00000	00000
092h	PIE2	—	_	_	—	—	LCDIE	_	—	0	0
093h	<u> </u>	Unimpleme	nted							_	_
094h	<u> </u>	Unimpleme	nted							_	_
095h	OPTION_REG	WPUEN	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	00-1 11qq	qq-q qquu
097h	WDTCON	—	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	01 0110	01 0110
098h	—	Unimpleme	nted			1		1	1	—	_
099h	OSCCON	—	IRCF3	IRCF2	IRCF1	IRCF0	—	SCS1	SCS0	-011 1-00	-011 1-00
09Ah	OSCSTAT	T10SCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS	0-q000	d-dd0d
09Bh	ADRESL	A/D Result	Register Lov	N						xxxx xxxx	uuuu uuuu
09Ch	ADRESH	A/D Result	Register Hig	h		1		1		XXXX XXXX	uuuu uuuu
09Dh	ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS2	ADCS1	ADCS0	—	—	ADPREF1	ADPREF0	0000	0000
09Fh	—	Unimpleme	nted							—	—

TABLE 3-5:	SPECIAL	FUNCTION	REGISTER	SUMMARY
				0011111/1111

 ${\bf x}$ = unknown, ${\bf u}$ = unchanged, ${\bf g}$ = value depends on condition, - = unimplemented, read as '0', ${\bf r}$ = reserved. Shaded locations are unimplemented, read as '0'. These registers can be addressed from any bank. Unimplemented, read as '1'. PIC16LF1904/7 only. Legend:

Note 1:

2:

						•					
Addr	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
Ban	k 2					•				-	
10Ch	LATA	PORTA Da	ta Latch							xxxx xxxx	uuuu uuuu
10Dh	LATB	PORTB Da	ta Latch							xxxx xxxx	uuuu uuuu
10Eh	LATC	PORTC Da	ita Latch							xxxx xxxx	uuuu uuuu
10Eh	LATD ⁽³⁾	PORTD Da	ta Latch							xxxx xxxx	uuuu uuuu
10Eh	LATE ⁽³⁾	—	—	—	_	_	LATE2	LATE1	LATE0	xxx	uuu
111h to 115h	_	Unimpleme	ented							_	_
116h	BORCON	SBOREN	BORFS	_	_	_	_	_	BORRDY	10q	uuu
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	_	_	ADFVR1	ADFVR0	0q0000	0q0000
118h to 11Fh	_	Unimpleme	ented							-	_
Ban	ik 3										
18Ch	ANSELA	_	_	ANSA5	_	ANSA3	ANSA2	ANSA1	ANSA0	1- 1111	11 1111
18Dh	ANSELB	_	_	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	11 1111	11 1111
18Eh	—	Unimpleme	ented							_	_
18Fh	—	Unimpleme	ented							_	_
190h	ANSELE ⁽³⁾	_	—	_	_	_	ANSE2	ANSE1	ANSE0	111	111
191h	PMADRL	Program M	emory Addre	ess Register	Low Byte					0000 0000	0000 0000
192h	PMADRH	(2)	Program M	emory Addre	ess Register I	High Byte				1000 0000	1000 0000
193h	PMDATL	Program M	emory Read	Data Regis	ter Low Byte					xxxx xxxx	uuuu uuuu
194h	PMDATH	_	_	Program M	lemory Read	Data Registe	r High Byte			xx xxxx	uu uuuu
195h	PMCON1	(2)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000
196h	PMCON2	Program M	emory Contr	ol Register	2					0000 0000	0000 0000
197h	—	Unimpleme	ented							_	_
198h	—	Unimpleme	ented							_	_
199h	RCREG	USART Re	ceive Data F	Register						0000 0000	0000 0000
19Ah	TXREG	USART Tra	ansmit Data I	Register						0000 0000	0000 0000
19Bh	SPBRG				BRG	<7:0>				0000 0000	0000 0000
19Ch	SPBRGH				BRG<	<15:8>				0000 0000	0000 0000
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
19Fh	BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	_	WUE	ABDEN	01-0 0-00	01-0 0-00
Ban	ik 4		•		•		•	•		·	
20Ch	_	Unimpleme	ented							_	_
20Dh	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111
20Eh	_	Unimpleme	ented							_	_
20Fh	_	Unimpleme	ented							_	_
210h	WPUE	—	_	_		WPUE3	_		_	1	1

SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) TABLE 3-5

Bank 5

211h to 21Fh

29Fh

28Ch

_ _

Bank 6

Dali	K U			
30Ch	_	Unimplemented	_	-
21Eb				
SIFI				
Legen	d: x = unkr Shaded	pwn, $u =$ unchanged, $q =$ value depends on condition, - = unimplemented, read as '0', $r =$ reserved.		

These registers can be addressed from any bank. Unimplemented, read as '1'. PIC16LF1904/7 only. Note 1:

Unimplemented

Unimplemented

2:

TADIE 2 5. SPECIAL EUNCTION DECISTED SUMMARY (CONTINUED)

IAD	LE 3-3: 3	SPECIAL	FUNCT		ISIER 3	UIVIIVIAR		INUED)			
Addr	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
Ban	k 7	•	•	•		•		•	•		
38Ch	_	Unimpleme	ented							_	_
 393h											
394h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	0000 0000	0000 0000
395h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	0000 0000	0000 0000
396h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	0000 0000	0000 0000
397h	—	Unimpleme	ented							-	—
39Fh											
Ban	ık 8-14										
x0Ch or x8Ch to x1Fh or x9Fh	_	Unimpleme	ented							_	_
Ban	ık 15										
78Ch 790h	_	Unimpleme	ented							-	—
791h	LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX1	LMUX0	000- 0011	000- 0011
792h	LCDPS	WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0	0000 0000	0000 0000
793h	LCDREF	LCDIRE	—	LCDIRI	-	VLCD3PE	VLCD2PE	VLCD1PE	—	0-0- 000-	0-0- 000-
794h	LCDCST	—	—	—	_	—	LCDCST2	LCDCST1	LCDCST0	000	000
795h	LCDRL	LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0	0000 -000	0000 -000
796h	—	Unimpleme	ented								—
797h	_	Unimpleme	ented							—	—
798h	LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000	uuuu uuuu
799n	LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000	uuuu uuuu
79An		SE23	SE22	SEZI	SE20	SE 19 SE 27	SE10	SE17	SE 10	0000 0000	uuuu uuuu
79Dh	_	Unimpleme	ented		0120	OLZI	OLZO	0125	0124		
79Fh		Unimpiente	incu								
7A0h	LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	XXXX XXXX	uuuu uuuu
7A1h	LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	XXXX XXXX	uuuu uuuu
7A2h	LCDDATA2 ⁽³⁾	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	XXXX XXXX	uuuu uuuu
7A3h	LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	XXXX XXXX	uuuu uuuu
7A4h	LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	XXXX XXXX	uuuu uuuu
7A5h	LCDDATA5 ⁽³⁾	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	XXXX XXXX	uuuu uuuu
7A6h	LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	XXXX XXXX	uuuu uuuu
7A7h	LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	XXXX XXXX	uuuu uuuu
7A8h	LCDDATA8 ⁽³⁾	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	XXXX XXXX	uuuu uuuu

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

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

Unimplemented, read as '1'. PIC16LF1904/7 only. 2:

Addr	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	15 (Continued)										
7A9h	LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	XXXX XXXX	uuuu uuuu
7AAh	LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	XXXX XXXX	uuuu uuuu
7ABh	LCDDATA11 ⁽³⁾	SEG23 COM3	SEG22 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	SEG15 COM3	XXXX XXXX	uuuu uuuu
7ACh	LCDDATA12	—	—	—	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0	x xxxx	u uuuu
7ADh	—	Unimpleme	nted							—	—
7AEh	_	Unimpleme	nted				_	_	-	—	_
7AFh	LCDDATA15	—	—	—	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1	x xxxx	u uuuu
7B0h	_	Unimpleme	nted							—	_
7B1h	—	Unimpleme	nted							—	—
7B2h	LCDDATA18	—	—	—	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2	x xxxx	u uuuu
7B3h	_	Unimpleme	nted								_
7B4h	_	Unimpleme	nted								_
7B5h	LCDDATA21	—	—	—	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3	x xxxx	u uuuu
7B6h —	—	Unimpleme	nted							-	—
7EFh											
Ban	k 16-30	-								1	
x0Ch or x8Ch to x1Fh or x9Fh	_	Unimpleme	nted							_	_
Ban	ik 31										
F8Ch — FE3h	_	Unimpleme	nted							_	-
FE4h	STATUS SHAD	_	_		_	_	Z SHAD	DC SHAD	C SHAD	xxx	uuu
FE5h	WREG SHAD	Working Re	egister Norm	al (Non-ICD) Shadow		_			xxxx xxxx	uuuu uuuu
FE6h	BSR SHAD	_	_	_	Bank Select	Register No	rmal (Non-IC	D) Shadow		x xxxx	u uuuu
FE7h	PCLATH_SHAD	_	Program Co	ounter Latch	High Registe	r Normal (No	on-ICD) Sha	dow		-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	Indirect Dat	ta Memory A	ddress 0 Lo	w Pointer Nor	mal (Non-IC	D) Shadow			xxxx xxxx	uuuu uuuu
FE9h	FSR0H_SHAD	Indirect Dat	ta Memory A	ddress 0 Hi	gh Pointer No	rmal (Non-IC	D) Shadow			xxxx xxxx	uuuu uuuu
FEAh	FSR1L_SHAD	Indirect Dat	ta Memory A	ddress 1 Lo	w Pointer Nor	rmal (Non-IC	D) Shadow			xxxx xxxx	uuuu uuuu
FEBh	FSR1H_SHAD	Indirect Dat	ta Memory A	ddress 1 Hi	gh Pointer No	rmal (Non-IC	D) Shadow			xxxx xxxx	uuuu uuuu
FECh	—	Unimpleme	nted							_	—
FEDh	STKPTR	_	_	_	Current Stac	k Pointer				1 1111	1 1111
FEEh	TOSL	Top of Stac	k Low byte		•					xxxx xxxx	uuuu uuuu
FEFh	тоѕн	—	Top of Stack	k High byte						-xxx xxxx	-uuu uuuu
L			1							1	1

TABLE 3-5: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'. These registers can be addressed from any bank. Legend:

1: 2: Note

Unimplemented, read as '1'. PIC16LF1904/7 only.

3.3 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 3-4 shows the five situations for the loading of the PC.

FIGURE 3-4: LOADING OF PC IN DIFFERENT SITUATIONS



3.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the program counter will change to the values contained in the PCLATH register.

3.3.2 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to Application Note AN556, *"Implementing a Table Read"* (DS00556).

3.3.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed calls by combining PCLATH and W to form the destination address. A computed CALLW is accomplished by loading the W register with the desired address and executing CALLW. The PCL register is loaded with the value of W and PCH is loaded with PCLATH.

3.3.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the W register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address PC + 1 + W.

If using BRA, the entire PC will be loaded with PC + 1 +, the signed value of the operand of the BRA instruction.

3.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 3-5). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer if the STVREN bit is programmed to '0' (Configuration Word 2). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

Note: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

3.4.1 ACCESSING THE STACK

The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

Note:	Care should be taken when modifying the
	STKPTR while interrupts are enabled.

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC and then decrement the STKPTR.

Reference Figure 3-5 through Figure 3-8 for examples of accessing the stack.



FIGURE 3-5: ACCESSING THE STACK EXAMPLE 1

		0x0F		
		0x0E		
		0x0D		_
		0x0C		_
		0x0B		_
		0x0A		_
		0x09		This figure shows the stack configuration
		0x08		after the first CALL or a single interrupt.
		0x07		return address will be placed in the
		0x06		decremented to the empty state (0x1F).
		0x05		_
		0x04		-
		0x03		_
		0x02		_
		0x01		
TOSH:TOSL		0x00	Return Address	STKPTR = 0x00
: 3-7: ACC	ESSING THE	E STA	CK EXAMPLE	3
E 3-7: ACC	ESSING THE	E STA	CK EXAMPLE	<u>3</u>
3-7: ACC	ESSING THE	5 STA 0x0F	CK EXAMPLE	<u>3</u>
: 3-7: ACC	ESSING THE	E STA 0x0F 0x0E	CK EXAMPLE	<u>3</u>
3-7: ACC	ESSING THE	E STA 0x0F 0x0E 0x0D 0x0D	CK EXAMPLE	3
: 3-7: ACC	ESSING THE	0x0F 0x0E 0x0D 0x0D	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt the stack looks like the figure
: 3-7: ACC	ESSING THE	0x0F 0x0F 0x0D 0x0D 0x0D 0x0D 0x0D 0x0D	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will be not the rotation of detections
3-7: ACC	ESSING THE	STA 0x0F 0x0E 0x0D 0x0D 0x0D 0x0D 0x0D 0x0A 0x0A	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
: 3-7: ACC	ESSING THE	E STA 0x0F 0x0D	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
: 3-7: ACC		0x0F 0x0F 0x0D 0x0D 0x0D 0x0D 0x0D 0x0D 0x0D 0x0D 0x0D 0x0B 0x0D 0x0B 0x0D 0x0B 0x0D 0x0B 0x0D 0x0B		3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
3-7: ACC		STA 0x0F 0x0E 0x0D 0x0D 0x0D 0x0D 0x0A 0x0B 0x0A 0x0B 0x0A 0x0B 0x0A 0x0A 0x0B 0x0A 0x0B	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
3-7: ACC		0x0F 0x0F 0x0D	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
3-7: ACC TOSH:TOSL		STA 0x0F 0x0D 0x005 0x04	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
3-7: ACC TOSH:TOSL		E STA 0x0F 0x0E 0x0D 0x0C 0x0A 0x0A 0x08 0x07 0x08 0x07 0x06 0x05 0x04 0x03	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
3-7: ACC		0x0F 0x0DF 0x0D 0x0D	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.
3-7: ACC TOSH:TOSL		STA 0x0F 0x0D 0x005 0x04 0x02 0x01	CK EXAMPLE	3 After seven CALLS or six CALLS and an interrupt, the stack looks like the figure on the left. A series of RETURN instructions will repeatedly place the return addresses into the Program Counter and pop the stack.

 $\ensuremath{\textcircled{}^{\odot}}$ 2011-2016 Microchip Technology Inc.

FIGURE 3-8: ACCESSING THE STACK EXAMPLE 4



3.4.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Word 2 is programmed to '1', the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

3.5 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory



3.5.1 TRADITIONAL DATA MEMORY

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

FIGURE 3-10: TRADITIONAL DATA MEMORY MAP


3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

FIGURE 3-11: LINEAR DATA MEMORY MAP



3.5.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-12: PROGRAM FLASH MEMORY MAP



4.0 DEVICE CONFIGURATION

Device Configuration consists of Configuration Word 1 and Configuration Word 2, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h and Configuration Word 2 at 8008h.

Note: The DEBUG bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.

R/P-1

U-1

R/P-1

bit 8

bit 0

CLKOUTEN BOREN<1:0> bit 13 R/P-1 R/P-1 R/P-1 R/P-1 R/P-1 U-1 R/P-1 **MCLRE** CP PWRTE WDTE<1:0> FOSC<1:0> bit 7 Legend: R = Readable bit P = Programmable bit U = Unimplemented bit, read as '1' '1' = Bit is set -n = Value when blank or after Bulk Erase '0' = Bit is cleared bit 13-12 Unimplemented: Read as '1' **CLKOUTEN:** Clock Out Enable bit bit 11 1 = CLKOUT function is disabled. I/O function on the CLKOUT pin. 0 = CLKOUT function is enabled on the CLKOUT pin bit 10-9 BOREN<1:0>: Brown-out Reset Enable bits 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 bit 8 Unimplemented: Read as '1' CP: Code Protection bit 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 WPUE3 bit. bit 5 **PWRTE:** Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled bit 4-3 WDTE<1:0>: Watchdog Timer Enable bit 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 00 = INTOSC oscillator: I/O function on CLKIN pin 01 = ECL: External Clock, Low-Power mode (0-0.5 MHz): device clock supplied to CLKIN pin

10 = ECM: External Clock, Medium-Power mode (0.5-4 MHz): device clock supplied to CLKIN pin 11 = ECH: External Clock, High-Power mode (4-20 MHz): device clock supplied to CLKIN pin

REGISTER 4-1: CONFIGURATION WORD 1

U-1

U-1

R/P-1

R/P-1

REGISTER 4-2: CONFIGURATION WORD 2

		R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1		
		LVP ⁽¹⁾	DEBUG ⁽³⁾	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		
	_	—	_	—	—	WRI	<1:0> hit 0		
							DILU		
Legend:									
R = Readal	ble bit	P = Programn	nable bit	U = Unimplem	nented bit, read	l as '1'			
'0' = Bit is c	leared	'1' = Bit is set		-n = Value wh	en blank or afte	er Bulk Erase			
bit 13 LVP: Low-Voltage Programming Enable bit ⁽¹⁾ 1 = Low-voltage programming enabled									
bit 12	DEBUG: In-C 1 = In-Circuit 0 = In-Circuit	Fircuit Debugge Debugger disal Debugger enab	r Mode bit ⁽³⁾ bled, ICSPCLK bled, ICSPCLK	and ICSPDAT	are general po are dedicated	urpose I/O pins to the debugge	er		
bit 11	LPBOR: Low 1 = Low-Powe 0 = Low-Powe	-Power BOR bi er BOR is disat er BOR is enab	t bled led						
bit 10	BORV: Brown 1 = Brown-ou 0 = Brown-ou	n-out Reset Vol t Reset voltage t Reset voltage	tage Selection (VBOR), low tri (VBOR), high t	bit ⁽²⁾ ip point selecte rip point select	ed				
bit 9	STVREN: Sta 1 = Stack Ove 0 = Stack Ove	ack Overflow/Ur erflow or Under erflow or Under	nderflow Reset flow will cause flow will not ca	Enable bit a Reset use a Reset					
bit 8-2	Unimplemen	ted: Read as '1	,						
bit 1-0 WRT<1:0>: Flash Memory Self-Write Protection bits <u>4 kW Flash memory (PIC16LF1904 only)</u> : 11 = Write protection off 10 = 000h to 1FFh write-protected, 200h to FFFh may be modified by PMCON control 01 = 000h to 7FFh write-protected, 800h to FFFh may be modified by PMCON control 00 = 000h to FFFh write-protected, no addresses may be modified by PMCON control <u>8 kW Flash memory (PIC16LF1906/7 only)</u> : 11 = Write protection off 10 = 0000h to 01EFh write protected, 200h to 1EFFh may be modified by PMCON control									
	 10 = 0000h to 01FFh write-protected, 200h to 1FFFh may be modified by PMCON control 01 = 0000h to 0FFFh write-protected, 1000h to 1FFFh may be modified by PMCON control 00 = 0000h to 1FFFh write-protected, no addresses may be modified by PMCON control 								
Note 1:	Iote 1: The LVP bit cannot be programmed to '0' when Programming mode is entered via LVP.								
2: 3:	See <u>VBOR</u> parame The DEBUG bit ir debuggers and pr	BOR parameter for specific trip point voltages. EBUG bit in Configuration Words is managed automatically by device development tools including gers and programmers. For normal device operation, this bit should be maintained as a '1'.							

4.2 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection is controlled independently. Internal access to the program memory is unaffected by any code protection setting.

4.2.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the \overline{CP} bit in Configuration Word 1. When $\overline{CP} = 0$, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See Section 4.3 "Write Protection" for more information.

4.3 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Word 2 define the size of the program memory block that is protected.

4.4 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See Section 10.4 "User ID, Device ID and Configuration Word Access" for more information on accessing these memory locations. For more information on checksum calculation, see the "PIC16F193X/LF193X/PIC16F194X/LF194X/PIC16LF 190X Memory Programming Specification" (DS41397).

4.5 **Device ID and Revision ID**

The memory location 8006h is where the Device ID and Revision ID are stored. The upper nine bits hold the Device ID. The lower five bits hold the Revision ID. See Section 10.4 "User ID, Device ID and Configuration Word Access" for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

REGISTER 4-3: DEVICEID: DEVICE ID REGISTER

		R	R	R	R	R	R
				DEV	<8:3>		
		bit 13					bit 8
R	R	R	R	R	R	R	R
	DEV<2:0>				REV<4:0>		
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit	•	U = Unimpleme	ented bit, read as	'1'	
u = Bit is uncha	anged	x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other Resets			

P = Programmable bit

bit 13-5 DEV<8:0>: Device ID bits

'1' = Bit is set

Dovico	DEVICEID<13:0> Values						
Device	DEV<8:0>	REV<4:0>					
PIC16LF1904	10 1100 100	x xxxx					
PIC16LF1906	10 1100 011	x xxxx					
PIC16LF1907	10 1100 010	x xxxx					

'0' = Bit is cleared

bit 4-0 REV<4:0>: Revision ID bits These bits are used to identify the revision (see Table under DEV<8:0> above).

A simplified block diagram of the On-Chip Reset Circuit

is shown in Figure 5-1.

5.0 RESETS

There are multiple ways to reset this device:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- Low-Power Brown-out Reset (LPBOR)
- MCLR Reset
- WDT Reset
- RESET instruction
- Stack Overflow
- Stack Underflow
- · Programming mode exit

To allow VDD to stabilize, an optional Power-up Timer can be enabled to extend the Reset time after a BOR or POR event.

FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



5.1 Power-on Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

5.1.1 POWER-UP TIMER (PWRT)

The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is enabled by clearing the PWRTE bit in Configuration Word 1.

The Power-up Timer starts after the release of the POR and BOR.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00607).

5.2 Brown-Out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Word 1. The four operating modes are:

- · BOR is always on
- · BOR is off when in Sleep
- · BOR is controlled by software
- · BOR is always off

Refer to Table 5-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV bit in Configuration Word 2.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Figure 5-2 for more information.

BOREN<1:0>	SBOREN	Device Mode	BOR Mode	Device Operation upon release of POR	Device Operation upon wake- up from Sleep
11	Х	Х	Active	Waits for BOR ready ⁽¹⁾	
1.0	37	Awake	Active	Waite for DOD ready	
10	X	Sleep	Disabled	waits for i	BOR ready
0.1	1	х	Active	Waits for B	OR ready ⁽¹⁾
UI	0	х	Disabled	Begins in	nmediately
00	X	х	Disabled	Begins immediately	

TABLE 5-1:BOR OPERATING MODES

Note 1: In these specific cases, "release of POR" and "wake-up from Sleep," there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits.

5.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Word 1 are set to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

5.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Word 1 are set to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

5.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Word 1 are set to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.



REGISTER 5-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS	—	—	—	—	—	BORRDY
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	q = Value depends on condition

bit 7	SBOREN: Software Brown-out Reset Enable bit
	If BOREN <1:0> in Configuration Word 1 ≠ 01:
	SBOREN is read/write, but has no effect on the BOR.
	If BOREN <1:0> in Configuration Word 1 = 01:
	1 = BOR Enabled
	0 = BOR Disabled
bit 6	BORFS: Brown-out Reset Fast Start bit ⁽¹⁾
	If BOREN<1:0> = 11 (Always on) or BOREN<1:0> = 00 (Always off)
	BORFS is Read/Write, but has no effect.
	If BOREN <1:0> = <u>10</u> (Disabled in Sleep) or BOREN<1:0> = <u>01</u> (Under software control):
	1 = Band gap is forced on always (covers Sleep/wake-up/operating cases)
	0 = Band gap operates normally, and may turn off
bit 5-1	Unimplemented: Read as '0'
bit 0	BORRDY: Brown-out Reset Circuit Ready Status bit
	1 = The Brown-out Reset circuit is active
	0 = The Brown-out Reset circuit is inactive

5.3 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) is an essential part of the Reset subsystem. Refer to Figure 5-1 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external VDD pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit (BOR) is changed to indicate that a BOR Reset has occurred. The same bit is set for both the BOR and the LPBOR. Refer to Register 5-2.

5.3.1 ENABLING LPBOR

The LPBOR is controlled by the LPBOR bit of Configuration Word 2. When the device is erased, the LPBOR module defaults to disabled.

5.3.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is to be OR'd together with the Reset signal of the BOR module to provide the generic BOR signal, which goes to the PCON register and to the power control block.

5.4 MCLR

The $\overline{\text{MCLR}}$ is an optional external input that can reset the device. The $\overline{\text{MCLR}}$ function is controlled by the MCLRE bit of Configuration Word 1 and the LVP bit of Configuration Word 2 (Table 5-2).

TABLE 5-2: MCLR CONFIGURATION

MCLRE	LVP	MCLR
0	0	Disabled
1	0	Enabled
x	1	Enabled

5.4.1 MCLR ENABLED

When MCLR is enabled and the pin is held low, the device is held in Reset. The MCLR pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

Note: A Reset does not drive the MCLR pin low.

5.4.2 MCLR DISABLED

When MCLR is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See Section 11.5 "PORTE Registers" for more information.

5.5 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The TO and PD bits in the STATUS register are changed to indicate the WDT Reset. See Section 9.0 "Watchdog Timer" for more information.

5.6 RESET Instruction

A RESET instruction will cause a device Reset. The \overline{RI} bit in the PCON register will be set to '0'. See Table 5-4 for default conditions after a RESET instruction has occurred.

5.7 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Word 2. See Section 5.7 "Stack Overflow/Underflow Reset" for more information.

5.8 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

5.9 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overrightarrow{\text{PWRTE}}$ bit of Configuration Word 1.

5.10 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

- 1. Power-up Timer runs to completion (if enabled).
- 2. Oscillator start-up timer runs to completion (if required for oscillator source).
- 3. MCLR must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 6.0 "Oscillator Module"** for more information.

The Power-up Timer and oscillator start-up timer run independently of MCLR Reset. If MCLR is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing MCLR high, the device will begin execution immediately (see Figure 5-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.



5.11 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON registers are updated to indicate the cause of the Reset. Table 5-3 and Table 5-4 show the Reset conditions of these registers.

STKOVF	STKUNF	RWDT	RMCLR	RI	POR	BOR	то	PD	Condition
0	0	1	1	1	0	x	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	Illegal, TO is set on POR
0	0	1	1	1	0	x	x	0	Illegal, PD is set on POR
0	0	u	1	1	u	0	1	1	Brown-out Reset
u	u	0	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	u	0	u	u	u	u	u	MCLR Reset during normal operation
u	u	u	0	u	u	u	1	0	MCLR Reset during Sleep
u	u	u	u	0	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

TABLE 5-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

TABLE 5-4: RESET CONDITION FOR SPECIAL REGISTERS⁽²⁾

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	1 1000	00-1 110x
MCLR Reset during normal operation	0000h	u uuuu	uu-u Ouuu
MCLR Reset during Sleep	0000h	1 Ouuu	uu-u Ouuu
WDT Reset	0000h	0 uuuu	uu-0 uuuu
WDT Wake-up from Sleep	PC + 1	0 Ouuu	uu-u uuuu
Brown-out Reset	0000h	1 luuu	00-1 11u0
Interrupt Wake-up from Sleep	PC + 1 ⁽¹⁾	1 Ouuu	uu-u uuuu
RESET Instruction Executed	0000h	u uuuu	uu-u u0uu
Stack Overflow Reset (STVREN = 1)	0000h	u uuuu	lu-u uuuu
Stack Underflow Reset (STVREN = 1)	0000h	u uuuu	ul-u uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

2: If a Status bit is not implemented, that bit will be read as '0'.

5.12 Power Control (PCON) Register

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

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

REGISTER 5-2: PCON: POWER CONTROL REGISTER

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR
bit 7	•		•				bit 0

r								
Legend:								
HC = Bit is cleared by hardware			HS = Bit is set by hardware					
R = Readable bit W = Writable bit			U = Unimplemented bit, read as '0'					
u = Bit is uncha	anged	x = Bit is unknown	-m/n = Value at POR and BOR/Value at all other Resets					
'1' = Bit is set		'0' = Bit is cleared	q = Value depends on condition					
bit 7	STKOVF: Sta	ck Overflow Flag bit						
	1 = A Stack C	Overflow occurred						
	0 = A Stack C	Overflow has not occurred o	or set to '0' by firmware					
bit 6	STKUNF: Sta	ck Underflow Flag bit						
	$1 = A \operatorname{Stack} l$	Inderflow occurred						
	$0 = A \operatorname{Stack} U$	Inderflow has not occurred	or set to '0' by firmware					
bit 5	Unimplemen	ted: Read as '0'						
bit 4	RWDT: Watch	ndog Timer Reset Flag bit						
	1 = A Watchd 0 = A Watchd	og Timer Reset has not oco og Timer Reset has occurre	curred or set to '1' by firmware ed (set to '0' in hardware when a Watchdog Timer Reset)					
bit 3	RMCLR: MCI	R Reset Flag bit						
	1 = A MCLR	Reset has not occurred or s	et to '1' by firmware					
	0 = A MCLR I	Reset has occurred (set to '	0' in hardware when a MCLR Reset occurs)					
bit 2	RI: RESET INS	struction Flag bit						
	1 = A RESET	instruction has not been ex	ecuted or set to '1' by firmware					
	$0 = \mathbf{A}$ RESET i	nstruction has been execute	ed (set to '0' in hardware upon executing a RESET instruction)					
bit 1	POR: Power-	on Reset Status bit						
1 = No Power-on Reset occurred			a act in a officiar o Davier on Deact accure)					
h:+ 0	0 = A Power-0	on Reset occurred (must be	e set in soliware aller a Power-on Reset occurs)					
DIT U	BOR: Brown-	out Reset Status bit						
⊥ = NO Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset)								

The PCON register bits are shown in Register 5-2.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	BORFS			—		_	BORRDY	45
PCON	STKOVF	STKUNF	_	RWDT	RMCLR	RI	POR	BOR	49
STATUS	—	_		TO	PD	Z	DC	С	21
WDTCON	—			V	VDTPS<4:0	>		SWDTEN	75

TABLE 5-5: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

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

6.0 OSCILLATOR MODULE

6.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 6-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external clock circuits. In addition, the system clock source can be supplied from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Fast start-up oscillator allows internal circuits to power up and stabilize before switching to the 16 MHz HFINTOSC

The oscillator module can be configured in one of the following clock modes:

- 1. ECL External Clock Low-Power mode (0 MHz to 0.5 MHz)
- 2. ECM External Clock Medium-Power mode (0.5 MHz to 4 MHz)
- ECH External Clock High-Power mode (4 MHz to 20 MHz)
- 4. INTOSC Internal oscillator (31 kHz to 16 MHz).

Clock Source modes are selected by the FOSC<1:0> bits in the Configuration Word 1. The FOSC bits determine the type of oscillator that will be used when the device is first powered.

The EC clock mode relies on an external logic level signal as the device clock source.

The INTOSC internal oscillator block produces a low and high-frequency clock source, designated LFINTOSC and HFINTOSC (see Internal Oscillator Block, Figure 6-1). A wide selection of device clock frequencies may be derived from these two clock sources.



SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM FIGURE 6-1:

6.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on external circuitry for the clock source to function. An example is: oscillator module (EC mode) circuit.

Internal clock sources are contained internally within the oscillator module. The internal oscillator block has two internal oscillators that are used to generate the internal system clock sources: the 16 MHz High-Frequency Internal Oscillator and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bits in the OSCCON register. See **Section 6.3 "Clock Switching"** for additional information.

6.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in the Configuration Word 1 to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to:
 - Secondary oscillator during run-time, or
 - An external clock source determined by the value of the FOSC bits.

See Section 6.3 "Clock Switching" for more information.

6.2.1.1 EC Mode

The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. CLKOUT is available for general purpose I/O or CLKOUT. Figure 6-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through Configuration Word 1:

- High power, 4-20 MHz (FOSC = 11)
- Medium power, 0.5-4 MHz (FOSC = 10)
- Low power, 0-0.5 MHz (FOSC = 01)

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC[®] MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.



EXTERNAL CLOCK (EC) MODE OPERATION



6.2.1.2 Secondary Oscillator

The secondary oscillator is a separate crystal oscillator that is associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the T1CKI/T1OSO and T1OSI device pins.

The secondary oscillator can be used as an alternate system clock source and can be selected during run-time using clock switching. Refer to **Section 6.3 "Clock Switching"** for more information.

FIGURE 6-3: QUARTZ CRYSTAL OPERATION (SECONDARY OSCILLATOR)



- Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.
 - Always verify oscillator performance over the VDD and temperature range that is expected for the application.
 - **3:** For oscillator design assistance, reference the following Microchip Applications Notes:
 - AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC[®] and PIC[®] Devices" (DS00826)
 - AN849, "Basic PIC[®] Oscillator Design" (DS00849)
 - AN943, "Practical PIC[®] Oscillator Analysis and Design" (DS00943)
 - AN949, "Making Your Oscillator Work" (DS00949)
 - TB097, "Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS" (DS91097)
 - AN1288, "Design Practices for Low-Power External Oscillators" (DS01288)

6.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in Configuration Word 1 to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run-time. See Section 6.3 "Clock Switching"for more information.

In **INTOSC** mode, CLKIN is available for general purpose I/O. CLKOUT is available for general purpose I/O or CLKOUT.

The function of the CLKOUT pin is determined by the state of the CLKOUTEN bit in Configuration Word 1.

The internal oscillator block has two independent oscillators that provides the internal system clock source.

- 1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz.
- 2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

6.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source.

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 6-1). The frequency derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See Section 6.2.2.4 "Internal Oscillator Clock Switch Timing" for more information.

The HFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<1:0> = 11, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'.

The High-Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running and can be utilized.

The High-Frequency Internal Oscillator Status Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

6.2.2.2 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a multiplexer (see Figure 6-1). Select 31 kHz, via software, using the IRCF<3:0> bits of the OSCCON register. See Section 6.2.2.4 "Internal Oscillator Clock Switch Timing" for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT) and Watchdog Timer (WDT).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<3:0> bits of the OSCCON register = 000) as the system clock source (SCS bits of the OSCCON register = 1x), or when any of the following are enabled:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired LF frequency, and
- FOSC<1:0> = 01, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

Peripherals that use the LFINTOSC are:

- Power-up Timer (PWRT)
- Watchdog Timer (WDT)

The Low-Frequency Internal Oscillator Ready bit (LFIOFR) of the OSCSTAT register indicates when the LFINTOSC is running and can be utilized.

6.2.2.3 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The output of the 16 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 6-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

Note: Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

6.2.2.4 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 6-4). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

- 1. IRCF<3:0> bits of the OSCCON register are modified.
- 2. If the new clock is shut down, a clock start-up delay is started.
- 3. Clock switch circuitry waits for a falling edge of the current clock.
- 4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
- 5. The new clock is now active.
- 6. The OSCSTAT register is updated as required.
- 7. Clock switch is complete.

See Figure 6-4 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 6-2.

Start-up delay specifications are located in the oscillator tables of **Section 22.0** "**Electrical Specifications**".

FIGURE 6-4:	INTERNAL OSCILLATOR SWITCH TIMING
HFINTOSC -	LFINTOSC (WDT disabled)
HFINTOSC	Oscillator Delay ⁽¹⁾ 2-cycle Sync Running
LFINTOSC	
IRCF <3:0>	$\neq 0$ $= 0$
System Clock	
HFINTOSC →	LFINTOSC (WDT enabled)
HFINTOSC	2-cycle Sync
LFINTOSC	
IRCF <3:0>	$\neq 0$ $X = 0$
System Clock	
LFINTOSC \rightarrow	HFINTOSC
LFINTOSC	
HFINTOSC	
IRCF <3:0>	= 0 X ≠ 0
System Clock	
Note 1: See	Table 6-1, "Oscillator Switching Delays" for more information.

TABLE 6-1: OSCILLATOR SWITCHING DELAYS

Switch From	Switch To	Oscillator Delay		
	LFINTOSC	1 cycle of each clock source		
Any clock course	HFINTOSC	2 μs (approx.)		
Ally Clock Source	ECH, ECM, ECL	2 cycles		
	Secondary Oscillator	1024 Secondary Oscillator Cycles		

6.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS bits:

- Default system oscillator determined by FOSC bits in Configuration Word 1
- Secondary oscillator 32 kHz crystal
- Internal Oscillator Block (INTOSC)

6.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS) bits of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by value of the FOSC<1:0> bits in the Configuration Word 1.
- When the SCS bits of the OSCCON register = 01, the system clock source is the secondary oscillator.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 6-2.

6.3.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<1:0> bits in the Configuration Word 1, or from the internal clock source. The OST does not reflect the status of the secondary oscillator.

6.3.3 SECONDARY OSCILLATOR

The secondary oscillator is a separate crystal oscillator associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the T1OSI and T1CKI/T1OSO device pins.

The secondary oscillator is enabled using the T1OSCEN control bit in the T1CON register. See **Section 17.0 "Timer1 Module with Gate Control**" for more information about the Timer1 peripheral.

6.3.4 SECONDARY OSCILLATOR READY (T1OSCR) BIT

The user must ensure that the secondary oscillator is ready to be used before it is selected as a system clock source. The Secondary Oscillator Ready (T1OSCR) bit of the OSCSTAT register indicates whether the secondary oscillator is ready to be used. After the T1OSCR bit is set, the SCS bits can be configured to select the secondary oscillator.

6.4 Oscillator Control Registers

REGISTER 6-1: OSCCON: OSCILLATOR CONTROL REGISTER

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 = Reada	able bit	W = Writable	bit	U = Unimplen	nented bit, read	d as '0'				
u = Bit is u	inchanged	x = Bit is unkr	nown	-n/n = Value a	at POR and BC	R/Value at all	other Resets			
'1' = Bit is	set	'0' = Bit is cle	ared							
bit 7	Unimplemen	ted: Read as '	0'							
bit 6-3	IRCF<3:0>:	nternal Oscillat	or Frequency	Select bits						
	000x = 31 kH	000x = 31 kHz LF								
	001x = 31.25	5 kHz								
	0100 = 62.5	kHz								
	0101 = 125 k									
	0110 = 250 k 0111 = 500 k	uiz Hz (default un	n Reset)							
	1000 = 125 k	(Hz ⁽¹⁾								
	1001 = 250 k	(Hz ⁽¹⁾								
	1010 = 500 k	(Hz ⁽¹⁾								
	1011 = 1 MH	z								
	1100 = 2 MH	z								
	1101 = 4 MH	Z								
	1110 = 8 MH	IZ								
h:+ 0			o'							
	Unimplemen	ited: Read as	0							
bit 1-0	SCS<1:0>: S	System Clock S	elect bits							
	1x = Internal	oscillator block	C							
	01 = Second	ary oscillator			lord 1					
		etermined by F	000-1.0/11							
Note 1:	Duplicate frequen	cy derived from	HFINTOSC.							

R-1/q	U-0	R-q/q	R-0/q	U-0	U-0	R-0/0	R-0/q
T10SCR	_	OSTS	HFIOFR	—	_	LFIOFR	HFIOFS
bit 7						I	bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, read	d as '0'	
u = Bit is uncha	anged	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is cle	ared	q = Condition	nal		
bit 7	T1OSCR: Tim <u>If T1OSCEN</u> : 1 = Timer1 c 0 = Timer1 c <u>If T1OSCEN</u> : 1 = Timer1 c	ther1 Oscillator = 1: oscillator is real oscillator is not = 0: clock source is	Ready bit dy ready always ready				
bit 6	Unimplemen	ted: Read as '	0'				
bit 5	OSTS: Oscilla 1 = Running 0 = Running	ator Start-up Ti from the exter from an intern	me-out Status nal clock sour al oscillator (F	bit ce (EC) OSC<1:0> = 0	0)		
bit 4	HFIOFR: High 1 = HFINTOS 0 = HFINTOS	h-Frequency Ir SC is ready SC is not ready	iternal Oscillat	or Ready bit			
bit 3-2	Unimplemen	ted: Read as '	0'				
bit 1	LFIOFR: Low-Frequency Internal Oscillator Ready bit 1 = LFINTOSC is ready 0 = LFINTOSC is not ready						
bit 0	 HFINTOSC is not ready HFIOFS: High-Frequency Internal Oscillator Stable bit HFINTOSC 16 MHz oscillator is stable and is driving the INTOSC HFINTOSC 16 MHz oscillator is not stable, the start-up oscillator is driving INTOSC 						

REGISTER 6-2: OSCSTAT: OSCILLATOR STATUS REGISTER

TABLE 6-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON			IRCF<3:0>				SCS<1:0>		58
OSCSTAT	T1OSCR	—	OSTS	HFIOFR	-	-	LFIOFR	HFIOFS	59
T1CON	TMR10	CS<1:0>	T1CKP	S<1:0>	T1OSCEN	T1SYNC		TMR10N	139

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

TABLE 6-3: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
	13:8	_	_	_	—	CLKOUTEN	BOREI	N<1:0>	—	20
CONFIGT	7:0	CP	MCLRE	PWRTE	WDTE<1:0>			FOSC	<1:0>	- 39

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

7.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

This chapter contains the following information for Interrupts:

- · Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- · Automatic Context Saving

Many peripherals produce Interrupts. Refer to the corresponding chapters for details.

FIGURE 7-1: INTERRUPT LOGIC

TMR0IF Wake-up TMR0IE (If in Sleep mode) INTE **Peripheral Interrupts** INTE (TMR1IF) PIR1<0> **IOCIF** Interrupt (TMR1IF) PIR1<0> IOCIE to CPU PEIE PIRn<7> GIE PIEn<7>

A block diagram of the interrupt logic is shown in Figure 7.1.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- · GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 and PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- · Current prefetched instruction is flushed
- · GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See Section 7.5 "Automatic Context Saving")
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The RETFIE instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual	inte	rrupt	flag	bits	s are	e set,
	regardless	of	the	state	of	any	other
	enable bits						

2: All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7.3 for more details.



	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4
CLKOUT			Interru during	pt Sampled Q1				
Interrupt								
GIE								
PC	PC-1	PC	PC	+1	0004h	0005h		
Execute	1 Cycle Insti	ruction at PC	Inst(PC)	NOP	NOP	Inst(0004h)		
Interrupt								
GIE								
PC	PC-1	РС	PC+1/FSR	New PC/	0004h	0005h		
Execute-	2 Cycle Insti	ruction at PC	Inst(PC)	NOP	NOP	Inst(0004h)		
Interrupt								
GIE								
			[]	[]		\		
PC	PC-1	PC	FSR ADDR	PC+1	PC+2	0004h	0005h	
Execute	3 Cycle Inst	ruction at PC	INST(PC)	NOP	NOP	NOP	Inst(0004h)	Inst(0005h)
Interrupt								
GIE								
PC	PC-1	PC	FSR ADDR	PC+1	P	0+2	0004h	0005h
Execute	3 Cycle Insti	ruction at PC	INST(PC)	NOP	NOP	NOP	NOP	Inst(0004h)



5: INTF is enabled to be set any time during the Q4-Q1 cycles.

7.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate interrupt enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to **Section 8.0** "**Power-Down Mode (Sleep)**" for more details.

7.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

7.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for TO and PD)
- · BSR register
- · FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

7.6 Interrupt Control Registers

7.6.1 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, interrupt-on-change and external INT pin interrupts.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 7-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0/0	R-0/0						
GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF
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	GIE: Global Interrupt Enable bit 1 = Enables all active interrupts 0 = Disables all interrupts
bit 6	PEIE: Peripheral Interrupt Enable bit 1 = Enables all active peripheral interrupts 0 = Disables all peripheral interrupts
bit 5	TMR0IE: Timer0 Overflow Interrupt Enable bit 1 = Enables the Timer0 interrupt 0 = Disables the Timer0 interrupt
bit 4	INTE: INT External Interrupt Enable bit 1 = Enables the INT external interrupt 0 = Disables the INT external interrupt
bit 3	IOCIE: Interrupt-on-Change Interrupt Enable bit 1 = Enables the interrupt-on-change interrupt 0 = Disables the interrupt-on-change interrupt
bit 2	TMR0IF: Timer0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed 0 = TMR0 register did not overflow
bit 1	INTF: INT External Interrupt Flag bit 1 = The INT external interrupt occurred 0 = The INT external interrupt did not occur
bit 0	IOCIF: Interrupt-on-Change Interrupt Flag bit 1 = When at least one of the interrupt-on-change pins changed state 0 = None of the interrupt-on-change pins have changed state

7.6.2 PIE1 REGISTER

bit 5

bit 4

bit 3-1 bit 0

The PIE1 register contains the interrupt enable bits, as shown in Register 7-2.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 7-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

RCIE: USART Receive Interrupt Enable bit 1 = Enables the USART receive interrupt 0 = Disables the USART receive interrupt

TXIE: USART Transmit Interrupt Enable bit 1 = Enables the USART transmit interrupt 0 = Disables the USART transmit interrupt

TMR1IE: Timer1 Overflow Interrupt Enable bit 1 = Enables the Timer1 overflow interrupt 0 = Disables the Timer1 overflow interrupt

Unimplemented: Read as '0'

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	—	—	—	TMR1IE
bit 7							bit 0

Legend:			
R = Readable	bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unch	anged	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	TMR1GIE: T	imer1 Gate Interrupt Enab	le bit
	1 = Enables f	the Timer1 gate acquisitio	n interrupt
	0 = Disables	the Timer1 gate acquisition	on interrupt
bit 6	ADIE: A/D C	onverter (ADC) Interrupt E	Enable bit
	1 = Enables f	the ADC interrupt	
	0 = Disables	the ADC interrupt	

7.6.3 PIE2 REGISTER

The PIE2 register contains the interrupt enable bits, as shown in Register 7-3.

Note:	Bit PEIE of the INTCON register must be
	set to enable any peripheral interrupt.

REGISTER 7-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	U-0	U-0	U-0	U-0	R/W-0/0	U-0	U-0
—	—	—	—	—	LCDIE	—	—
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 7-3	Unimplemented: Read as '0'
bit 2	LCDIE: LCD Module Interrupt Enable bit
	1 = Enables the LCD module interrupt
	0 = Disables the LCD module interrupt
bit 1-0	Unimplemented: Read as '0'

7.6.4 PIR1 REGISTER

The PIR1 register contains the interrupt flag bits, as shown in Register 7-4.

0 = Interrupt is not pending

1 = Interrupt is pending0 = Interrupt is not pending

Unimplemented: Read as '0'

TMR1IF: Timer1 Overflow Interrupt Flag bit

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 7-4: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	—	—	—	TMR1IF
bit 7							bit 0

	Legend:							
R = Readable bit		bit W = Writable bit	U = Unimplemented 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 cleared						
	bit 7	TMR1GIF: Timer1 Gate Interrupt Flag bit						
		1 = Interrupt is pending						
		0 = Interrupt is not pending						
	bit 6	ADIF: A/D Converter Interrupt Flag bit						
		1 = Interrupt is pending						
		0 = Interrupt is not pending						
	bit 5	RCIF: USART Receive Interrupt Flag bit						
		1 = Interrupt is pending						
		0 = Interrupt is not pending						
	bit 4	TXIF: USART Transmit Interrupt Flag bit						
		1 = Interrupt is pending						

bit 3-1

bit 0

7.6.5 PIR2 REGISTER

The PIR2 register contains the interrupt flag bits, as shown in Register 7-5.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 7-5: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

U-0	U-0	U-0	U-0	U-0	R/W-0/0	U-0	U-0
—	—	—	—	—	LCDIF	—	—
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 7-3 Unimplemented: Read as '0'
- bit 2 LCDIF: LCD Module Interrupt Flag bit 1 = Interrupt is pending 0 = Interrupt is not pending
- bit 1-0 Unimplemented: Read as '0'

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	65
OPTION_REG	WPUEN	INTEDG	TOCS	T0SE	PSA	PS<2:0>			130
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—		TMR1IE	66
PIE2	_		_		_	LCDIE		_	67
PIR1	TMR1GIF	ADIF	RCIF	TXIF	_	_	_	TMR1IF	68
PIR2	_	_	_	_	_	LCDIF	_	_	69

TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

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

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. Secondary oscillator is unaffected and peripherals that operate from it may continue operation in Sleep.
- 7. ADC is unaffected, if the dedicated FRC clock is selected.
- 8. Capacitive Sensing oscillator is unaffected.
- 9. I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).
- 10. 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
- · Modules using Secondary oscillator

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 **13.0** "Fixed Voltage Reference (FVR)" for more information.

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 5.11, 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 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.

CLKIN ⁽¹⁾	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1	- - 	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4
CLKOUT ⁽²⁾	<u></u> /	, //	<u>\</u>	1 1 	· · / /	//	\/	
Interrupt flag				ı ∙ ≠	Interrupt Laten	су(1)		
GIE bit (INTCON reg.)		<u>.</u> 	Processor in Sleep		<u>.</u> 			
Instruction Flow PC	(PC	X PC + 1) Х РС	+ 2	X PC + 2	X PC + 2	X 0004h	x 0005h
Instruction {	Inst(PC) = Sleep	Inst(PC + 1)	1 1 1		Inst(PC + 2)	i i i i	Inst(0004h)	Inst(0005h)
Instruction { Executed {	Inst(PC - 1)	Sleep	1 1 1		Inst(PC + 1)	Dummy Cycle	Dummy Cycle	Inst(0004h)
Note 1: 0	GIE = 1 assumed.	In this case after	wake-up, the	processo	r calls the ISR at (0004h. If GIE = 0,	execution will cont	inue in-line.

FIGURE 8-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT

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

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	65
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	110
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	110
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	110
PIE1	TMR1GIE	ADIE	RCIE	TXIE	_	_	—	TMR1IE	66
PIE2	—	_	_	_	_	LCDIE	—	—	67
PIR1	TMR1GIF	ADIF	RCIF	TXIF	_	_	—	TMR1IF	68
PIR2	—	_	_	_	_	LCDIF	—	—	69
STATUS	—	_	_	TO	PD	Z	DC	С	21
WDTCON	_		WDTPS<4:0>					SWDTEN	75

Legend: — = unimplemented location, read as '0'. Shaded cells are not used in Power-down mode.
9.0 WATCHDOG TIMER

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 (typical)
- Multiple Reset conditions
- Operation during Sleep

FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM



9.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1ms. See **Section 22.0 "Electrical Specifications"** for the LFINTOSC tolerances.

9.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Word 1. See Table 9-1.

9.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Word 1 are set to '11', the WDT is always on.

WDT protection is active during Sleep.

9.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Word 1 are set to '10', the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

9.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Word 1 are set to '01', the WDT is controlled by the SWDTEN bit of the WDTCON register.

WDT protection is unchanged by Sleep. See Table 9-1 for more details.

TABLE 9-1:	WDT OPERATING MODES
------------	---------------------

WDTE<1:0>	SWDTEN	Device Mode	WDT Mode
11	х	Х	Active
10	37	Awake	Active
TO	X	Sleep	Disabled
01	1	~	Active
UT	0	~	Disabled
0.0	х	Х	Disabled

TABLE 9-2: WDT CLEARING CONDITIONS

Conditions	WDT
WDTE<1:0> = 00	
WDTE<1:0> = 01 and SWDTEN = 0	
WDTE<1:0> = 10 and enter Sleep	Cleared
CLRWDT Command	Cleared
Oscillator Fail Detected	
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK	
Change INTOSC divider (IRCF bits)	Unaffected

9.3 Time-Out Period

The WDTPS bits of the WDTCON register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is two seconds.

9.4 Clearing the WDT

The WDT is cleared when any of the following conditions occur:

- Any Reset
- CLRWDT instruction is executed
- · Device enters Sleep
- · Device wakes up from Sleep
- Oscillator fail event
- WDT is disabled
- Oscillator Start-up TImer (OST) is running

See Table 9-2 for more information.

9.5 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting.

When the device exits Sleep, the WDT is cleared again. The WDT remains clear until the OST, if enabled, completes. See **Section 6.0** "Oscillator **Module**" for more information on the OST.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the <u>device</u> wakes up and resumes operation. The TO and PD bits in the STATUS register are changed to indicate the event. See **Section 3.0** "Memory Organization" and STATUS register (Register 3-1) for more information.

9.6 Watchdog Control Register

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
_	—			WDTPS<4:02	>		SWDTEN
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	i as '0'	
u = Bit is unch	anged	x = Bit is unkr	nown	-m/n = Value	at POR and BC	R/Value at all	other Resets
'1' = Bit is set	-	'0' = Bit is clea	ared				
							I
bit 7-6	Unimplemen	ted: Read as '	0'				
bit 5-1	WDTPS<4:0:	>: Watchdog Ti	mer Period Se	elect bits ⁽¹⁾			
	Bit Value = F	Prescale Rate					
	00000 = 1:3	32 (Interval 1 m	s nominal)				
	00001 = 1:6	64 (Interval 2 m	s nominal)				
	00010 = 1:1	28 (Interval 4 r	ns nominal)				
	00011 = 1:2	256 (Interval 8 r	ns nominal)				
	00100 = 1:5	512 (Interval 16	ms nominal)				
	00101 = 1:1	024 (Interval 3	2 ms nominal)			
	00110 = 1:2	2048 (Interval 6	4 ms nominal)			
	00111 = 1.4 01000 = 1.8	1090 (Interval 1.	20 ms nomina 56 ms nomina	41 <i>)</i>			
	01000 = 1.0 01001 = 1.1	16384 (Interval	512 ms nomir	nal)			
	01010 = 1:3	32768 (Interval	1s nominal)				
	01011 = 1:6	5536 (Interval	2s nominal) (Reset value)			
	01100 = 1:1	l31072 (2 ¹⁷) (Ir	nterval 4s non	ninal)			
	01101 = 1:2	262144 (2 ¹⁸) (Ir	nterval 8s nom	ninal)			
	01110 = 1:5	524288 (2 ¹⁹) (Ir	nterval 16s no	minal)			
	01111 = 1:1	1048576 (2 ²⁰) (1	Interval 32s n	ominal)			
	10000 = 1:2	2097152 (2 ²¹) (Interval 64s n	ominal)			
	10001 = 1:4	194304 (2 ²²) (Interval 128s	nominal)			
	10010 = 1:8	3388608 (223) (Interval 256s	nominal)			
	10011 = Re	served. Result	s in minimum	interval (1:32)			
	•			, , , , , , , , , , , , , , , , , , ,			
	•						
	•						
	11111 = Re	served. Result	s in minimum	interval (1:32)			
bit 0	SWDTEN: So	oftware Enable/	Disable for W	atchdog Timer	bit		
	<u>If WDTE<1:0</u>	> = <u>00</u> :					
	This bit is ign	ored.					
	If WDTE<1:0	> = 01:					
	1 = WDT is t	urned on					
	This bit is ion	$\sim -1x$.					
	i ilis bit is igri						

REGISTER 9-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

Note 1: Times are approximate. WDT time is based on 31 kHz LFINTOSC.

PIC16LF1904/6/7

TABLE 9-3:	SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER
------------	---

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	—		IRCF<3:0>			—	SCS	<1:0>	58
STATUS	—	_	—	TO	PD	Z	DC	С	21
WDTCON	—		— WDTPS<4:0>					SWDTEN	75

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Watchdog Timer.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH WATCHDOG TIMER

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	_	—	_	_	CLKOUTEN	BOREI	N<1:0>	—	20
	7:0	CP	MCLRE	PWRTE	WDTE	E<1:0>		FOSC	<1:0>	- 39

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

10.0 FLASH PROGRAM MEMORY CONTROL

The Flash Program Memory is readable and writable during normal operation over the full VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When accessing the program memory, the PMDATH:PMDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the PMADRH:PMADRL register pair forms a 2-byte word that holds the 15-bit address of the program memory location being read.

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the operating voltage range of the device.

The Flash Program Memory can be protected in two ways; by code protection (CP bit in Configuration Word 1) and write protection (WRT<1:0> bits in Configuration Word 2).

Code protection ($\overline{CP} = 0$)⁽¹⁾, disables access, reading and writing, to the Flash Program Memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all Flash Program Memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the Flash Program Memory as defined by the bits WRT<1:0>. Write protection does not affect a device programmers ability to read, write or erase the device.

Note 1: Code protection of the entire Flash Program Memory array is enabled by clearing the CP bit of Configuration Word 1.

10.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address up to a maximum of 32K words of program memory. When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADRL register.

10.1.1 PMCON1 AND PMCON2 REGISTERS

PMCON1 is the control register for Flash Program Memory accesses.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared by hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

The PMCON2 register is a write-only register. Attempting to read the PMCON2 register will return all '0's.

To enable writes to the program memory, a specific pattern (the unlock sequence), must be written to the PMCON2 register. The required unlock sequence prevents inadvertent writes to the program memory write latches and Flash Program Memory.

10.2 Flash Program Memory Overview

It is important to understand the Flash Program Memory structure for erase and programming operations. Flash Program Memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum size that can be erased by user software.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the PMDATH:PMDATL register pair.

See Table 10-1 for Erase Row size and the number of write latches for Flash program memory.

Note: If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase. Then, new data and retained data can be written into the write latches to reprogram the row of Flash Program Memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations.

TABLE 10-1:FLASH MEMORYORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC16LF1904/6/7	32	32

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

- 1. Write the desired address to the PMADRH:PMADRL register pair.
- 2. Clear the CFGS bit of the PMCON1 register.
- 3. Then, set control bit RD of the PMCON1 register.

Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF PMCON1, RD" instruction to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

PMDATH: PMDATL register pair will hold this value until another read or until it is written to by the user.

Note:	The two instructions following a program
	memory read are required to be NOPS.
	This prevents the user from executing a
	2-cycle instruction on the next instruction
	after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ





EXAMPLE 10-1: FLASH PROGRAM MEMORY READ

* This code block will read 1 word of program * memory at the memory address: PROG_ADDR_HI : PROG_ADDR_LO data will be returned in the variables; * PROG_DATA_HI, PROG_DATA_LO BANKSEL PMADRL ; Select Bank for PMCON registers MOVLW PROG_ADDR_LO ; MOVWF PMADRL ; Store LSB of address PROG_ADDR_HI MOVLW ; MOVWL PMADRH ; Store MSB of address BCF PMCON1,CFGS ; Do not select Configuration Space BSF PMCON1,RD ; Initiate read NOP ; Ignored (Figure 10-1) NOP ; Ignored (Figure 10-1) MOVF PMDATL,W ; Get LSB of word MOVWF PROG_DATA_LO ; Store in user location ; Get MSB of word MOVF PMDATH,W MOVWF PROG_DATA_HI ; Store in user location

10.2.2 FLASH MEMORY UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the Flash Program Memory from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- Row Erase
- · Load program memory write latches
- Write of program memory write latches to program memory
- Write of program memory write latches to User IDs

The unlock sequence consists of the following steps:

- 1. Write 55h to PMCON2
- 2. Write AAh to PMCON2
- 3. Set the WR bit in PMCON1
- 4. NOP instruction
- 5. NOP instruction

Once the WR bit is set, the processor will always force two NOP instructions. When an Erase Row or Program Row operation is being performed, the processor will stall internal operations (typical 2 ms), until the operation is complete and then resume with the next instruction. When the operation is loading the program memory write latches, the processor will always force the two NOP instructions and continue uninterrupted with the next instruction.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 10-3: FLASH PROGRAM MEMORY UNLOCK



10.2.3 ERASING FLASH PROGRAM MEMORY

While executing code, program memory can only be erased by rows. To erase a row:

- 1. Load the PMADRH:PMADRL register pair with any address within the row to be erased.
- 2. Clear the CFGS bit of the PMCON1 register.
- 3. Set the FREE and WREN bits of the PMCON1 register.
- 4. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
- 5. Set control bit WR of the PMCON1 register to begin the erase operation.

See Example 10-2.

After the "BSF PMCON1, WR" instruction, the processor requires two cycles to set up the erase operation. The user must place two NOP instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction.

FIGURE 10-4:

FLASH PROGRAM MEMORY ERASE FLOWCHART



PIC16LF1904/6/7

EXAMPLE 10-2: ERASING ONE ROW OF PROGRAM MEMORY

- ; This row erase routine assumes the following:
- ; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
- ; 2. ADDRH and ADDRL are located in shared data memory $0\,\mathrm{x}70$ $0\,\mathrm{x}7F$ (common RAM)

	BCF BANKSEL MOVF MOVWF	INTCON,GIE PMADRL ADDRL,W PMADRL	; Disable ints so required sequences will execute properly ; Load lower 8 bits of erase address boundary
	MOVF MOVWF	ADDRH,W PMADRH	; Load upper 6 bits of erase address boundary
	BCF	PMCON1,CFGS	; Not configuration space
	BSF	PMCON1, FREE	; Specify an erase operation
	BSF	PMCON1,WREN	; Enable writes
Required Sequence	MOVLW MOVWF MOVWF BSF NOP NOP	55h PMCON2 0AAh PMCON2 PMCON1,WR	<pre>; Start of required sequence to initiate erase ; Write 55h ; ; Write AAh ; Set WR bit to begin erase ; NOP instructions are forced as processor starts ; row erase of program memory. ; ; The processor stalls until the erase process is complete ; after erase processor continues with 3rd instruction</pre>
	BCF BSF	PMCON1,WREN INTCON,GIE	; Disable writes ; Enable interrupts

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 10-2 (row writes to program memory with 32 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper ten bits of PMADRH:PMADRL, (PMADRH<6:0>:PMADRL<7:5>) with the lower five bits of PMADRL, (PMADRL<4:0>) 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.
- 11. 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.



PIC16LF1904/6/7





PIC16LF1904/6/7

EXAMPLE 10-3: WRITING TO FLASH PROGRAM MEMORY

; This write routine assumes the following: ; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR ; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR, stored in little endian format ; ; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH: ADDRL ; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM) INTCON,GIE BCF ; Disable ints so required sequences will execute properly PMADRH ; Bank 3 BANKSEL MOVF ADDRH,W ; Load initial address MOVWF PMADRH MOVF ADDRL,W MOVWE PMADRL LOW DATA_ADDR ; Load initial data address MOVLW MOVWF FSROL MOVLW HIGH DATA_ADDR ; Load initial data address MOVWF FSROH ; PMCON1,CFGS ; Not configuration space BCF BSF PMCON1.WREN ; Enable writes BSF PMCON1,LWLO ; Only Load Write Latches LOOP MOVIW FSR0++ ; Load first data byte into lower MOVWE PMDATT. ; MOVIW FSR0++ ; Load second data byte into upper MOVWF PMDATH ; Check if lower bits of address are '00000' MOVF PMADRL,W 0x1F ; Check if we're on the last of 32 addresses XORLW ANDLW 0x1F STATUS, Z ; Exit if last of 32 words, BTFSC GOTO START_WRITE ; MOVLW 55h ; Start of required write sequence: MOVWF PMCON2 ; Write 55h Required Sequence MOVLW 0AAh MOVWF PMCON2 ; Write AAh ; Set WR bit to begin write BSF PMCON1,WR NOP ; NOP instructions are forced as processor ; loads program memory write latches NOP INCF PMADRL, F ; Still loading latches Increment address GOTO LOOP ; Write next latches START_WRITE BCF PMCON1,LWLO ; No more loading latches - Actually start Flash program ; memory write MOVLW 55h ; Start of required write sequence: MOVWF PMCON2 ; Write 55h Required Sequence MOVLW 0AAh ; MOVWF PMCON2 ; Write AAh BSF PMCON1,WR ; Set WR bit to begin write NOP ; NOP instructions are forced as processor writes ; all the program memory write latches simultaneously NOP ; to program memory. ; After NOPs, the processor ; stalls until the self-write process in complete ; after write processor continues with 3rd instruction BCF PMCON1,WREN ; Disable writes BSF INTCON, GIE ; Enable interrupts

10.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

- 1. Load the starting address of the row to be modified.
- 2. Read the existing data from the row into a RAM image.
- 3. Modify the RAM image to contain the new data to be written into program memory.
- 4. Load the starting address of the row to be rewritten.
- 5. Erase the program memory row.
- 6. Load the write latches with data from the RAM image.
- 7. Initiate a programming operation.

FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



10.4 User ID, Device ID and Configuration Word Access

Instead of accessing program memory, the User ID's, Device ID/Revision ID and Configuration Words can be accessed when CFGS = 1 in the PMCON1 register. This is the region that would be pointed to by PC<15> = 1, but not all addresses are accessible. Different access may exist for reads and writes. Refer to Table 10-2.

When read access is initiated on an address outside the parameters listed in Table 10-2, the PMDATH:PMDATL register pair is cleared, reading back '0's.

TABLE 10-2:	USER ID, DEVICE ID AND CONFIGURATION WORD ACCESS (CFGS = 1)	
-------------	---	--

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8006h	Device ID/Revision ID	Yes	No
8007h-8008h	Configuration Words 1 and 2	Yes	No

EXAMPLE 10-4: CONFIGURATION WORD AND DEVICE ID ACCESS

* This code block will read 1 word of program memory at the memory address:

* PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;

* PROG_DATA_HI, PROG_DATA_LO

BANKSEL MOVLW MOVWF CLRF	PMADRL PROG_ADDR_LO PMADRL PMADRH	;;;;	Select correct Bank Store LSB of address Clear MSB of address
BSF BCF BSF NOP NOP BSF	PMCON1, CFGS INTCON, GIE PMCON1, RD INTCON, GIE	;;;;;;	Select Configuration Space Disable interrupts Initiate read Executed (See Figure 10-1) Ignored (See Figure 10-1) Restore interrupts
MOVF MOVWF MOVF MOVWF	PMDATL,W PROG_DATA_LO PMDATH,W PROG_DATA_HI	; ; ; ;	Get LSB of word Store in user location Get MSB of word Store in user location

10.5 Write Verify

It is considered good programming practice to verify that program memory writes agree with the intended value. Since program memory is stored as a full page then the stored program memory contents are compared with the intended data stored in RAM after the last write is complete.

FIGURE 10-8: FLASH PROGRAM MEMORY VERIFY FLOWCHART



10.6 Flash Program Memory Control Registers

REGISTER 10-1: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u		
			PMDA	T<7:0>					
bit 7							bit 0		
Legend:									
R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'					
u = Bit is unchanged	d	x = Bit is unknown	1	-n/n = Value at	POR and BOR/Va	lue at all other Re	esets		
'1' = Bit is set		'0' = Bit is cleared							

bit 7-0

PMDAT<7:0>: Read/write value for Least Significant bits of program memory

REGISTER 10-2: PMDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—						
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 5-0 PMDAT<13:8>: Read/write value for Most Significant bits of program memory

REGISTER 10-3: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE 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				
PMADR<7:0>											
bit 7							bit 0				
Legend:											
R = Readable bit		W = Writable bit		U = Unimpleme	ented bit, read as '	כי					

R = Readable bit	vv = vvritable bit	U = Unimplemented bit, read as U
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 PMADR<7:0>: Specifies the Least Significant bits for program memory address

REGISTER 10-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

U-1	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		
—				PMADR<14:8>					
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 Unimplemented: Read as '1'

bit 6-0 **PMADR<14:8>**: Specifies the Most Significant bits for program memory address

U	_{J-1} (1)	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q ⁽²⁾	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0		
	_	CFGS	LWLO	FREE	WRERR	WREN	WR	RD		
bit 7								bit 0		
Legen	d:									
R = Re	eadab	le bit	W = Writable b	it	U = Unimpleme	nted bit, read as	ʻ0'			
S = Bit	t can	only be set	x = Bit is unkno	own	-n/n = Value at I	POR and BOR/V	alue at all other F	Resets		
'1' = Bi	it is s	et	'0' = Bit is clear	ed	HC = Bit is clea	red by hardware				
h:4 7			ad Deed es (1)							
		Unimplement								
bit 6		CFGS: Config	uration Select bit	r ID and Dovice						
1 = Access Conliguration, Oser ID and Device ID registers $0 = Access Flash Program Memory$										
bit 5		LWLO: Load V	Vrite Latches Onl	y bit ⁽³⁾						
		1 = Only the	addressed progra	am memory write	e latch is loaded/u	updated on the n	ext WR comman	d		
		0 = The addr	essed program m	emory write latc	h is loaded/update	ed and a write of	all program memo	ory write latches		
L :1 4			tiated on the next							
DIT 4		1 = Performs	in Flash Erase Er	nable bit	WR command (ha	irdware cleared i	inon completion)			
		0 = Performs	a write operation	on the next WF	R command					
bit 3		WRERR: Prog	gram/Erase Error	Flag bit						
		1 = Condition	indicates an improper program or erase sequence attempt or termination (bit is set automatically							
		on any se	et attempt (write " ram or erase ope	1) of the WR bit	i). d normally					
bit 2		WREN: Progr	am/Frase Enable	bit	a normany.					
511 2		1 = Allows pr	ogram/erase cycl	es						
		0 = Inhibits p	rogramming/erasi	ing of program F	⁻ lash					
bit 1		WR: Write Cor	ntrol bit							
		1 = Initiates a	a program Flash p	program/erase o	peration.		n in normalata			
		The WR	ation is self-timed bit can only be se	t and the bit is c t (not cleared) ii	leared by nardwa n software	re once operatio	n is complete.			
		0 = Program/	erase operation t	o the Flash is co	omplete and inact	tive.				
bit 0		RD: Read Cor	ntrol bit							
		1 = Initiates a	a program Flash r	program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can onl						
		(not clear	red) in software.	n Flash read						
Note	1.	Unimplemented bit	read as '1'	i i iusii i cau.						
	2:	The WRERR bit is a	utomatically set b	y hardware whe	en a program mer	nory write or eras	se operation is sta	arted (WR = 1).		

REGISTER 10-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER

- 3: The LWLO bit is ignored during a program memory erase operation (FREE = 1).

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0					
	Program Memory Control Register 2											
bit 7							bit 0					
Legend:												
R = Readable	bit	W = Writable I	oit	U = Unimplemented bit, read as '0'								
S = Bit can only be set 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									

REGISTER 10-6: PMCON2: PROGRAM MEMORY CONTROL 2 REGISTER

bit 7-0 Flash Memory Unlock Pattern bits

To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the PMCON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes.

TABLE 10-3: SUMMARY OF REGISTERS ASSOCIATED WITH FLASH PROGRAM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page	
PMCON1	(1)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	91	
PMCON2	Program Memory Control Register 2									
PMADRL	PMADRL<7:0>									
PMADRH	(1)	(1) PMADRH<6:0>								
PMDATL				PMDA	FL<7:0>				90	
PMDATH	_	—		PMDATH<5:0>						
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Flash Program Memory module. Note 1: Unimplemented, read as '1'.

TABLE 10-4: SUMMARY OF CONFIGURATION WORD WITH FLASH PROGRAM MEMORY

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page	
CONFIG1	13:8	—	_	—	- CLKOUTEN		BOREN<1:0>		—	20	
CONFIGT	7:0	CP	MCLRE	PWRTE	WDTE	E<1:0>	—	FOSC	Bit 8/0 — ><1:0> <1:0>	39	
CONFICS	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	—	40	
CONFIG2	7:0	_	_	_	_	_	_	WRT	<1:0>	40	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Flash Program Memory.

11.0 I/O PORTS

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

Each port has three standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- · LATx registers (output latch)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- · WPUx (weak pull-up)

TABLE 11-1: PORT AVAILABILITY PER DEVICE

Device	PORTA	PORTB	PORTC	РОКТD	PORTE
PIC16LF1906	•	•	٠		•
PIC16LF1904/7	•	•	٠	٠	•

The data latch (LATA register) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATA register has the same affect as a write to the corresponding PORTA register. A read of the LATA register reads of the values held in the I/O port latches, while a read of the PORTA register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 11-1.

FIGURE 11-1: GENERIC I/O PORT



EXAMPLE 11-1: INITIALIZING PORTA

; This code exa	mple illustrates
-----------------	------------------

- ; initializing the PORTA register. The
- ; other ports are initialized in the same
- ; manner.

BANKSEL	PORTA	;
CLRF	PORTA	;Init PORTA
BANKSEL	LATA	;Data Latch
CLRF	LATA	;
BANKSEL	ANSELA	;
CLRF	ANSELA	;digital I/O
BANKSEL	TRISA	;
MOVLW	B'00111000'	;Set RA<5:3> as inputs
MOVWF	TRISA	;and set RA<2:0> as
		;outputs

11.1 PORTA Registers

PORTA is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 11-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). The exception is RA3, which is input only and its TRIS bit will always read as '1'. Example 11-1 shows how to initialize PORTA.

Reading the PORTA register (Register 11-1) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATA).

The TRISA register (Register 11-2) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.1.1 ANSELA REGISTER

The ANSELA register (Register 11-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELA bits default to the Analog						
	mode after Reset. To use any pins as						
	digital general purpose or peripheral						
	inputs, the corresponding ANSEL bits						
	must be initialized to '0' by user software.						

11.1.2 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-2.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as ADC, comparator and CapSense inputs, are not shown in the priority lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown in Table 11-2.

TABLE 11-2:	PORTA OUTPUT PRIORITY

Pin Name	Function Priority ⁽¹⁾
RA0	SEG12 (LCD)
	AN0
	RA0
RA1	SEG7
	AN1
	RA1
RA2	COM2
	AN2
	RA2
RA3	VREF+
	COM3
	SEG15
	AN3
	RA3
RA4	SEG4
	TOCKI
	RA4
RA5	SEG5
	AN4
	RA5
RA6	CLKOUT
	SEG1
	RA6
RA7	CLKIN
	SEG2
	RA7

Note 1: Priority listed from highest to lowest.

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
bit 7				·			bit 0
Legend:							
R = Readable b	bit	W = Writable b	bit	U = Unimpler	nented bit, read a	as '0'	
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all o		/Value at all oth	er Resets				
'1' = Bit is set		'0' = Bit is clea	red				

REGISTER 11-1: PORTA: PORTA REGISTER

bit 7-0 RA<7:0>: PORTA I/O Value bits⁽¹⁾ 1 = Port pin is ≥ VIH 0 = Port pin is ≤ VIL

REGISTER 11-2: TRISA: PORTA TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
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	TRISA<7:4>: PORTA Tri-State Control bits 1 = PORTA pin configured as an input (tri-stated) 0 = PORTA pin configured as an output
bit 3	TRISA3: RA3 Port Tri-State Control bit This bit is always '1' as RA3 is an input only
bit 2-0	TRISA<2:0>: PORTA Tri-State Control bits 1 = PORTA pin configured as an input (tri-stated) 0 = PORTA pin configured as an output

REGISTER 11-3: LATA: PORTA DATA LATCH REGISTER

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATA7 | LATA6 | LATA5 | LATA4 | LATA3 | LATA2 | LATA1 | LATA0 |
| 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 LATA<7:0>: RA<7:4> Output Latch Value bits⁽¹⁾

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

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

U-0	U-0	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	
—	_	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable bit		U = Unimplemented bit, read as '0'				
u = Bit is unch	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					

REGISTER 11-4: ANSELA: PORTA ANALOG SELECT REGISTER

bit 7-6	Unimplemented: Read as '0'
bit 5	 ANSA5: Analog Select between Analog or Digital Function on pins RA5, respectively 0 = Digital I/O. Pin is assigned to port or digital special function. 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.
bit 4	Unimplemented: Read as '0'
bit 3-0	 ANSA<3:0>: Analog Select between Analog or Digital Function on pins RA<3:0>, respectively 0 = Digital I/O. Pin is assigned to port or digital special function. 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5		ANSA3	ANSA2	ANSA1	ANSA0	96
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	95
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		130
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	95
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	95

TABLE 11-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

TABLE 11-4: SUMMARY OF CONFIGURATION WORD WITH PORTA

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
	13:8	_	_	—	_	CLKOUTEN	BORE	N<1:0>	_	20
CONFIGT	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		_	FOSC	C<1:0>	- 39

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

11.2 PORTB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 11-6). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 11-5) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATB).

The TRISB register (Register 11-6) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.2.1 ANSELB REGISTER

The ANSELB register (Register 11-8) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no effect on digital output functions. A pin with TRIS clear and ANSELB set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELB bits default to the Analog				
	mode after reset. To use any pins as				
	digital general purpose or peripheral				
	inputs, the corresponding ANSEL bits				
	must be initialized to '0' by user software.				

11.2.2 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each PORTB pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-5.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in Table 11-5.

TABLE 11-5:	PORTB OUTPUT PRIORITY
-------------	-----------------------

Pin Name	Function Priority ⁽¹⁾
RB0	SEG0 AN12 INT IOC RB0
RB1	SEG24 AN10 VLCD1 IOC RB1
RB2	SEG25 AN8 VLCD2 IOC RB2
RB3	SEG26 AN9 VLCD3 IOC RB3
RB4	COM0 AN11 IOC RB4
RB5	COM1 AN13 IOC RB5
RB6	ICDCLK SEG14 IOC RB6
RB7	ICDDAT SEG13 IOC RB7

Note 1: Priority listed from highest to lowest.

REGISTER 11-5: PORTB: PORTB REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	
bit 7							bit 0	
Legend:								
R = Readable b	oit	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 clea	ared					

bit 7-0 **RB<7:0>**: PORTB General Purpose I/O Pin bits⁽¹⁾ 1 = Port pin is ≥ VIH 0 = Port pin is ≤ VIL

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

REGISTER 11-6: TRISB: PORTB TRI-STATE REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 |
| 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

TRISB<7:0>: PORTB Tri-State Control bits

1 = PORTB pin configured as an input (tri-stated)

0 = PORTB pin configured as an output

REGISTER 11-7: LATB: PORTB DATA LATCH REGISTER

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATB7 | LATB6 | LATB5 | LATB4 | LATB3 | LATB2 | LATB1 | LATB0 |
| 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 LATB<7:0>: PORTB Output Latch Value bits⁽¹⁾

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

REGISTER 11-8: ANSELB: PORTB ANALOG S	ELECT REGISTER
---------------------------------------	----------------

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1		
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0		
bit 7							bit 0		
Legend:									
R = Readable bi	it	W = Writable bi	t	U = Unimplemented bit, read as '0'					
u = Bit is unchar	nged	x = Bit is unkno	wn	-n/n = Value at	POR and BOR/V	alue at all other	Resets		
'1' = Bit is set		'0' = Bit is clear	ed						
bit 7-6	Unimplemente	ed: Read as '0'							
bit 5-0	ANSB<5:0>: Analog Select between Analog or Digital Function on pins RB<5:0>, respectively								

ANSB<5:0>: Analog Select between Analog or Digital Function on pins RB<5:0>, respectively
 0 = Digital I/O. Pin is assigned to port or digital special function.

1 = Analog input. Pin is assigned to port of digital special function.
 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.

REGISTER 11-9: WPUB: WEAK PULL-UP PORTB REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 |
| 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

WPUB<7:0>: Weak Pull-up Register bits

1 = Pull-up enabled

0 = Pull-up disabled

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.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	99
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	98
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	98
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	98
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	99

TABLE 11-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTB.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

11.3 PORTC Registers

PORTC is an 8-bit wide bidirectional port. The corresponding data direction register is TRISC (Register 11-6). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 11-5) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATC).

The TRISC register (Register 11-6) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.3.1 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-7.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority. Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in Table 11-7.

Pin Name	Function Priority ⁽¹⁾
RC0	T1OSO T1CKI RC0
RC1	T1OSI RC1
RC2	SEG2 RC2
RC3	SEG6 RC3
RC4	SEG11 T1G RC4
RC5	SEG10 RC5
RC6	SEG9 RC6 TX/CK
RC7	SEG8 RC7 RX/DT

TABLE 11-7: PORTC OUTPUT PRIORITY

Note 1: Priority listed from highest to lowest.

REGISTER 11-10: PORTC: PORTC REGISTER

				-	1011 / 00	1\/ VV-//U	
RCO	RC5	RC4	RC3	RC2	RC1	RC0	
						bit 0	
	W = Writable I	bit	U = Unimplemented bit, read as '0'				
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at al			R/Value at all o	ther Resets			
	'0' = Bit is clea	ared					
	led	W = Writable I led x = Bit is unkn '0' = Bit is clea	W = Writable bit ed x = Bit is unknown '0' = Bit is cleared	W = Writable bit U = Unimpler led x = Bit is unknown -n/n = Value a '0' = Bit is cleared	W = Writable bit U = Unimplemented bit, read led x = Bit is unknown -n/n = Value at POR and BOF '0' = Bit is cleared	W = Writable bit U = Unimplemented bit, read as '0' led x = Bit is unknown -n/n = Value at POR and BOR/Value at all o '0' = Bit is cleared	

bit 7-0 RC<7:0>: PORTC General Purpose I/O Pin bits⁽¹⁾ 1 = Port pin is ≥ VIH 0 = Port pin is ≤ VIL

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

REGISTER 11-11: TRISC: PORTC TRI-STATE REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| 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

TRISC<7:0>: PORTC Tri-State Control bits

1 = PORTC pin configured as an input (tri-stated)

0 = PORTC pin configured as an output

REGISTER 11-12: LATC: PORTC DATA LATCH REGISTER

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| 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 LATC<7:0>: PORTC Output Latch Value bits⁽¹⁾

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

PIC16LF1904/6/7

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	98
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	98
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	98

TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTC.

11.4 PORTD Registers (PIC16LF1904/7 only)

PORTD is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISD (Register 11-14). Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTD register (Register 11-13) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATD).

The TRISD register (Register 11-14) controls the PORTD pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISD register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.4.1 PORTD FUNCTIONS AND OUTPUT PRIORITIES

Each PORTD pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-9.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in Table 11-9.

Pin Name	Function Priority ⁽¹⁾
RD0	RD0
RD1	RD1
RD2	RD2
RD3	RD3
RD4	RD4
RD5	RD5
RD6	RD6
RD7	RD7

TABLE 11-9: PORTD OUTPUT PRIORITY

Note 1: Priority listed from highest to lowest.

REGISTER 11-13: PORTD: PORTD REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	
bit 7							bit 0	
Legend:								
R = Readable b	oit	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 clea	ared					

bit 7-0 **RD<7:0>**: PORTD General Purpose I/O Pin bits⁽¹⁾ 1 = Port pin is ≥ VIH 0 = Port pin is ≤ VIL

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

REGISTER 11-14: TRISD: PORTD TRI-STATE REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD5 | TRISD5 | TRISD5 | TRISD4 |
| 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

TRISD<7:0>: PORTD Tri-State Control bits

1 = PORTD pin configured as an input (tri-stated)

0 = PORTD pin configured as an output

REGISTER 11-15: LATD: PORTB DATA LATCH REGISTER

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATD7 | LATD6 | LATD5 | LATD4 | LATD3 | LATD2 | LATD1 | LATD0 |
| 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 LATD<7:0>: PORTD Output Latch Value bits⁽¹⁾

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

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	104
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	104
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	104

TABLE 11-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD⁽¹⁾

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTD. Note 1: PIC16LF1904/7 only.

11.5 PORTE Registers

 $\frac{\text{RE3}}{\text{MCLR}}$ is input only, and also functions as $\overline{\text{MCLR}}$. The $\overline{\text{MCLR}}$ feature can be disabled via a configuration fuse. RE3 also supplies the programming voltage. The TRIS bit for RE3 (TRISE3) always reads '1'.

REGISTER 11-16: PORTE: PORTE REGISTER

U-0 U-0 U-0 U-0 U-0 U-0 U-0 R-x/u RE2⁽¹⁾ RE1⁽¹⁾ RE0⁽¹⁾ RE3 ___ _ 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 Unimplemented: Read as '0'

bit 3-0 **RE<3:0>**: PORTE Input Pin bit⁽¹⁾

- 1 = Port pin is > VIH
- 0 = Port pin is < VIL
- 2: RE<2:0> are not implemented on the PIC16LF1906. Read as '0'. Writes to RE<2:0> are actually written to the corresponding LATE register. Reads from the PORTE register is the return of actual I/O pin values.

REGISTER 11-17: TRISE: PORTE TRI-STATE REGISTER

U-0	U-0	U-0	U-0	U-1 ⁽¹⁾	R/W-1/1 ⁽²⁾	R/W-1/1 ⁽²⁾	R/W-1/1 ⁽²⁾
—	—	_	—	_	TRISE2	TRISE1	TRISE0
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 Unimplemented: Read as '0'

bit 3 Unimplemented: Read as '1'

- bit 2-0 TRISE<2:0>: PORTE Tri-State Control bits⁽²⁾
 - 1 = Port output driver is disabled
 - 0 = Port output driver is enabled

Note 1: Unimplemented, read as '1'.

2: TRISE<2:0> are not implemented on the PIC16LF1906. Read as '0'.

No output priorities, RE3 is an input only pin.

REGISTER 11-18: LATE: PORTE DATA LATCH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	_	_	_	_	LATE2 ⁽²⁾	LATE1 ⁽²⁾	LATE0 ⁽²⁾
bit 7							bit 0
Logond							

Le	g	e	n	d	:

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 7-3 Unimplemented: Read as '0'

bit 2-0 LATE<2:0>: PORTE Output Latch Value bits⁽¹⁾

- Note 1: Writes to PORTE are actually written to the corresponding LATE register. Reads from the PORTE register is return of actual I/O pin values.
 - 2: LATE<2:0> are not implemented on the PIC16LF1906. Read as '0'.

REGISTER 11-19: ANSELE: PORTE ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—	ANSE2 ⁽²⁾	ANSE1 ⁽²⁾	ANSE0 ⁽²⁾
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 2-0 ANSE<2:0>: Analog Select between Analog or Digital Function on pins RE<2:0>, respectively

- 0 = Digital I/O. Pin is assigned to port or digital special function.
- 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.
- Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.
 - 2: ANSE<2:0> are not implemented on the PIC16LF1906. Read as '0'.

REGISTER 11-20: WPUE: WEAK PULL-UP PORTE REGISTER

U-0	U-0	U-0	U-0	R/W-1/1	U-0	U-0	U-0		
—	—	—	—	WPUE3	—	_	—		
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			nown	-n/n = Value at POR and BOR/Value at all other Resets					
'1' = Bit is set		'0' = Bit is clea	ared						

bit 7-4	Unimplemented: Read as '0'
bit 3	WPUE3: Weak Pull-up Register bit
	1 = Pull-up enabled
	0 = Pull-up disabled

bit 2-0 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.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS<4:0>					GO/DONE	ADON	121
ANSELE	—	_	_	_	_	ANSE2 ⁽²⁾	ANSE1 ⁽²⁾	ANSE0 ⁽²⁾	99
LATE	—	_	_		—	LATE2 ⁽²⁾	LATE1 ⁽²⁾	LATE0 ⁽²⁾	106
PORTE	—	_	_	_	RE3	RE2 ⁽²⁾	RE1 ⁽²⁾	RE0 ⁽²⁾	106
TRISE	_	_	_	_	_(1)	TRISE2(2)	TRISE1 ⁽²⁾	TRISE0 ⁽²⁾	106
WPUE	—	_	_	_	WPUE3	—	_	_	108

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTE.

Note 1: Unimplemented, read as '1'.

2: PIC16LF1904/7 only.
12.0 INTERRUPT-ON-CHANGE

The PORTB pins can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual PORTB pin, or combination of PORTB pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- · Rising and falling edge detection
- Individual pin interrupt flags

Figure 12-1 is a block diagram of the IOC module.

12.1 Enabling the Module

To allow individual PORTB pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

12.2 Individual Pin Configuration

For each PORTB pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated IOCBPx bit of the IOCBP register is set. To enable a pin to detect a falling edge, the associated IOCBNx bit of the IOCBN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both the IOCBPx bit and the IOCBNx bit of the IOCBP and IOCBN registers, respectively.

12.3 Interrupt Flags

The IOCBFx bits located in the IOCBF register are status flags that correspond to the interrupt-on-change pins of PORTB. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCBFx bits.

12.4 Clearing Interrupt Flags

The individual status flags, (IOCBFx bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 12-1:

MOVLW 0xff XORWF IOCBF, W ANDWF IOCBF, F

12.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCBF register will be updated prior to the first instruction executed out of Sleep.

FIGURE 12-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM



12.6 Interrupt-On-Change Registers

REGISTER 12-1: IOCBP: INTERRUPT-ON-CHANGE POSITIVE EDGE 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		
IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0		
bit 7							bit 0		
Legend:									
R = Readable bit W = Writable bit			bit	U = Unimplemented bit, read as '0'					
u = Bit is unchanged x =		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 7-0

IOCBP<7:0>: Interrupt-on-Change Positive Edge Enable bits

- 1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
- 0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 12-2: IOCBN: INTERRUPT-ON-CHANGE NEGATIVE EDGE 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	
IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	
bit 7 bit								

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 IOCBN<7:0>: Interrupt-on-Change Negative Edge Enable bits

- 1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.
- 0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 12-3: IOCBF: INTERRUPT-ON-CHANGE FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	
IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	
bit 7 bit (

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 7-0

IOCBF<7:0>: Interrupt-on-Change 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.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—		ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	99
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	110
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	110
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	110
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	98

TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-	CHANGE
--	--------

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

13.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference (FVR) is a stable voltage reference, independent of VDD, with 1.024V or 2.048V selectable output levels. The output of the FVR can be configured as the FVR input channel on the ADC.

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

13.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC is routed through two independent programmable gain amplifiers. Each amplifier can be configured to amplify the reference voltage by 1x or 2x, to produce the two possible voltage levels.

The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 15.0** "**Analog-to-Digital Converter** (**ADC**) **Module**" for additional information.

13.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 22.0** "**Electrical Specifications**" for the minimum delay requirement.

FIGURE 13-1: VOLTAGE REFERENCE BLOCK DIAGRAM



TABLE 13-1:	PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

Peripheral	Conditions	Description
HFINTOSC	FOSC<2:0> = 100 and IRCF<3:0> = 000x	INTOSC is active and device is not in Sleep.
	BOREN<1:0> = 11	BOR always enabled.
BOR	BOREN<1:0> = 10 and BORFS = 1	BOR disabled in Sleep mode, BOR Fast Start enabled.
	BOREN<1:0> = 01 and BORFS = 1	BOR under software control, BOR Fast Start enabled.

13.3 FVR Control Registers

|--|

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0				
FVREN	FVRRDY ⁽¹⁾	TSEN	TSRNG	_	—	ADFVI	R<1:0>				
bit 7							bit 0				
Legend:											
R = Readabl	le bit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'					
u = Bit is und	changed	x = Bit is unkr	nown	-n/n = Value	at POR and BOI	R/Value at all o	ther Resets				
'1' = Bit is se	et	'0' = Bit is cle	ared	q = Value de	pends on conditi	ion					
bit 7	FVREN: Fixe	d Voltage Refe	rence Enable	bit							
	0 = Fixed Vo	Itage Referenc	e is disabled								
	1 = Fixed Vol	Itage Referenc	e is enabled	(4)							
bit 6	FVRRDY: Fix	-VRRDY: Fixed Voltage Reference Ready Flag bit ⁽¹⁾									
	0 = Fixed Vol	0 = Fixed Voltage Reference output is not ready or not enabled									
		lage Relefenc		ady for use							
DIT 5	ISEN: lempe	erature Indicator Enable bit									
	1 = Tempera	0 = remperature indicator is disabled 1 = Temperature Indicator is enabled									
hit 4	TSRNG. Tem	perature Indica	tor Range Se	lection hit							
	0 = VOUT = V	$0 = V_{0} = 1 = V_{0} = 2V_{T} (I_{0} = R_{0})$									
	1 = VOUT = V	1 = VOUT = VDD - 4VT (High Range)									
bit 3-2	Unimplemen	ted: Read as '	0'								
bit 1-0	ADFVR<1:0>	: ADC Fixed V	oltage Refere	nce Selection I	bit						
	00 = ADC Fix	00 = ADC Fixed Voltage Reference Peripheral output is off.									
	01 = ADC Fix	ed Voltage Re	ference Peripl	heral output is	1x (1.024V)						
	10 = ADC Fix	ed Voltage Re	ference Peripl	heral output is	2x (2.048V) ⁽²⁾						
	11 = Reserve	D									
Note 1: F	VRRDY will output	ut the true state	e of the band	gap.							

^{2:} Fixed Voltage Reference output cannot exceed VDD.

TABLE 13-2: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

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	—		ADFVR1	ADFVR0	113

Legend: Shaded cells are not used with the Fixed Voltage Reference.

14.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between of -40° C and $+85^{\circ}$ C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, *"Use and Calibration of the Internal Temperature Indicator"* (DS01333) for more details regarding the calibration process.

14.1 Circuit Operation

Figure 14-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 14-1 describes the output characteristics of the temperature indicator.

EQUATION 14-1: VOUT RANGES

High Range: VOUT = VDD - 4VT

Low Range: VOUT = VDD - 2VT

The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 13.0 "Fixed Voltage Reference (FVR)**" for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher VDD is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 14-1: TEMPERATURE CIRCUIT DIAGRAM



14.2 Minimum Operating VDD vs. Minimum Sensing Temperature

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, VDD, must be high enough to ensure that the temperature circuit is correctly biased.

Table 14-1 shows the recommended minimum VDD vs.range setting.

TABLE 14-1: RECOMMENDED VDD VS. RANGE

Min. VDD, TSRNG = 1	Min. VDD, TSRNG = 0
3.6V	1.8V

14.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 15.0** "**Analog-to-Digital Converter (ADC) Module**" for detailed information.

14.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 μ s after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 μ s between sequential conversions of the temperature indicator output.

15.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 15-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

FIGURE 15-1: ADC BLOCK DIAGRAM

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.



15.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- · Channel selection
- · ADC voltage reference selection
- ADC conversion clock source
- · Interrupt control
- Result formatting

15.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 11.0 "I/O Ports**" for more information.

Note:	Analog voltages on any pin that is defined									
	as a digital input may cause the input									
	buffer to conduct excess current.									

15.1.2 CHANNEL SELECTION

There are up to 11 channel selections available:

- AN<13:0> pins
- Temperature Indicator
- FVR (Fixed Voltage Reference) Output

Refer to Section 13.0 "Fixed Voltage Reference (FVR)" and Section 14.0 "Temperature Indicator Module" for more information on these channel selections.

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 15.2 "ADC Operation"** for more information.

15.1.3 ADC VOLTAGE REFERENCE

The ADPREF bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- Vdd

15.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- · FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 15-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the A/D conversion requirements in **Section 22.0 "Electrical Specifications"** for more information. Table 15-1 gives examples of appropriate ADC clock selections.

Note: Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

TABLE 15-1:	ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES
-------------	---

ADC Clock	Period (TAD)	Device Frequency (Fosc)					
ADC Clock Source	ADCS<2:0>	20 MHz	20 MHz 16 MHz		4 MHz	1 MHz	
Fosc/2	000	100 ns ⁽²⁾	125 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 μs	
Fosc/4	100	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μs	4.0 μs	
Fosc/8	001	400 ns ⁽²⁾	0.5 μs ⁽²⁾	1.0 μs	2.0 μs	8.0 μs ⁽³⁾	
Fosc/16	101	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs ⁽³⁾	
Fosc/32	010	1.6 μs	2.0 μs	4.0 μs	8.0 μs ⁽³⁾	32.0 μs ⁽³⁾	
Fosc/64	110	3.2 μs	4.0 μs	8.0 μs ⁽³⁾	16.0 μs ⁽³⁾	64.0 μs ⁽³⁾	
FRC	x11	1.0-6.0 μs ^(1,4)					

Legend: Shaded cells are outside of recommended range.

- **Note 1:** The FRC source has a typical TAD time of 1.6 μ s for VDD.
 - 2: These values violate the minimum required TAD time.
 - 3: For faster conversion times, the selection of another clock source is recommended.
 - 4: The ADC clock period (TAD) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock FOSC. However, the FRC clock source must be used when conversions are to be performed with the device in Sleep mode.





15.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

Note 1:	The ADIF bit is set at the completion of
	every conversion, regardless of whether or not the ADC interrupt is enabled.

2: The ADC operates during Sleep only when the FRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

15.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 15-3 shows the two output formats.

FIGURE 15-3: 10-BIT A/D CONVERSION RESULT FORMAT



15.2 ADC Operation

15.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/DONE bit should not be set in the
	same instruction that turns on the ADC.
	Refer to Section 15.2.5 "A/D Conversion
	Procedure".

15.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

15.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

Note: A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

15.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

15.2.5 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

- 1. Configure Port:
 - Disable pin output driver (Refer to the TRIS register)
 - Configure pin as analog (Refer to the ANSEL register)
- 2. Configure the ADC module:
 - Select ADC conversion clock
 - Configure voltage reference
 - Select ADC input channel
 - Turn on ADC module
- 3. Configure ADC interrupt (optional):
 - Clear ADC interrupt flag
 - · Enable ADC interrupt
 - Enable peripheral interrupt
 - Enable global interrupt⁽¹⁾
- 4. Wait the required acquisition time⁽²⁾.
- 5. Start conversion by setting the GO/DONE bit.
- 6. Wait for ADC conversion to complete by one of the following:
 - Polling the GO/DONE bit
 - Waiting for the ADC interrupt (interrupts enabled)
- 7. Read ADC Result.
- 8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to Section 15.3 "A/D Acquisition Requirements".

EXAMPLE 15-1: A/D CONVERSION

;This code block configures the ADC ; for polling, Vdd and Vss references, Frc ;clock and ANO input. ;Conversion start & polling for completion ; are included. BANKSEL ADCON1 ; B'11110000' ;Right justify, Frc MOVLW ;clock MOVWF ADCON1 ;Vdd and Vss Vref BANKSEL TRISA ; BSF TRISA,0 ;Set RA0 to input BANKSEL ANSEL ; BSF ANSEL,0 ;Set RA0 to analog BANKSEL ADCON0 B'00000001' ;Select channel ANO MOVLW ;Turn ADC On MOVWE ADCON0 SampleTime ; Acquisiton delay CALL ADCON0, ADGO ; Start conversion BSF BTFSC ADCON0, ADGO ; Is conversion done? GOTO \$-1 ;No, test again ADRESH ; BANKSEL ADRESH,W ;Read upper 2 bits MOVF MOVWF RESULTHI ;store in GPR space BANKSEL ADRESL ; ADRESL,W MOVF ;Read lower 8 bits MOVWE RESULTLO ;Store in GPR space

15.2.6 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

REGISTER 15-1: ADCON0: A/D CONTROL REGISTER 0

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
—			CHS<4:0>			GO/DONE	ADON
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		Unimplemented: Read as '0'
bit 6-2	2	CHS<4:0>: Analog Channel Select bits
		00000 = ANO
		00001 = AN1
		00010 = AN2
		00011 = AN3
		00100 = AN4
		$00101 = AN5^{(3)}$
		$00110 = AN6^{(3)}$
		$00111 = AN7^{(3)}$
		01000 = AN8
		01001 = AN9
		01010 = AN10
		01011 = AN11
		01100 = AN12
		01101 = AN13 01110 = Recorded No channel connected
		•
		•
		11100 = Reserved. No channel connected.
		11101 = Temperature Indicator ⁽²⁾
		11110 = Reserved. No channel connected.
		11111 = FVR (Fixed Voltage Reference) Buffer 1 Output ⁽¹⁾
bit 1		GO/DONE: A/D Conversion Status bit
		1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
		This bit is automatically cleared by hardware when the A/D conversion has completed.
		0 = A/D conversion completed/not in progress
bit 0		ADON: ADC Enable bit
		1 = ADC is enabled
		0 = ADC is disabled and consumes no operating current
Note	1:	See Section 13.0 "Fixed Voltage Reference (FVR)" for more information.
	2:	See Section 14.0 "Temperature Indicator Module" for more information.
	3:	ADC channel is reserved on the PIC16LF1906 28-pin device.

REGISTER 15-2: ADCON1: A/D CONTROL REGISTER 1										
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0			
ADFM	ADFM ADCS<2:0>			_	_	ADPRE	F<1:0>			
bit 7							bit 0			
Legend:										
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, rea	d as '0'				
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	OR/Value at all	other Resets			
'1' = Bit is set		'0' = Bit is cle	ared							
bit 7 bit 6-4	ADFM: A/D 1 = Right ju loaded. 0 = Left jus loaded. ADCS<2:0> 000 = Fosc 001 = Fosc 010 = Fosc	Result Format S stified. Six Mosi tified. Six Least : A/D Conversic /2 /8	Select bit t Significant bit Significant bit n Clock Selec	ts of ADRESH s of ADRESL a t bits	are set to '0' v are set to '0' v	when the conve	ersion result is			
	011 = FRC (100 = Fosc 101 = Fosc 110 = Fosc 111 = FRC (clock supplied f /4 /16 /64 clock supplied f	rom a dedicate	ed RC oscillator	r) r)					
bit 3-2	Unimpleme	nted: Read as '	0'							
bit 1-0	ADPREF<1: 00 = VREF+ 01 = Reserv 10 = VREF+ 11 = Reserv	:0>: A/D Positive is connected to ved is connected to ved	e Voltage Refe VDD external VREF	rence Configui + pin ⁽¹⁾	ration bits					
				.						

Note 1: When selecting the FVR or the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 22.0 "Electrical Specifications"** for details.

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
			ADRE	S<9:2>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	oit	U = Unimpler	nented bit, read	d as '0'	
u = Bit is unch	anged	x = Bit is unkn	own	-n/n = Value a	at POR and BO	R/Value at all	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				

REGISTER 15-3: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

bit 7-0 **ADRES<9:2>**: ADC Result Register bits Upper eight bits of 10-bit conversion result

REGISTER 15-4: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>		—	—	_	—	—	—
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 7-6 ADRES<1:0>: ADC Result Register bits Lower two bits of 10-bit conversion result bit 5-0 Reserved: Do not use.

REGISTER 15-5: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	_	—	—	ADRE	S<9:8>
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	l as '0'	
u = Bit is uncha	anged	x = Bit is unkr	iown	-n/n = Value a	at POR and BO	R/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				

bit 7-2 Reserved: Do not use.

bit 1-0 ADRES<9:8>: ADC Result Register bits Upper two bits of 10-bit conversion result

REGISTER 15-6: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | ADRES | S<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 ADRES<7:0>: ADC Result Register bits Lower eight bits of 10-bit conversion result

15.3 A/D 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 15-4. 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 15-4. 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 A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 15-1 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 15-1: 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:

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - I}\right) = V_{CHOLD} \qquad ;[1] V_{CHOLD} charged to within 1/2 lsb$$

$$V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{CHOLD} \qquad ;[2] V_{CHOLD} charge response to V_{APPLIED} V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - I}\right) \qquad ;combining [1] and [2]$$

Note: Where n = number of bits of the ADC.

Solving for TC:

$$TC = -C_{HOLD}(R_{IC} + R_{SS} + R_{S}) \ln(1/2047)$$

= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885)
= 1.715\mus

Therefore:

$$TACQ = 2\mu s + 1.715\mu s + [(50^{\circ}C - 25^{\circ}C)(0.05\mu s/^{\circ}C)]$$

= 4.96\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.

PIC16LF1904/6/7









Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	121
ADCON1	ADFM	ADCS2	ADCS1	ADCS0	-	—	ADPREF1	ADPREF0	122
ADRESH	A/D Result I	Register High	1						123, 124
ADRESL	A/D Result I	Register Low							123, 124
ANSELA	—	_	ANSA5	-	ANSA3	ANSA2	ANSA1	ANSA0	96
ANSELB	—	_	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	99
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE	TXIE	-	—	—	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF	TXIF	_	—	—	TMR1IF	68
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	95
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	98
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR1	ADFVR0	113

TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for ADC module.

16.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- · Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 16-1 is a block diagram of the Timer0 module.

16.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

16.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the TMR0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note: The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

FIGURE 16-1: BLOCK DIAGRAM OF THE TIMER0

16.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge is determined by the TMR0SE bit in the OPTION_REG register.



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 pro-
	cessor from Sleep since the timer is fro-
	zen 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 22.0** "**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.

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>	
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplei	mented bit, read	d as '0'	
u = Bit is unch	anged	x = Bit is unk	nown	-n/n = Value	at POR and BO	R/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is cle	eared				
bit 7	WPUEN: We	ak Pull-up Ena	ıble bit				
	1 = All weak	pull-ups are dis	sabled (except	MCLR, if it is	enabled)		
	0 = vveak pu	II-ups are enab	ied by individu	ial wpux latch	values		
DIT 6	INTEDG: Inte	errupt Edge Se					
	\perp = Interrupt	on rising edge	of INT pin				
bit 5		mer0 Clock Sol	urce Select hit				
bit 5	1 = Transition	n on TOCKI nin					
	0 = Internal ii	nstruction cycle	e clock (Fosc/4	4)			
bit 4	TMR0SE: Tir	mer0 Source E	dge Select bit				
	1 = Incremen	nt on high-to-low	w transition on	T0CKI pin			
	0 = Incremen	nt on low-to-hig	h transition on	T0CKI pin			
bit 3	PSA: Presca	ller Assignmen	t bit				
	1 = Prescale	r is not assigne	ed to the Timer	0 module			
	0 = Prescale	r is assigned to	the limeru m	odule			
bit 2-0	PS<2:0>: Pre	escaler Rate S	elect dits				
	Bit	Value Timer0	Rate				
	(000 1:2	2				
	(001 1:4 010 1:8	F 3				
	(011 1:1	6				
		100 1:3	32				

REGISTER 16-1: OPTION_REG: OPTION REGISTER

TABLE 16-1:	SUMMARY	OF REGISTERS	ASSOCIATED	WITH TIMER0

1:64

1:128

1:256

101

110 111

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	65
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		130		
TMR0	Timer0 Module Register							128*	
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	95

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

* Page provides register information.

17.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- · Programmable internal or external clock source
- · 2-bit prescaler
- · Dedicated 32 kHz oscillator circuit
- Multiple Timer1 gate (count enable) sources
- · Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- · Selectable Gate Source Polarity
- · Gate Toggle mode
- Gate Single-pulse mode
- Gate Value Status
- Gate Event Interrupt





Figure 17-1 is a block diagram of the Timer1 module.

17.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 17-1 displays the Timer1 enable selections.

TABLE 17-1:	TIMER1 ENABLE
	SELECTIONS

TMR10N	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	Always On
1	1	Count Enabled

17.2 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Table 17-2 displays the clock source selections.

17.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of Fosc as determined by the Timer1 prescaler.

When the Fosc internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.

The following asynchronous source may be used:

 Asynchronous event on the T1G pin to Timer1 gate

17.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input T1CKI or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

- Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:
 - · Timer1 enabled after POR
 - Write to TMR1H or TMR1L
 - Timer1 is disabled
 - Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON = 1) when T1CKI is low.

TMR1CS1	TMR1CS0	T1OSCEN	Clock Source
0	0	x	Instruction Clock (Fosc/4)
0	1	x	System Clock (FOSC)
1	0	0	External Clocking on T1CKI Pin
1	0	1	Osc. Circuit on T1OSI/T1OSO Pins
1	1	x	Reserved

TABLE 17-2: CLOCK SOURCE SELECTIONS

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 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO. This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

Note: The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to using Timer1. A suitable delay similar to the OST delay can be implemented in software by clearing the TMR1IF bit then presetting the TMR1H:TMR1L register pair to FC00h. The TMR1IF flag will be set when 1024 clock cycles have elapsed, thereby indicating that the oscillator is running and reasonably stable.

17.5 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 17.5.1 "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.

17.5.1 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.

17.6 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.

17.6.1 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
1	1	1	Counts

17.6.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS 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
00	Timer1 Gate Pin
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)

17.6.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

17.6.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

17.6.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 17-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note:	Enabling Toggle mode at the same time
	as changing the gate polarity may result in
	indeterminate operation.

17.6.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/DONE bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/DONE bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/DONE bit is once again set in software. See Figure 17-5 for timing details.

If the Single Pulse Gate mode is disabled by clearing the T1GSPM bit in the T1GCON register, the T1GGO/DONE bit should also be cleared.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 gate source to be measured. See Figure 17-6 for timing details.

17.6.5 TIMER1 GATE VALUE STATUS

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

17.6.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

17.7 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- · PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note: The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

17.8 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- · PEIE bit of the INTCON register must be set
- TISYNC bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured
- T1OSCEN bit of the T1CON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

Timer1 oscillator will continue to operate in Sleep regardless of the T1SYNC bit setting.



PIC16LF1904/6/7



FIGURE 17-4: TIMER1 GATE TOGGLE MODE



FIGURE 17-5:	TIMER1 GATE SINGLE-PL	PULSE MODE
		_
IMRIGE		
T1GPOL		
T1GSPM		Cleared by bardware on
T1GG <u>O/</u> DONE	← Set by software Counting enabled on rising edge of T1G	falling edge of T1GVAL
T1G_IN		
T1CKI		
T1GVAL		
Timer1	<u>N</u>	N + 1 N + 2
TMR1GIF	— Cleared by software	Set by hardware on falling edge of T1GVAL

PIC16LF1904/6/7

FIGURE 17-6:	TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE
TMR1GE	
T1GPOL	
T1GSPM	
T1GTM	
T1GG <u>O/</u> DONE	Cleared by hardware on <u>Counting enabled on</u> <u>Figure of T1C</u>
T1G_IN	
т1СКІ	
T1GVAL	
Timer1	N N + 1 N + 2 N + 3 N + 4
TMR1GIF	- Cleared by software falling edge of T1GVAL

17.9 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 17-1, is used to control Timer1 and select the various features of the Timer1 module.

REGISTER 17-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u		
TMR1CS<1:0>		T1CKP	S<1:0>	T1OSCEN	T1SYNC	—	TMR10N		
bit 7		•					bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimplen	nented bit, read	l as '0'			
u = Bit is unch	anged	x = Bit is unkn	iown	-n/n = Value at POR and BOR/Value at all other					
'1' = Bit is set		'0' = Bit is clea	ared						
bit 7-6	TMR1CS<1:0 11 = Reserver 10 = Timer1 c <u>If T10SC</u> External <u>If T10SC</u> Crystal c 01 = Timer1 c 00 = Timer1 c	b : Timer1 Cloc d clock source is p <u>CEN = 0</u> : clock from T10 <u>CEN = 1</u> : bscillator on T10 clock source is s clock source is is	ck Source Sele pin or oscillato CKI pin (on the OSI/T1OSO p system clock (instruction clo	ect bits or: e rising edge) ins (Fosc) ck (Fosc/4)					
bit 5-4	5-4 T1CKPS<1:0>: Timer1 Input Clock Prescale Select bits 11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value 00 = 1:1 Prescale value								
bit 3	T1OSCEN: L 1 = Dedicate 0 = Dedicate	P Oscillator En d Timer1 oscilla d Timer1 oscilla	able Control b ator circuit ena ator circuit dis	it abled abled					
bit 2 T1SYNC: Timer1 External Clock Input Synchronization Control bit $\frac{\text{TMR1CS} < 1:0 > = 1X}{1 = \text{ Do not synchronize external clock input}}$ $0 = \text{Synchronize external clock input with system clock (Fosc)}$									
bit 1 bit 0	TMR1CS<1:0 This bit is igno Unimplemen TMR1ON: Tin 1 = Enables 0 = Stops Tin Clears Tin	<u>> = 0X</u> ored. Timer1 us ted: Read as '(ner1 On bit Timer1 ner1 mer1 gate flip-f	ses the interna o' lop	al clock when T	MR1CS<1:0> =	- 1x.			

17.10 Timer1 Gate Control Register

The Timer1 Gate Control register (T1GCON), shown in Register 17-2, is used to control Timer1 gate.

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u		
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS	6<1:0>		
bit 7							bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimplemented 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	other Resets		
'1' = Bit is set		'0' = Bit is clea	ared	HC = Bit is cle	eared by hardw	are			
bit 7 TMR1GE: Timer1 Gate Enable bit <u>If TMR1ON = 0</u> : This bit is ignored <u>If TMR1ON = 1</u> : 1 = Timer1 counting is controlled by the Timer1 gate function									
bit 6	T1GPOL: Tim 1 = Timer1 ga	ner1 Gate Pola ate is active-hig	rity bit gh (Timer1 cou	unts when gate	is high)				
	0 = Timer1 ga	ate is active-lov	w (Timer1 cou	nts when gate i	s low)				
bit 5	T1GTM: Time 1 = Timer1 G 0 = Timer1 G Timer1 gate fl	er1 Gate Toggle ate Toggle mo ate Toggle mo ip-flop toggles	e Mode bit de is enabled de is disabled on every rising	and toggle flip- g edge.	flop is cleared				
bit 4	T1GSPM: Timer1 Gate Single-Pulse Mode bit								
	1 = Timer1 ga 0 = Timer1 ga	ate Single-Puls ate Single-Puls	e mode is ena e mode is disa	abled and is cor abled	ntrolling Timer1	gate			
bit 3	T1GGO/DON	E: Timer1 Gate	e Single-Pulse	Acquisition Sta	atus bit				
	1 = Timer1 ga 0 = Timer1 ga	ate single-pulse ate single-pulse	e acquisition is e acquisition h	s ready, waiting as completed c	for an edge or has not been	started			
bit 2	T1GVAL: Tim	er1 Gate Curre	ent State bit						
	Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L. Unaffected by Timer1 Gate Enable (TMR1GE).								
bit 1-0	T1GSS<1:0>:	: Timer1 Gate	Source Select	bits					
	00 = Timer1 gate pin 01 = Timer0 overflow output 10 = Reserved 11 = Reserved								

REGISTER 17-2: T1GCON: TIMER1 GATE CONTROL REGISTER

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	65
PIE1	TMR1GIE	ADIE	RCIE	TXIE	_	_	_	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF	TXIF	_	_	_	TMR1IF	68
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								135*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	101
T1CON	TMR1CS1	TMR1CS0	T1CKP	T1CKPS<1:0> T1OSCEN T1SYNC — TMR1ON					139
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS1	T1GSS0	140

TABLE 17-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.

* Page provides register information.

18.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

- · Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- · Address detection in 9-bit mode
- · Input buffer overrun error detection
- · Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- · Programmable clock and data polarity

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- · Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

Block diagrams of the EUSART transmitter and receiver are shown in Figure 18-1 and Figure 18-2.





FIGURE 18-2: EUSART RECEIVE BLOCK DIAGRAM



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These registers are detailed in Register 18-1, Register 18-2 and Register 18-3, respectively.

For all modes of EUSART operation, the TRIS control bits corresponding to the RX/DT and TX/CK pins should be set to '1'. The EUSART control will automatically reconfigure the pin from input to output, as needed.

When the receiver or transmitter section is not enabled then the corresponding RX/DT or TX/CK pin may be used for general purpose input and output.

18.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a VOH Mark state which represents a '1' data bit, and a VoL Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 18-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

18.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

18.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note: The TXIF transmitter interrupt flag is set when the TXEN enable bit is set.

18.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one TCY immediately following the Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

18.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDCON register. The default state of this bit is '0' which selects high true transmit Idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true Idle and data bits. The SCKP bit controls transmit data polarity only in Asynchronous mode. In Synchronous mode the SCKP bit has a different function. See Section 18.5.1.2 "Clock Polarity".

18.1.1.4 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of the TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.
18.1.1.5 TSR Status

The TRMT bit of the TXSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user needs to poll this bit to determine the TSR status.

Note:	The TSR register is not mapped in data
	memory, so it is not available to the user.

18.1.1.6 Transmitting 9-Bit Characters

The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set, the EUSART will shift 9 bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TXREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 18.1.2.7** "Address **Detection**" for more information on the Address mode.

18.1.1.7 Asynchronous Transmission Set-up:

- Initialize the SPBRGH:SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 18.4 "EUSART Baud Rate Generator (BRG)").
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- 3. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
- 4. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
- 5. Set the SCKP control bit if inverted transmit data polarity is desired.
- Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
- 7. If interrupts are desired, set the TXIE interrupt enable bit. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
- 8. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
- 9. Load 8-bit data into the TXREG register. This will start the transmission.



FIGURE 18-3: ASYNCHRONOUS TRANSMISSION

PIC16LF1904/6/7



TABLE 18-1: 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
BAUD1CON	ABDOVF	RCIDL		SCKP	BRG16		WUE	ABDEN	153
BAUD2CON	ABDOVF	RCIDL	_	SCKP	BRG16	_	WUE	ABDEN	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE ⁽¹⁾	TXIE ⁽¹⁾	—	—	-	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF ⁽¹⁾	TXIF ⁽¹⁾	—	_		TMR1IF	68
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152
SPBRGL			EUSART	Baud Rate	Generator, L	ow Byte			154*
SPBRGH			EUSART	Baud Rate	Generator, H	ligh Byte			154*
TXREG	EUSART Transmit Register								
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for asynchronous transmission.

* Page provides register information.

Note 1: PIC16LF1904/7 only.

18.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode would typically be used in RS-232 systems. The receiver block diagram is shown in Figure 18-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

18.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

Note 1: If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

If the RX/DT pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

18.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See Section 18.1.2.4 "Receive Framing Error" for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

Note: If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See Section 18.1.2.5 "Receive Overrun Error" for more information on overrun errors.

18.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting the following bits:

- RCIE interrupt enable bit of the PIE1 register
- PEIE peripheral interrupt enable bit of the INTCON register
- GIE global interrupt enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

18.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive
	FIFO have framing errors, repeated reads
	of the RCREG will not clear the FERR bit.

18.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated If a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by resetting the EUSART by clearing the SPEN bit of the RCSTA register.

18.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set, the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

18.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

18.1.2.8 Asynchronous Reception Set-up:

- Initialize the SPBRGH:SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 18.4 "EUSART Baud Rate Generator (BRG)").
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- 3. Enable the serial port by setting the SPEN bit and the RX/DT pin TRIS bit. The SYNC bit must be clear for asynchronous operation.
- 4. If interrupts are desired, set the RCIE interrupt enable bit and set the GIE and PEIE bits of the INTCON register.
- 5. If 9-bit reception is desired, set the RX9 bit.
- 6. Enable reception by setting the CREN bit.
- 7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 8. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
- 9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.
- 10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

18.1.2.9 9-bit Address Detection Mode Set-up

This mode would typically be used in RS-485 systems. To set up an asynchronous reception with address detect enable:

- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 18.4 "EUSART Baud Rate Generator (BRG)").
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- If interrupts are desired, set the RCIE interrupt enable bit and set the GIE and PEIE bits of the INTCON register.
- 5. Enable 9-bit reception by setting the RX9 bit.
- 6. Enable address detection by setting the ADDEN bit.
- 7. Enable reception by setting the CREN bit.
- 8. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 9. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
- 10. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
- 11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
- 12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	153
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE ⁽¹⁾	TXIE ⁽¹⁾	—	—	-	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF ⁽¹⁾	TXIF ⁽¹⁾	—	—	_	TMR1IF	68
RCREG			El	JSART Rec	eive Regist	er			147*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152
SPBRGL			EUSART	Baud Rate	Generator,	Low Byte			154*
SPBRGH			EUSART	Baud Rate	Generator,	High Byte			154*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	101
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151

TABLE 18-2: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for asynchronous reception.

* Page provides register information.

Note 1: PIC16LF1904/7 only.

18.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as VDD or temperature changes, and this directly affects the asynchronous baud rate. The Auto-Baud Detect feature (see Section 18.4.1 "Auto-Baud Detect") can be used to compensate for changes in the INTOSC frequency.

There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

18.3 Register Definitions: EUSART Control

REGISTER 18-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0			
CSRC	TX9 TXEN ⁽¹⁾ SYNC		SENDB	BRGH	TRMT	TX9D				
bit 7 bit 0										

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

bit 7		CSRC: Clock Source Select bit
		Asynchronous mode:
		Don't care
		Synchronous mode:
		1 = Master mode (clock generated internally from BRG)
		0 = Slave mode (clock from external source)
bit 6		TX9: 9-bit Transmit Enable bit
		1 = Selects 8-bit transmission $ 0 = Selects 8-bit transmission$
bit 5		TXEN: Transmit Enable hit(1)
bit 5		1 = Transmit enabled
		0 = Transmit disabled
bit 4		SYNC: EUSART Mode Select bit
		1 = Synchronous mode
		0 = Asynchronous mode
bit 3		SENDB: Send Break Character bit
		Asynchronous mode:
		 1 = Send Sync Break on next transmission (cleared by hardware upon completion) 0 = Sync Break transmission completed
		Synchronous mode:
		Don't care
bit 2		BRGH: High Baud Rate Select bit
		Asynchronous mode:
		1 = High speed
		0 = Low speed
		Linused in this mode
bit 1		TRMT: Transmit Shift Register Status bit
DIC I		1 = TSR empty
		0 = TSR full
bit 0		TX9D: Ninth bit of Transmit Data
		Can be address/data bit or a parity bit.
Note	1:	SREN/CREN overrides TXEN in Sync mode.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimple	mented bit, read	l as '0'	
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	eared	x = Bit is unkr	nown
bit 7	SPEN: Serial 1 = Serial po 0 = Serial po	Port Enable bi rt enabled (cor rt disabled (hel	t ifigures RX/D d in Reset)	T and TX/CK p	oins as serial po	rt pins)	
bit 6	RX9: 9-bit Re	ceive Enable b	, it				
	1 = Selects 9 0 = Selects 8	-bit reception 3-bit reception					
bit 5	SREN: Single Asynchronous Don't care Synchronous 1 = Enables 0 = Disables This bit is clea Synchronous Don't care	e Receive Enat s mode: <u>mode – Maste</u> single receive single receive ared after recej <u>mode – Slave</u>	ole bit <u>r</u> : otion is comp	lete.			
bit 4	CREN: Contir	nuous Receive	Enable bit				
	Asynchronous 1 = Enables 0 = Disables Synchronous 1 = Enables 0 = Disables	<u>s mode</u> : receiver receiver <u>mode</u> : continuous rec continuous rec	eive until ena	ble bit CREN is	s cleared (CREN	N overrides SR	EN)
bit 3	ADDEN: Add Asynchronous 1 = Enables a 0 = Disables Asynchronous Don't care	ress Detect En <u>s mode 9-bit (F</u> address detect address detec <u>s mode 8-bit (F</u>	able bit (X9 = 1): ion, enable in tion, all bytes (X9 = 0):	iterrupt and loa are received a	d the receive bund ninth bit can	uffer when RSR be used as pa	R<8> is set rity bit
bit 2	FERR: Framin 1 = Framing 0 = No framin	ng Error bit error (can be u ng error	pdated by rea	ading RCREG	register and rec	eive next valid	byte)
bit 1	OERR: Overr 1 = Overrun 0 = No overru	un Error bit error (can be c un error	leared by clea	aring bit CREN)		
bit 0	RX9D: Ninth I This can be a	bit of Received ddress/data bit	Data or a parity bi	it and must be	calculated by us	er firmware.	

REGISTER 18-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0				
ABDOVF	RCIDL		SCKP	BRG16		WUE	ABDEN				
bit 7				<u> </u>			bit 0				
Legend:											
R = Readable	bit	W = Writable	W = Writable bit U = Unimplemented bit, read as '0'								
u = Bit is uncha	anged	x = Bit is unkr	nown	-n/n = Value	at POR and BO	R/Value at all of	ther Resets				
'1' = Bit is set		'0' = Bit is cle	ared								
bit 7	ABDOVF: Au	to-Baud Detec	t Overflow bit								
	Asynchronous 1 = Auto-baud 0 = Auto-baud Synchronous Don't care	Asynchronous mode: 1 = Auto-baud timer overflowed 0 = Auto-baud timer did not overflow Synchronous mode: Den/4 core									
bit 6	RCIDL: Rece	ive Idle Flaq bi	t								
	RCIDL: Receive Idle Flag bit Asynchronous mode: 1 = Receiver is Idle 0 = Start bit has been received and the receiver is receiving Synchronous mode: Don't care										
bit 5	Unimplemen	ted: Read as '	0'								
bit 4	SCKP: Synch	ronous Clock I	Polarity Select	t bit							
	Asynchronous	<u>s mode</u> :									
	1 = Transmit i 0 = Transmit i	inverted data to non-inverted da	o the TX/CK p ata to the TX/	in CK pin							
	Synchronous 1 = Data is clo 0 = Data is clo	<u>mode</u> : ocked on rising ocked on falling	edge of the ogen o	clock clock							
bit 3	BRG16: 16-b	it Baud Rate G	enerator bit								
	1 = 16-bit Ba 0 = 8-bit Bau	ud Rate Gener d Rate Genera	ator is used tor is used								
bit 2	Unimplemen	ted: Read as '	0'								
bit 1	WUE: Wake-u	up Enable bit									
	Asynchronous	<u>s mode</u> :									
	 1 = Receiver will autom 0 = Receiver Synchronous Don't care 	is waiting for a atically clear a is operating no <u>mode</u> :	i falling edge. fter RCIF is se rmally	No character et.	will be received,	byte RCIF will	be set. WUE				
bit 0	ABDEN: Auto	-Baud Detect	Enable bit								
	Asynchronous 1 = Auto-Bau 0 = Auto-Bau Synchronous Don't care	<u>s mode</u> : id Detect mode id Detect mode <u>mode</u> :	e is enabled (o e is disabled	clears when au	ito-baud is comp	lete)					

REGISTER 18-3: BAUDCON: BAUD RATE CONTROL REGISTER

18.4 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH:SPBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Example 18-1 provides a sample calculation for determining the desired baud rate, actual baud rate, and baud rate % error.

Typical baud rates and error values for various Asynchronous modes have been computed for your convenience and are shown in Table 18-5. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate. If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

EXAMPLE 18-1: CALCULATING BAUD RATE ERROR



C	onfiguration Bi	ts		Baud Rate Formula		
SYNC	BRG16	BRGH	BRG/EUSART Mode			
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	x	8-bit/Synchronous	Fosc/[4 (n+1)]		
1	1 x 16-bit/Syr		16-bit/Synchronous			

TABLE 18-3: BAUD RATE FORMULAS

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

TABLE 18-4: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page	
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	153	
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	153	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152	
SPBRGL			EUSART	Baud Rate G	enerator, Low	[,] Byte			154*	
SPBRGH		EUSART Baud Rate Generator, High Byte								
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151	

Legend: — = unimplemented, read as '0'. Shaded bits are not used by the BRG.

* Page provides register information.

		SYNC = 0, BRGH = 0, BRG16 = 0											
BAUD	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Foso	c = 16.00	0 MHz	Fosc	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	_		_	_	_	_			_		_		
1200	1221	1.73	255	1200	0.00	239	1202	0.16	207	1200	0.00	143	
2400	2404	0.16	129	2400	0.00	119	2404	0.16	103	2400	0.00	71	
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17	
10417	10417	0.00	29	10286	-1.26	27	10417	0.00	23	10165	-2.42	16	
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8	
57.6k	—	_	_	57.60k	0.00	7	—	_	_	57.60k	0.00	2	
115.2k	—		_	—	—		_		—	_	—		

TABLE 18-5: BAUD RATES FOR ASYNCHRONOUS MODES

					SYNC	C = 0, BRGH	l = 0, BRG	G16 = 0				
BAUD	Fos	c = 8.000	0 MHz	Fos	c = 4.000) MHz	Fosc	: = 3.686	4 MHz	Fos	c = 1.00) 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	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	_	_	_	_	_	—
19.2k	—	_	—	—	_	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	_	—	57.60k	0.00	0	—	—	—
115.2k	—	_	_	—	_	—	_	_	—	_	_	_

					SYNC	C = 0, BRGH	l = 1, BRC	G16 = 0				
BAUD	Foso	: = 20.00	0 MHz	Foso	: = 18.43	2 MHz	Foso	c = 16.00	0 MHz	Fosc	= 11.059	92 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	—	_	—	—	_		-	—	—	_	—	—
1200	—	_	—	—	_	_	—	_	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	_	_
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5

					l = 1, BRC	= 1, BRG16 = 0						
BAUD	Fos	c = 8.00	0 MHz	Fos	c = 4.00	0 MHz	Foso	= 3.686	4 MHz	Fos	c = 1.00	0 MHz
RATE	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 18-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

					SYNC	C = 0, BRGH	l = 0, BRC	616 = 1				
BAUD	Fosc	= 20.00	0 MHz	Fosc	: = 18.43	2 MHz	Fosc	: = 16.00	0 MHz	Fosc	= 11.059 % Error 0.00 0.00 0.00 0.00 0.53 0.00 0.00 0.00	92 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	-0.01	4166	300.0	0.00	3839	300.03	0.01	3332	300.0	0.00	2303
1200	1200	-0.03	1041	1200	0.00	959	1200.5	0.04	832	1200	0.00	575
2400	2399	-0.03	520	2400	0.00	479	2398	-0.08	416	2400	0.00	287
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.818	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.636	-1.36	10	115.2k	0.00	9	111.11k	-3.55	8	115.2k	0.00	5

					SYNC	C = 0, BRGH	l = 0, BRG	616 = 1				
BAUD	Fos	c = 8.00	0 MHz	Fos	c = 4.000) MHz	Fosc	: = 3.686	4 MHz	Fos	c = 1.000) MHz
RATE	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	—	_	_

				SYNC = 0	, BRGH	= 1, BRG16	= 1 or SΥ	′NC = 1,	BRG16 = 1			
BAUD	Foso	: = 20.00	0 MHz	Foso	: = 18.43	2 MHz	Fosc	: = 16.00	0 MHz	Fosc	= 11.059	92 MHz
RATE	Actual Rate	% Error	SPBRG value (decimal)									
300	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	13332	300.0	0.00	9215
1200	1200	-0.01	4166	1200	0.00	3839	1200.1	0.01	3332	1200	0.00	2303
2400	2400	0.02	2082	2400	0.00	1919	2399.5	-0.02	1666	2400	0.00	1151
9600	9597	-0.03	520	9600	0.00	479	9592	-0.08	416	9600	0.00	287
10417	10417	0.00	479	10425	0.08	441	10417	0.00	383	10433	0.16	264
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.23k	0.16	207	19.20k	0.00	143
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.97k	0.64	68	57.60k	0.00	47
115.2k	116.3k	0.94	42	115.2k	0.00	39	114.29k	-0.79	34	115.2k	0.00	23

TABLE 18-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

				SYNC = 0	, BRGH	= 1, BRG16	ট = 1 or S	/NC = 1,	BRG16 = 1					
BAUD	Fos	c = 8.00	0 MHz	Fos	c = 4.000) MHz	Foso	: = 3.686	4 MHz	Fos	c = 1.00) 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	3332	300.0	0.00	3071	300.1	0.04	832		
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207		
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103		
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25		
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23		
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12		
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	_	_		
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	_	_		

18.4.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCON register starts the auto-baud calibration sequence (Figure 18.4.2). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRGL begins counting up using the BRG counter clock as shown in Table 18-6. The fifth rising edge will occur on the RX/ DT pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGH:SPBRGL register pair, the ABDEN bit is automatically cleared, and the RCIF interrupt flag is set. A read operation on the RCREG needs to be performed to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRGL register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 18-6. During ABD, both the SPBRGH and SPBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH and SPBRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

- Note 1: If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte <u>following</u> the Break character (see <u>Section 18.4.3</u> "Auto-Wake-up on Break").
 - 2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.
 - During the auto-baud process, the autobaud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRGL register pair.

TABLE 18-6: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

Note: During the ABD sequence, SPBRGL and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

XXXXh 0000h **BRG** Value 001Ch - Edge #1 - Edge #2 - Edge #3 - Edge #4 Edge #5 RX/DT pin bit 2 bit 3 bit 6 bit 7 Start bit 0 bit 1 bit 4 bit 5 Stop bit Auto Cleared Set by User ABDEN bit RCIDL RCIF bit (Interrupt) Read RCREG XXh SPBRGL 1Ch SPBRGH XXh 00h Note 1: The ABD sequence requires the EUSART module to be configured in Asynchronous mode

FIGURE 18-6: AUTOMATIC BAUD RATE CALIBRATION

18.4.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRGL register pair. After the ABDOVF has been set, the counter continues to count until the fifth rising edge is detected on the RX/DT pin. Upon detecting the fifth RX/DT edge, the hardware will set the RCIF interrupt flag and clear the ABDEN bit of the BAUDCON register. The RCIF flag can be subsequently cleared by reading the RCREG. The ABDOVF flag can be cleared by software directly.

To terminate the auto-baud process before the RCIF flag is set, clear the ABDEN bit then clear the ABDOVF bit. The ABDOVF bit will remain set if the ABDEN bit is not cleared first.

18.4.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 18-7), and asynchronously if the device is in Sleep mode (Figure 18-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

18.4.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be ten or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Startup Time

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

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared by hardware by a rising edge on RX/DT. The interrupt condition is then cleared by software by reading the RCREG register and discarding its contents.

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



FIGURE 18-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



18.4.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 18-9 for the timing of the Break character sequence.

18.4.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

- 1. Configure the EUSART for the desired mode.
- 2. Set the TXEN and SENDB bits to enable the Break sequence.
- 3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
- 4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
- 5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

18.4.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the Received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

A Break character has been received when;

- RCIF bit is set
- · FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 18.4.3** "Auto-Wake-up on **Break**". By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCON register before placing the EUSART in Sleep mode.



FIGURE 18-9: SEND BREAK CHARACTER SEQUENCE

18.5 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

18.5.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for synchronous master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

The TRIS bits corresponding to the RX/DT and TX/CK pins should be set.

18.5.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

18.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock and is sampled on the rising edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock and is sampled on the falling edge of each clock.

18.5.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note:	The TSR register is not mapped in data
	memory, so it is not available to the user.

18.5.1.4 Synchronous Master Transmission Set-up:

- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 18.4 "EUSART Baud Rate Generator (BRG)").
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC. Set the TRIS bits corresponding to the RX/DT and TX/ CK I/O pins.
- 4. Disable Receive mode by clearing bits SREN and CREN.
- 5. Enable Transmit mode by setting the TXEN bit.
- 6. If 9-bit transmission is desired, set the TX9 bit.
- 7. If interrupts are desired, set the TXIE, GIE and PEIE interrupt enable bits.
- 8. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
- 9. Start transmission by loading data to the TXREG register.



FIGURE 18-10: SYNCHRONOUS TRANSMISSION

FIGURE 18-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16		WUE	ABDEN	153
BAUD2CON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE ⁽¹⁾	TXIE ⁽¹⁾	—	—	—	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF ⁽¹⁾	TXIF ⁽¹⁾	_	_	_	TMR1IF	68
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152
SPBRGL			EUSART I	Baud Rate (Generator, Lo	w Byte			154*
SPBRGH			EUSART	Baud Rate (Generator, Hi	gh Byte			154*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	98
TRISG	—	—	_	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	98
TXREG			EU	ISART Tran	smit Registe	ſ			144*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151

TABLE 18-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous master transmission. * Page provides register information.

Note 1: PIC16LF1904/7 only.

18.5.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver must be disabled by setting the corresponding TRIS bits when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTA register) or the Continuous Receive Enable bit (CREN of the RCSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREG. The RCIF bit remains set as long as there are un-read characters in the receive FIFO.

18.5.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/ CK pin output driver must be disabled by setting the associated TRIS bit when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

18.5.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREG is read to access the FIFO. When this happens the OERR bit of the RCSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG.

If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

18.5.1.8 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

18.5.1.9 Synchronous Master Reception Setup:

- 1. Initialize the SPBRGH, SPBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC. Disable RX/DT and TX/CK output drivers by setting the corresponding TRIS bits.
- 4. Ensure bits CREN and SREN are clear.
- 5. If using interrupts, set the GIE and PEIE bits of the INTCON register and set RCIE.
- 6. If 9-bit reception is desired, set bit RX9.
- 7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
- 8. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCIE was set.
- 9. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 10. Read the 8-bit received data by reading the RCREG register.
- 11. 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.

FIGURE 18-12:	SYNCHRONOUS RECEPTION (MASTER MODE, SREN)
RX/DT pin TX/CK pin (SCKP = 0)	
TX/CK pin (SCKP = 1) Write to bit SREN	
SREN bit	'0'
RCIF bit (Interrupt) Read RCREG	
Note: Timing dia	gram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

TABLE 18-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	_	WUE	ABDEN	153
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16		WUE	ABDEN	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE ⁽¹⁾	TXIE ⁽¹⁾	—	_	_	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF ⁽¹⁾	TXIF ⁽¹⁾	—	_	_	TMR1IF	68
RCREG			E	USART Red	ceive Regist	er			147*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152
SPBRGL			EUSAR	F Baud Rate	Generator,	Low Byte			154*
SPBRGH	EUSART Baud Rate Generator, High Byte								154*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous master reception.

* Page provides register information.

Note 1: PIC16LF1904/7 only.

18.5.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for Synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

RX/DT and TX/CK pin output drivers must be disabled by setting the corresponding TRIS bits.

18.5.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see Section 18.5.1.3 "Synchronous Master Transmission"), except in the case of the Sleep mode. If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- 1. The first character will immediately transfer to the TSR register and transmit.
- 2. The second word will remain in TXREG register.
- 3. The TXIF bit will not be set.
- After the first character has been shifted out of TSR, the TXREG register will transfer the second character to the TSR and the TXIF bit will now be set.
- If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.
- 18.5.2.2 Synchronous Slave Transmission Set-up:
- 1. Set the SYNC and SPEN bits and clear the CSRC bit.
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- 3. Clear the CREN and SREN bits.
- If using interrupts, ensure that the GIE and PEIE bits of the INTCON register are set and set the TXIE bit.
- 5. If 9-bit transmission is desired, set the TX9 bit.
- 6. Enable transmission by setting the TXEN bit.
- 7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
- 8. Start transmission by writing the Least Significant eight bits to the TXREG register.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	153
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE ⁽¹⁾	TXIE ⁽¹⁾	—	—	_	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF ⁽¹⁾	TXIF ⁽¹⁾	_	_		TMR1IF	68
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152
SPBRGL			EUSART	Baud Rate	Generator,	Low Byte			154*
SPBRGH			EUSART	Baud Rate	Generator,	High Byte			154*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	98
TXREG			EU	JSART Trar	nsmit Regis	ter			144*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151

TABLE 18-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous slave transmission.

* Page provides register information.

Note 1: PIC16LF1904/7 only.

18.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (Section 18.5.1.5 "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.

- 18.5.2.4 Synchronous Slave Reception Setup:
- 1. Set the SYNC and SPEN bits and clear the CSRC bit.
- 2. Set the RX/DT and TX/CK TRIS controls to '1'.
- If using interrupts, ensure that the GIE and PEIE bits of the INTCON register are set and set the RCIE bit.
- 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
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	153
BAUD2CON	ABDOVF	RCIDL	—	SCKP	BRG16	_	WUE	ABDEN	153
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	65
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	_	_	TMR1IE	66
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	_	—	TMR1IF	68
RCREG			E	USART Rec	ceive Registe	er			147*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	152
SPBRGL			EUSART	Baud Rate	Generator, I	_ow Byte			154*
SPBRGH		EUSART Baud Rate Generator, High Byte							
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	151

TABLE 18-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for synchronous slave reception.

* Page provides register information.

19.0 LIQUID CRYSTAL DISPLAY (LCD) DRIVER MODULE

The Liquid Crystal Display (LCD) driver module generates the timing control to drive a static or multiplexed LCD panel. In the PIC16LF1904/6/7 device, the module drives the panels of up to four commons and up to 116 total segments. The LCD module also provides control of the LCD pixel data.

The LCD driver module supports:

- Direct driving of LCD panel
- · Three LCD clock sources with selectable prescaler
- Up to four common pins:
 - Static (1 common)
 - 1/2 multiplex (2 commons)
 - 1/3 multiplex (3 commons)
 - 1/4 multiplex (4 commons)
- 19 Segment pins (PIC16LF1906 only)
- 29 Segment pins (PIC16LF1904/7 only)

• Static, 1/2 or 1/3 LCD Bias

Note: COM3 and SEG15 share the same physical pin on the PIC16LF1906, therefore SEG15 is not available when using 1/4 multiplex displays.

19.1 LCD Registers

The module contains the following registers:

- LCD Control register (LCDCON)
- LCD Phase register (LCDPS)
- LCD Reference Ladder register (LCDRL)
- LCD Contrast Control register (LCDCST)
- LCD Reference Voltage Control register (LCDREF)
- Up to 4 LCD Segment Enable registers (LCDSEn)
- Up to 16 LCD data registers (LCDDATAn)

FIGURE 19-1: LCD DRIVER MODULE BLOCK DIAGRAM



TABLE 19-1: LCD SEGMENT AND DATA REGISTERS

	# of LCD Registers				
Device	Segment Enable	Data			
PIC16LF1904/7	3	16			
PIC16LF1906	4	12			

The LCDCON register (Register 19-1) controls the operation of the LCD driver module. The LCDPS register (Register 19-2) configures the LCD clock source prescaler and the type of waveform; Type-A or Type-B. The LCDSEn registers (Register 19-5) configure the functions of the port pins.

The following LCDSEn registers are available:

- LCDSE0 SE<7:0>
- LCDSE1 SE<15:8>
- LCDSE2 SE<23:16>(1)
- LCDSE3 SE<28:24>⁽¹⁾ (SE<26:24>⁽²⁾)

Once the module is initialized for the LCD panel, the individual bits of the LCDDATAn registers are cleared/set to represent a clear/dark pixel, respectively:

- LCDDATA0 SEG<7:0>COM0
- LCDDATA1 SEG<15:8>COM0
- LCDDATA2 SEG<23:16>COM0⁽¹⁾
- LCDDATA3 SEG<7:0>COM1
- LCDDATA4 SEG<15:8>COM1
- LCDDATA5 SEG<23:16>COM1⁽¹⁾
- LCDDATA6 SEG<7:0>COM2
- LCDDATA7 SEG<15:8>COM2
- LCDDATA8 SEG<23:16>COM2⁽¹⁾
- LCDDATA9 SEG<7:0>COM3
- LCDDATA10 SEG<15:8>COM3
- LCDDATA11 SEG<23:16>COM3⁽¹⁾
- LCDDATA12 SEG<28:24>COM0⁽¹⁾ (SEG<26:24>)⁽²⁾
- LCDDATA15 SEG<28:24>COM1⁽¹⁾ (SEG<26:24>)⁽²⁾
- LCDDATA18 SEG<28:24>COM2⁽¹⁾ (SEG<26:24>)⁽²⁾
- LCDDATA21 SEG<28:24>COM3⁽¹⁾ (SEG<26:24>)⁽²⁾

Note 1: PIC16LF1904/7 only.

2: PIC16LF1906 only.

As an example, LCDDATAn is detailed in Register 19-6.

Once the module is configured, the LCDEN bit of the LCDCON register is used to enable or disable the LCD module. The LCD panel can also operate during Sleep by clearing the SLPEN bit of the LCDCON register.

R/W-0/0	R/W-0/0	R/C-0/0	U-0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1			
LCDEN	SLPEN	WERR	_	(CS<1:0>	LMUX	<1:0>			
bit 7							bit 0			
Legend:										
R = Readable	e bit	W = Writable b	Vritable bit U = Unimplemented bit, read as '0'							
u = Bit is uncl	hanged	x = Bit is unkn	own	-n/n = Valu	e at POR and BO	R/Value at all ot	her Resets			
'1' = Bit is set		'0' = Bit is clea	red	C = Only c	learable bit					
bit 7	LCDEN: LCD	Driver Enable	bit							
	1 = LCD drive	er module is ena	abled							
	0 = LCD drive	er module is dis	abled							
bit 6	SLPEN: LCD	Driver Enable i	n Sleep Mo	de bit						
	 1 = LCD driver module is disabled in Sleep mode 0 = LCD driver module is enabled in Sleep mode 									
bit 5	WERR: LCD	Write Failed Eri	or bit							
	1 = LCDDAT	An register wri	tten while t	he WA bit of	the LCDPS regist	ter = 0 (must l	be cleared in			
	software))								
h :+ 4		vrite error	.,							
		ted: Read as ()							
DIT 3-2		CK Source Sele	Ct dits							
	00 = F0SC/25 01 = T1OSC	(Timer1)								
	1x = LFINTO	SC (31 kHz)								
bit 1-0	LMUX<1:0>:	Commons Sele	ect bits							
	Maximum Number of Pixels									
		wuitiple	PI	C16LF1906	PIC16LF1904/7	DIdS				
	00	Static (CON	/0)	19	29	Static				
	01	1/2 (COM<1	:0>)	38	58	1/2 or 1/3				
	10	1/3 (COM<2	:0>)	57	87	1/2 or 1/3				
	11	1/4 (COM<3	:0>)	72 ⁽¹⁾	116	1/3				

REGISTER 19-1: LCDCON: LIQUID CRYSTAL DISPLAY (LCD) CONTROL REGISTER

Note 1: On these devices, COM3 and SEG15 are shared on one pin, limiting the device from driving 72 segments.

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1
WFT	BIASMD	LCDA	WA		LP<	3:0>	
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplen	nented bit, reac	l as '0'	
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all c	other Resets
'1' = Bit is set		'0' = Bit is clea	ared	C = Only clea	rable bit		
bit 7	WFT: Wavefo	rm Type bit					
	1 = Type-B p	hase changes	on each fram	e boundary			
h:+ 0	0 = Type-A p	nase changes		common type			
DILO	When I MUX		DI				
	0 = Static Bia	<u>< 1.02 = 00.</u> s mode (do not	eat this hit to	· (1 ')			
	When LMUX	<1:0> = <u>01</u> :) I)			
	1 = 1/2 Bias r	node					
	0 = 1/3 Bias r	node					
	When LMUX<	< <u>1:0> = 10:</u>					
	1 = 1/2 Bias r	node					
	When I MUX	<1:0> = 11:					
	0 = 1/3 Bias r	node (do not se	et this bit to '1	L')			
bit 5	LCDA: LCD A	Active Status bi	t	,			
	1 = LCD drive	er module is act	tive				
	0 = LCD drive	er module is ina	active				
bit 4	WA: LCD Wri	te Allow Status	bit				
	1 = Writing to	the LCDDATA	n registers is	allowed			
			n registers is	not allowed			
bit 3-0	LP<3:0>: LCI	D Prescaler Se	lection bits				
	1111 = 1.10 1110 = 1.15						
	1101 = 1:14						
	1100 = 1:13						
	1011 = 1:12						
	1010 = 1.11 1001 = 1.10						
	1000 = 1:9						
	0111 = 1 :8						
	0110 = 1.7						
	0101 = 1.6 0100 = 1.5						
	0011 = 1 :4						
	0010 = 1 :3						
	0001 = 1:2						
	0000 = 1:1						

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
LCDIRE		LCDIRI		VLCD3PE	VLCD2PE	VLCD1PE	_
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimpler	nented bit, read	as '0'	
u = Bit is unch	nanged	x = Bit is unkn	own	-n/n = Value a	at POR and BOI	R/Value at all of	ther Resets
'1' = Bit is set		'0' = Bit is clea	ared	C = Only clea	rable bit		
bit 7	LCDIRE: LCD	D Internal Refer	ence Enable	bit			
	1 = Internal L 0 = Internal L	CD Reference	is enabled an is disabled	nd connected to	o the Internal Co	ontrast Control	circuit
bit 6	Unimplemen	ted: Read as 'd)'				
bit 5	LCDIRI: LCD	Internal Refere	ence Ladder I	dle Enable bit			
	Allows the Int 1 = When th 0 = The LCE	ernal FVR buffe ne LCD Referer D Internal FVR	er to shut dov ice Ladder is Buffer ignores	vn when the LC in power mode s the LCD Refe	D Reference La 'B', the LCD Int rence Ladder P	adder is in powe ernal FVR buffe ower mode.	er mode 'B' er is disabled.
bit 4	Unimplemen	ted: Read as ')'				
bit 3	VLCD3PE: V	LCD3 Pin Enat	le bit				
	1 = The VLC 0 = The VLC	D3 pin is conne D3 pin is not co	ected to the ir	nternal bias volt	age LCDBIAS3	(1)	
bit 2	VLCD2PE: V	LCD2 Pin Enat	le bit				
	1 = The VLC 0 = The VLC	D2 pin is conne D2 pin is not co	ected to the ir	nternal bias volt	age LCDBIAS2	(1)	
bit 1	VLCD1PE: V	LCD1 Pin Enat	le bit				
 1 = The VLCD1 pin is connected to the internal bias voltage LCDBIAS1⁽¹⁾ 0 = The VLCD1 pin is not connected 							
bit 0	Unimplemen	ted: Read as 'o)'				
Note 1: No	ormal pin control	s of TRISx and	ANSELx are	unaffected.			

REGISTER 19-3: LCDREF: LCD REFERENCE VOLTAGE CONTROL REGISTER

REGISTER 19-4: LCDCST: LCD CONTRAST CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—		LCDCST<2:0>	
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'							
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value at POR and BOR/Value at all other Resets			

C = Only clearable bit

bit 7-3	Unimplemented:	Read	as '	0'

1' = Bit is set

bit 2-0 LCDCST<2:0>: LCD Contrast Control bits

'0' = Bit is cleared

Selects the resistance of the LCD contrast control resistor ladder

Bit Value = Resistor ladder

000 = Minimum Resistance (Maximum contrast). Resistor ladder is shorted.

001 = Resistor ladder is at 1/7th of maximum resistance

010 = Resistor ladder is at 2/7th of maximum resistance

011 = Resistor ladder is at 3/7th of maximum resistance

100 = Resistor ladder is at 4/7th of maximum resistance

101 = Resistor ladder is at 5/7th of maximum resistance

110 = Resistor ladder is at 6/7th of maximum resistance

111 = Resistor ladder is at maximum resistance (Minimum contrast).

REGISTER 19-5: LCDSEn: LCD SEGMENT ENABLE 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
SEn	SEn	SEn	SEn	SEn	SEn	SEn	SEn
bit 7							bit 0
Legend:							
R = Readable b	bit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'	
u = Bit is uncha	anged	x = Bit is unkn	iown	-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 7-0 SEn: Segment Enable bits 1 = Segment function of the pin is enabled 0 = I/O function of the pin is enabled

REGISTER 19-6: LCDDATAn: LCD DATA REGISTERS

| R/W-x/u |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| SEGx-COMy |
| 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 SEGx-COMy: Pixel On bits

1 = Pixel on (dark) 0 = Pixel off (clear)

19.2 LCD Clock Source Selection

The LCD module has three possible clock sources:

- Fosc/256
- T10SC
- LFINTOSC

The first clock source is the system clock divided by 256 (Fosc/256). This divider ratio is chosen to provide about 1 kHz output when the system clock is 8 MHz. The divider is not programmable. Instead, the LCD prescaler bits LP<3:0> of the LCDPS register are used to set the LCD frame clock rate.

The second clock source is the T1OSC. This also gives about 1 kHz when a 32.768 kHz crystal is used with the Timer1 oscillator. To use the Timer1 oscillator as a clock source, the T1OSCEN bit of the T1CON register should be set.

The third clock source is the 31 kHz LFINTOSC, which provides approximately 1 kHz output.

The second and third clock sources may be used to continue running the LCD while the processor is in Sleep.

Using bits CS<1:0> of the LCDCON register can select any of these clock sources.

19.2.1 LCD PRESCALER

A 4-bit counter is available as a prescaler for the LCD clock. The prescaler is not directly readable or writable; its value is set by the LP<3:0> bits of the LCDPS register, which determine the prescaler assignment and prescale ratio.

The prescale values are selectable from 1:1 through 1:16.



FIGURE 19-2: LCD CLOCK GENERATION

19.3 LCD Bias Voltage Generation

The LCD module can be configured for one of three bias types:

- Static Bias (2 voltage levels: Vss and VLCD)
- 1/2 Bias (3 voltage levels: Vss, 1/2 VLcD and VLcD)
- 1/3 Bias (4 voltage levels: Vss, 1/3 VLCD, 2/3 VLCD and VLCD)

TABLE 19-2: LCD BIAS VOLTAGES

	Static Bias	1/2 Bias	1/3 Bias
LCD Bias 0	Vss	Vss	Vss
LCD Bias 1	_	1/2 Vdd	1/3 Vdd
LCD Bias 2	_	1/2 Vdd	2/3 Vdd
LCD Bias 3	VLCD3	VLCD3	VLCD3

So that the user is not forced to place external components and use up to three pins for bias voltage generation, internal contrast control and an internal reference ladder are provided internally to the PIC16LF1904/6/7. Both of these features may be used in conjunction with the external VLCD<3:1> pins, to provide maximum flexibility. Refer to Figure 19-3.

FIGURE 19-3: LCD BIAS VOLTAGE GENERATION BLOCK DIAGRAM



19.4 LCD Bias Internal Reference Ladder

The internal reference ladder can be used to divide the LCD bias voltage two or three equally spaced voltages that will be supplied to the LCD segment pins. To create this, the reference ladder consists of three matched resistors. Refer to Figure 19-3.

19.4.1 BIAS MODE INTERACTION

When in 1/2 Bias mode (BIASMD = 1), then the middle resistor of the ladder is shorted out so that only two voltages are generated. The current consumption of the ladder is higher in this mode, with the one resistor removed.

TABLE 19-3:LCD INTERNAL LADDERPOWER MODES (1/3 BIAS)

Power Mode	Nominal Resistance of Entire Ladder	Nominal IDD
Low	3 Mohm	1 µA
Medium	300 kohm	10 µA
High	30 kohm	100 µA

19.4.2 POWER MODES

The internal reference ladder may be operated in one of three power modes. This allows the user to trade off LCD contrast for power in the specific application. The larger the LCD glass, the more capacitance is present on a physical LCD segment, requiring more current to maintain the same contrast level.

Three different power modes are available, LP, MP and HP. The internal reference ladder can also be turned off for applications that wish to provide an external ladder or to minimize power consumption. Disabling the internal reference ladder results in all of the ladders being disconnected, allowing external voltages to be supplied.

Whenever the LCD module is inactive (LCDA = 0), the internal reference ladder will be turned off.
19.4.3 AUTOMATIC POWER MODE SWITCHING

As an LCD segment is electrically only a capacitor, current is drawn only during the interval where the voltage is switching. To minimize total device current, the LCD internal reference ladder can be operated in a different power mode for the transition portion of the duration. This is controlled by the LCDRL Register (Register 19-7). The LCDRL register allows switching between two power modes, designated 'A' and 'B'. 'A' Power mode is active for a programmable time, beginning at the time when the LCD segments transition. 'B' Power mode is the remaining time before the segments or commons change again. The LRLAT<2:0> bits select how long, if any, that the 'A' Power mode is active. Refer to Figure 19-4.

To implement this, the 5-bit prescaler used to divide the 32 kHz clock down to the LCD controller's 1 kHz base rate is used to select the power mode.

FIGURE 19-4: LCD INTERNAL REFERENCE LADDER POWER MODE SWITCHING DIAGRAM – TYPE A







R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
LRLA	LRLAP<1:0>		P<1:0>	—		LRLAT<2:0>	
bit 7							bit 0
r							
Legend:							
R = Readable	bit	W = Writable bit		U = Unimplemented bit, read as '0'			
u = Bit is unch	anged	x = Bit is unkno	own	-n/n = Value a	t POR and BOI	R/Value at all oth	er Resets
'1' = Bit is set		'0' = Bit is clea	= Bit is cleared				
bit 7-6	LRLAP<1:0>: LCD Reference Ladder A Time Power Control bits During Time interval A (Refer to Figure 19-4): 00 = Internal LCD Reference Ladder is powered down and unconnected 01 = Internal LCD Reference Ladder is powered in low-power mode 10 = Internal LCD Reference Ladder is powered in medium-power mode						
bit 5-4	LRLBP<1:0>: LCD Reference Ladder is powered in high-power mode LRLBP<1:0>: LCD Reference Ladder B Time Power Control bits During Time interval B (Refer to Figure 19-4): 00 = Internal LCD Reference Ladder is powered down and unconnected 01 = Internal LCD Reference Ladder is powered in low-power mode 10 = Internal LCD Reference Ladder is powered in medium-power mode 11 = Internal LCD Reference Ladder is powered in bith-power mode						
bit 3	Unimplement	ed: Read as '0'	,				
bit 2-0	LRLAT<2:0>:	LCD Reference	e Ladder A Tim	ne Interval Contr	rol bits		
	Sets the number of 32 kHz clocks that the A Time interval power mode is active						
	For type A waveforms (WFT = 0):						
	000 = Internal 001 = Internal 010 = Internal 011 = Internal 100 = Internal 110 = Internal 111 = Internal	LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference	e Ladder is alv e Ladder is in ' e Ladder is in '	vays in 'B' Powe A' Power mode A' Power mode A' Power mode A' Power mode A' Power mode A' Power mode A' Power mode	er mode for 1 clock and for 2 clocks and for 3 clocks and for 4 clocks and for 5 clocks and for 6 clocks and for 7 clocks and	'B' Power mode I 'B' Power mode	ofor 15 clocks e for 14 clocks e for 13 clocks e for 12 clocks e for 11 clocks e for 10 clocks de for 9 clocks
	For type B way	veforms (WFT =	1):				
	000 = Internal 001 = Internal 010 = Internal 011 = Internal 100 = Internal 101 = Internal 110 = Internal	LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference LCD Reference	e Ladder is alw e Ladder is in ' e Ladder is in '	vays in 'B' Powe A' Power mode A' Power mode A' Power mode A' Power mode A' Power mode A' Power mode	er mode. for 1 clock and for 2 clocks and for 3 clocks and for 4 clocks and for 5 clocks and for 6 clocks and for 7 clocks and	'B' Power mode I 'B' Power mode	a for 31 clocks e for 30 clocks e for 29 clocks e for 28 clocks e for 27 clocks e for 26 clocks e for 26 clocks e for 25 clocks

REGISTER 19-7: LCDRL: LCD REFERENCE LADDER CONTROL REGISTERS

The contrast control circuit is used to decrease the output voltage of the signal source by a total of

Whenever the LCD module is inactive (LCDA = 0), the

approximately 10%, when LCDCST = 111.

19.4.4 CONTRAST CONTROL

The LCD contrast control circuit consists of a seven-tap resistor ladder, controlled by the LCDCST bits. Refer to Figure 19-7.

contrast control ladder will be turned off (open).



19.4.5 INTERNAL REFERENCE

Under firmware control, an internal reference for the LCD bias voltages can be enabled. When enabled, the source of this voltage can be VDD. When no internal reference is selected, the LCD contrast control circuit is disabled and LCD bias must be provided externally.

Whenever the LCD module is inactive (LCDA = 0), the internal reference will be turned off.

When the internal reference is enabled and the Fixed Voltage Reference is selected, the LCDIRI bit can be used to minimize power consumption by tying into the LCD reference ladder automatic power mode switching. When LCDIRI = 1 and the LCD reference ladder is in Power mode 'B', the LCD internal FVR buffer is disabled.

Note: The LCD module automatically turns on the Fixed Voltage Reference when needed.

19.4.6 VLCD<3:1> PINS

The VLCD<3:1> pins provide the ability for an external LCD bias network to be used instead of the internal ladder. Use of the VLCD<3:1> pins does not prevent use of the internal ladder. Each VLCD pin has an independent control in the LCDREF register (Register 19-3), allowing access to any or all of the LCD Bias signals. This architecture allows for maximum flexibility in different applications

For example, the VLCD<3:1> pins may be used to add capacitors to the internal reference ladder, increasing the drive capacity.

For applications where the internal contrast control is insufficient, the firmware can choose to only enable the VLCD3 pin, allowing an external contrast control circuit to use the internal reference divider.

19.5 LCD Multiplex Types

The LCD driver module can be configured into one of four multiplex types:

- Static (only COM0 is used)
- 1/2 multiplex (COM<1:0> are used)
- 1/3 multiplex (COM<2:0> are used)
- 1/4 multiplex (COM<3:0> are used)

The LMUX<1:0> bit setting of the LCDCON register decides which of the LCD common pins are used (see Table 19-4 for details).

If the pin is a digital I/O, the corresponding TRIS bit controls the data direction. If the pin is a COM drive, then the TRIS setting of that pin is overridden.

Multiplex	LMUX <1:0>	СОМЗ	COM2	COM1	COM1
Static	00	Unused	Unused	Unused	Active
1/2	01	Unused	Unused	Active	Active
1/3	10	Unused	Active	Active	Active
1/4	11	Active	Active	Active	Active

TABLE 19-4: COMMON PIN USAGE

19.6 Segment Enables

The LCDSEn registers are used to select the pin function for each segment pin. The selection allows each pin to operate as either an LCD segment driver or as one of the pin's alternate functions. To configure the pin as a segment pin, the corresponding bits in the LCDSEn registers must be set to '1'.

If the pin is a digital I/O, the corresponding TRIS bit controls the data direction. Any bit set in the LCDSEn registers overrides any bit settings in the corresponding TRIS register.

Note: On a Power-on Reset, these pins are configured as normal I/O, not LCD pins.

19.7 Pixel Control

The LCDDATAx registers contain bits which define the state of each pixel. Each bit defines one unique pixel.

Register 19-6 shows the correlation of each bit in the LCDDATAx registers to the respective common and segment signals.

Any LCD pixel location not being used for display can be used as general purpose RAM.

19.8 LCD Frame Frequency

The rate at which the COM and SEG outputs change is called the LCD frame frequency.

TABLE 19-5: FRAME FREQUENCY FORMULAS

Multiplex	Frame Frequency ⁽²⁾ =
Static	Clock source ⁽¹⁾ /(4 x (LCD Prescaler) x 32 x 1))
1/2	Clock source ⁽¹⁾ /(2 x (LCD Prescaler) x 32 x 2))
1/3	Clock source ⁽¹⁾ /(1 x (LCD Prescaler) x 32 x 3))
1/4	Clock source ⁽¹⁾ /(1 x (LCD Prescaler) x 32 x 4))
Note 1:	Clock source is Fosc/256. T1OSC or LFIN-

Note 1: Clock source is FOSC/256, TIOSC or LFIN-TOSC.

2: See Figure 19-2.

TABLE 19-6:APPROXIMATE FRAME
FREQUENCY (IN Hz) USING
Fosc @ 8 MHz, TIMER1 @
32.768 kHz OR LFINTOSC

LP<3:0>	Static	1/2	1/3	1/4
2	122	122	162	122
3	81	81	108	81
4	61	61	81	61
5	49	49	65	49
6	41	41	54	41
7	35	35	47	35

19.9 LCD Waveform Generation

LCD waveforms are generated so that the net AC voltage across the dark pixel should be maximized and the net AC voltage across the clear pixel should be minimized. The net DC voltage across any pixel should be zero.

The COM signal represents the time slice for each common, while the SEG contains the pixel data.

The pixel signal (COM-SEG) will have no DC component and it can take only one of the two RMS values. The higher RMS value will create a dark pixel and a lower RMS value will create a clear pixel.

As the number of commons increases, the delta between the two RMS values decreases. The delta represents the maximum contrast that the display can have.

The LCDs can be driven by two types of waveform: Type-A and Type-B. In Type-A waveform, the phase changes within each common type, whereas in Type-B waveform, the phase changes on each frame boundary. Thus, Type-A waveform maintains 0 VDc over a single frame, whereas Type-B waveform takes two frames.

- Note 1: If Sleep has to be executed with LCD Sleep disabled (LCDCON<SLPEN> is '1'), then care must be taken to execute Sleep only when VDC on all the pixels is '0'.
 - 2: When the LCD clock source is Fosc/256, if Sleep is executed, irrespective of the LCDCON<SLPEN> setting, the LCD immediately goes into Sleep. Thus, take care to see that VDC on all pixels is '0' when Sleep is executed.

Figure 19-8 through Figure 19-18 provide waveforms for static, half-multiplex, 1/3-multiplex and 1/4-multiplex drives for Type-A and Type-B waveforms.



FIGURE 19-8: TYPE-A/TYPE-B WAVEFORMS IN STATIC DRIVE



FIGURE 19-9: TYPE-A WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE











FIGURE 19-13: TYPE-A WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE











FIGURE 19-17: TYPE-A WAVEFORMS IN 1/4 MUX, 1/3 BIAS DRIVE



19.10 LCD Interrupts

The LCD module provides an interrupt in two cases. An interrupt when the LCD controller goes from active to inactive controller. An interrupt also provides unframe boundaries for Type B waveform. The LCD timing generation provides an interrupt that defines the LCD frame timing.

19.10.1 LCD INTERRUPT ON MODULE SHUTDOWN

An LCD interrupt is generated when the module completes shutting down (LCDA goes from '1' to '0').

19.10.2 LCD FRAME INTERRUPTS

A new frame is defined to begin at the leading edge of the COM0 common signal. The interrupt will be set immediately after the LCD controller completes accessing all pixel data required for a frame. This will occur at a fixed interval before the frame boundary (TFINT), as shown in Figure 19-19. The LCD controller will begin to access data for the next frame within the interval from the interrupt to when the controller begins to access data after the interrupt (TFWR). New data must be written within TFWR, as this is when the LCD controller will begin to access the data for the next frame.

When the LCD driver is running with Type-B waveforms and the LMUX<1:0> bits are not equal to '00' (static drive), there are some additional issues that must be addressed. Since the DC voltage on the pixel takes two frames to maintain zero volts, the pixel data must not change between subsequent frames. If the pixel data were allowed to change, the waveform for the odd frames would not necessarily be the complement of the waveform generated in the even frames and a DC component would be introduced into the panel. Therefore, when using Type-B waveforms, the user must synchronize the LCD pixel updates to occur within a subframe after the frame interrupt.

To correctly sequence writing while in Type-B, the interrupt will only occur on complete phase intervals. If the user attempts to write when the write is disabled, the WERR bit of the LCDCON register is set and the write does not occur.

Note:	The LCD frame interrupt is not generated			
	when the Type-A waveform is selected			
	and when the Type-B with no multiplex			
	(STATIC) IS SEIECTED.			





19.11 Operation During Sleep

The LCD module can operate during Sleep. The selection is controlled by bit SLPEN of the LCDCON register. Setting the SLPEN bit allows the LCD module to go to Sleep. Clearing the SLPEN bit allows the module to continue to operate during Sleep.

If a SLEEP instruction is executed and SLPEN = 1, the LCD module will cease all functions and go into a very low-current Consumption mode. The module will stop operation immediately and drive the minimum LCD voltage on both segment and common lines. Figure 19-20 shows this operation.

The LCD module can be configured to operate during Sleep. The selection is controlled by bit SLPEN of the LCDCON register. Clearing SLPEN and correctly configuring the LCD module clock will allow the LCD module to operate during Sleep. Setting SLPEN and correctly executing the LCD module shutdown will disable the LCD module during Sleep and save power.

If a SLEEP instruction is executed and SLPEN = 1, the LCD module will immediately cease all functions, drive the outputs to Vss and go into a very low-current mode. The SLEEP instruction should only be executed after the LCD module has been disabled and the current cycle completed, thus ensuring that there are no DC voltages on the glass. To disable the LCD module, clear the LCDEN bit. The LCD module will complete the disabling process after the current frame, clear the LCDA bit and optionally cause an interrupt.

The steps required to properly enter Sleep with the LCD disabled are:

- Clear LCDEN
- Wait for LCDA = 0 either by polling or by interrupt
- Execute SLEEP

If SLPEN = 0 and SLEEP is executed while the LCD module clock source is FOSC/4, then the LCD module will halt with the pin driving the last LCD voltage pattern. Prolonged exposure to a fixed LCD voltage pattern will cause damage to the LCD glass. To prevent LCD glass damage, either perform the proper LCD module shutdown prior to Sleep, or change the LCD module clock to allow the LCD module to continue operation during Sleep.

If a SLEEP instruction is executed and SLPEN = 0 and the LCD module clock is either T1OSC or LFINTOSC, the module will continue to display the current contents of the LCDDATA registers. While in Sleep, the LCD data cannot be changed. If the LCDIE bit is set, the device will wake from Sleep on the next LCD frame boundary. The LCD module current consumption will not decrease in this mode; however, the overall device power consumption will be lower due to the shutdown of the CPU and other peripherals. Table 19-7 shows the status of the LCD module during a Sleep while using each of the three available clock sources.

Note:	When the LCDEN bit is cleared, the LCD
	module will be disabled at the completion
	of frame. At this time, the port pins will
	revert to digital functionality. To minimize
	power consumption due to floating digital
	inputs, the LCD pins should be driven low
	using the PORT and TRIS registers.

If a SLEEP instruction is executed and SLPEN = 0, the module will continue to display the current contents of the LCDDATA registers. To allow the module to continue operation while in Sleep, the clock source must be either the LFINTOSC or T1OSC external oscillator. While in Sleep, the LCD data cannot be changed. The LCD module current consumption will not decrease in this mode; however, the overall consumption of the device will be lower due to shut down of the core and other peripheral functions.

Table 19-7 shows the status of the LCD module during Sleep while using each of the three available clock sources:

TABLE 19-7:	LCD MODULE STATUS		
	DURING SLEEP		

Clock Source	SLPEN	Operational During Sleep
T1080	0	Yes
11030	1	No
	0	Yes
LFINTOSC	1	No
Eccc/4	0	No
FUSC/4	1	No

Note: The LFINTOSC or external T1OSC oscillator must be used to operate the LCD module during Sleep.

If LCD interrupts are being generated (Type-B waveform with a Multiplex mode not static) and LCDIE = 1, the device will awaken from Sleep on the next frame boundary.



19.12 Configuring the LCD Module

The following is the sequence of steps to configure the LCD module.

- 1. Select the frame clock prescale using bits LP<3:0> of the LCDPS register.
- 2. Configure the appropriate pins to function as segment drivers using the LCDSEn registers.
- 3. Configure the LCD module for the following using the LCDCON register:
 - Multiplex and Bias mode, bits LMUX<1:0>
 - Timing source, bits CS<1:0>
 - Sleep mode, bit SLPEN
- 4. Write initial values to pixel data registers, LCD-DATA0 through LCDDATA21.
- 5. Clear LCD Interrupt Flag, LCDIF bit of the PIR2 register and if desired, enable the interrupt by setting bit LCDIE of the PIE2 register.
- Configure bias voltages by setting the LCDRL, LCDREF and the associated ANSELx registers as needed.
- 7. Enable the LCD module by setting bit LCDEN of the LCDCON register.

19.13 Disabling the LCD Module

To disable the LCD module, write all '0's to the LCDCON register.

19.14 LCD Current Consumption

When using the LCD module the current consumption consists of the following three factors:

- Oscillator Selection
- · LCD Bias Source
- Capacitance of the LCD segments

The current consumption of just the LCD module can be considered negligible compared to these other factors.

19.14.1 OSCILLATOR SELECTION

The current consumed by the clock source selected must be considered when using the LCD module. See **Section 22.0 "Electrical Specifications"** for oscillator current consumption information.

19.14.2 LCD BIAS SOURCE

The LCD bias source, internal or external, can contribute significantly to the current consumption. Use the highest possible resistor values while maintaining contrast to minimize current.

19.14.3 CAPACITANCE OF THE LCD SEGMENTS

The LCD segments which can be modeled as capacitors which must be both charged and discharged every frame. The size of the LCD segment and its technology determines the segment's capacitance.

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	65
LCDCON	LCDEN	SLPEN	WERR	_	CS1	CS0	LMUX	<1:0>	173
LCDCST	—	—	_	_	_	I	_CDCST<2:0>	>	176
LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	177
LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	177
LCDDATA2 ⁽¹⁾	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	177
LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	177
LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	177
LCDDATA5 ⁽¹⁾	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	177
LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	177
LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	177
LCDDATA8 ⁽¹⁾	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	177
LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	177
LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	177
LCDDATA11 ⁽¹⁾	SEG23 COM3	SEG22 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	SEG15 COM3	177
LCDDATA12	—	—		SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0	177
LCDDATA15	_	—		SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1	177
LCDDATA18	—	—		SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2	177
LCDDATA21	_	_		SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3	177
LCDPS	WFT	BIASMD	LCDA	WA		LP<	<3:0>		174
LCDREF	LCDIRE	—	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	175
LCDRL	LRLA	P<1:0>	LRLBF	P<1:0>	—		LRLAT<2:0>		184
LCDSE0				SE	<7:0>				177
LCDSE1				SE	<15:8>				177
LCDSE2				SE<	23:16>				177
LCDSE3	_	—	_			SE<28:24>			177
PIE2	—	—	_	—	—	LCDIE	_	_	67
PIR2	—	—	_	—	_	LCDIF	—	_	69
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	_	TMR10N	139

 TABLE 19-8:
 SUMMARY OF REGISTERS ASSOCIATED WITH LCD OPERATION

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

Note 1: PIC16LF1904/7 only.

20.0 IN-CIRCUIT SERIAL PROGRAMMING[™] (ICSP[™])

ICSP[™] programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process, allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP[™] programming:

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- Vss

In Program/Verify mode the program memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the "PIC16F193X/LF193X/PIC16F194X/LF194X/PIC16LF 190X Memory Programming Specification" (DS41397).

20.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on MCLR/VPP to VIHH.

20.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC[®] Flash MCUs to be programmed using VDD only, without high voltage. When the LVP bit of Configuration Words is set to '1', the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to '0'.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

- 1. MCLR is brought to VIL.
- 2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{\text{MCLR}}$ must be held at VIL for as long as Program/Verify mode is to be maintained.

If low-voltage programming is enabled (LVP = 1), the $\overline{\text{MCLR}}$ Reset function is automatically enabled and cannot be disabled. See **Section 5.4 "MCLR"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

20.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP[™] header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6 connector) configuration. See Figure 20-1.





Another connector often found in use with the PICkit[™] programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 20-2.

For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes, or even jumpers. See Figure 20-3 for more information.









21.0 INSTRUCTION SET SUMMARY

Each PIC16 instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- · Byte Oriented
- · Bit Oriented
- · Literal and Control

The literal and control category contains the most varied instruction word format.

Table 21-3 lists the instructions recognized by the MPASMTM assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
х	Don't care location (= 0 or 1). The assembler will generate code with x = 0 . It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 21-2: ABBREVIATION DESCRIPTIONS

Field	Description		
PC	Program Counter		
TO	Time-out bit		
С	Carry bit		
DC	Digit carry bit		
Z	Zero bit		
PD	Power-down bit		

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register op	erations
OPCODE d	f (FILE #)
d = 0 for destination W d = 1 for destination f f = 7-bit file register addre	255
Bit-oriented file register oper	ations
OPCODE b (BIT	#) f (FILE #)
b = 3-bit bit address f = 7-bit file register addre	ess
Literal and control operation	s
General	
13 8 7	<u> </u>
OPCODE	k (literal)
k = 8-bit immediate value	
CALL and GOTO instructions on	ly
13 11 10	0
OPCODE	k (literal)
k = 11-bit immediate value	e
MOVLP Instruction only	6 0
OPCODE	k (literal)
k = 7 hit immediate value	
13	540
OPCODE	k (literal)
k = 5-bit immediate value	
BRA instruction only 13 9 8	0
OPCODE	k (literal)
k = 9-bit immediate value	9
FSR Offset instructions	5 0
OPCODE n	k (literal)
n = appropriate FSR k = 6-bit immediate value	, , , , , , , , , , , , , , , , , , ,
FSR Increment instructions 13	3210
OPCODE	n m (mode)
n = appropriate FSR m = 2-bit mode value	
OPCODE only 13	0
OPCODI	E
L	

Mnemonic, Operands				14-Bit Opcode				Status	
		Description	Cycles	MSb		-	LSb	Affected	Notes
		BYTE-ORIENTED FILE	REGISTER OPE	RATIC	NS				
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	С	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	С	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
		BYTE ORIENTED	SKIP OPERATIO	ONS					
DECESZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
		BIT-ORIENTED FILE R	EGISTER OPER	RATIO	NS			L	
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
		BIT-ORIENTED S		NS				I	
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
LITERAL	OPERA	TIONS	1					r	
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

TABLE 21-3: PIC16LF1904/6/7 ENHANCED INSTRUCTION SET

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

Mnemonic, Operands		Description	Cycles	Cycles 14-Bit		Opcode		Status	Notos
		Description		MSb			LSb	Affected	NOLES
		CONTROL OPERA	TIONS						
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	_	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	_	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
	INHERENT OPERATIONS								
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	TO, PD	
NOP	_	No Operation	1	00	0000	0000	0000		
OPTION	_	Load OPTION_REG register with W	1	00	0000	0110	0010		
RESET	_	Software device Reset	1	00	0000	0000	0001		
SLEEP	_	Go into Standby mode	1	00	0000	0110	0011	TO, PD	
TRIS	f	Load TRIS register with W	1	00	0000	0110	Offf		
	C-COMPILER OPTIMIZED								
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec	1	00	0000	0001	0nmm	Z	2, 3
		modifier, mm							
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	kkkk	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec	1	00	0000	0001	1nmm		2, 3
		modifier, mm							
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk	kkkk		2

TABLE 21-3: PIC16LF1904/6/7 ENHANCED INSTRUCTION SET (CONTINUED)

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See Table in the MOVIW and MOVWI instruction descriptions.

21.2 Instruction Descriptions

ADDFSR	Add Literal to FSRn
Syntax:	[label] ADDFSR FSRn, k
Operands:	$\begin{array}{l} -32 \leq k \leq 31 \\ n \in \left[0, 1 \right] \end{array}$
Operation:	$FSR(n) + k \rightarrow FSR(n)$
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair.

FSRn is limited to the range 0000h -FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

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

ADDLW	Add literal and W
Syntax:	[<i>label</i>] ADDLW k
Operands:	$0 \leq k \leq 255$
Operation:	$(W) + k \to (W)$
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ANDWF	AND W with f		
Syntax:	[<i>label</i>] ANDWF f,d		
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$		
Operation:	(W) .AND. (f) \rightarrow (destination)		
Status Affected:	Z		
Description:	AND 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'.		

ADDWF	Add W and f		
Syntax:	[<i>label</i>] ADDWF f,d		
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$		
Operation:	(W) + (f) \rightarrow (destination)		
Status Affected:	C, DC, Z		
Description:	Add 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'.		

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



ADD W and CARRY bit to f

Syntax:	[<i>label</i>] ADDWFC f {,d}
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	$(W) + (f) + (C) \rightarrow dest$
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data mem- ory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

BCF	Bit Clear f
Syntax:	[label] BCF f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	$0 \rightarrow (f \le b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BTFSC	Bit Test f, Skip if Clear
Syntax:	[label]BTFSC f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	skip if (f) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BRA	Relative Branch	BT
Syntax:	[<i>label</i>]BRA label [<i>label</i>]BRA \$+k	Syl
Operands:	-256 ≤ label - PC + 1 ≤ 255 -256 ≤ k ≤ 255	Ор
Operation:	$(PC) + 1 + k \rightarrow PC$	Sta
Status Affected:	None	De
Description:	Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range.	

BRW	Relative Branch with W
Syntax:	[label] BRW
Operands:	None
Operation:	$(PC) + (W) \to PC$
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incre- mented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruc- tion.

BSF	Bit Set f
Syntax:	[<i>label</i>] BSF f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	$1 \rightarrow (f \le b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS	Bit Test f, Skip if Set
Syntax:	[label] BTFSS f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b < 7 \end{array}$
Operation:	skip if (f) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

CALL	Call Subroutine
Syntax:	[label] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruc- tion.

CLRWDT	Clear Watchdog Timer
Syntax:	[label] CLRWDT
Operands:	None
Operation:	$\begin{array}{l} \text{O0h} \rightarrow \text{WDT} \\ \text{0} \rightarrow \text{WDT prescaler,} \\ \text{1} \rightarrow \overline{\text{TO}} \\ \text{1} \rightarrow \overline{\text{PD}} \end{array}$
Status Affected:	TO, PD
Description:	CLRWDT instruction resets the Watch- dog Timer. It also resets the prescaler of the WDT. Status bits \overline{TO} and \overline{PD} are set.

CALLW	Subroutine Call With W
Syntax:	[label] CALLW
Operands:	None
Operation:	(PC) +1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
Status Affected:	None
Description:	Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a 2-cycle instruction.

COMF	Complement f
Syntax:	[label] COMF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	$(\overline{f}) \rightarrow (destination)$
Status Affected:	Z
Description:	The contents of register 'f' are com- plemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

CLRF	Clear f
Syntax:	[label] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	$\begin{array}{l} 00h \rightarrow (f) \\ 1 \rightarrow Z \end{array}$
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

DECF	Decrement f
Syntax:	[label] DECF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	(f) - 1 \rightarrow (destination)
Status Affected:	Z
Description:	Decrement 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'.

CLRW	Clear W
Syntax:	[label] CLRW
Operands:	None
Operation:	$\begin{array}{l} \text{00h} \rightarrow (\text{W}) \\ \text{1} \rightarrow \text{Z} \end{array}$
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

DECFSZ	Decrement f, Skip if 0
Syntax:	[label] DECFSZ f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	(f) - 1 \rightarrow (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decre- mented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[<i>label</i>] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	k → PC<10:0> PCLATH<6:3> → PC<14:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

INCFSZ	Increment f, Skip if 0
Syntax:	[label] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 \rightarrow (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incre- mented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

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

INCF	Increment f
Syntax:	[<i>label</i>] INCF f,d
Operands:	$\begin{array}{l} 0\leq f\leq 127\\ d\in [0,1] \end{array}$
Operation:	(f) + 1 \rightarrow (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incre- mented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

IORWF	Inclusive OR W with f
Syntax:	[<i>label</i>] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) \rightarrow (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with regis- ter 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

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:	$0 \le f \le 127$ $d \in [0,1]$	
Operation:	$(f < 7 >) \rightarrow C$	Operation:	$(f) \rightarrow (dest)$	
(f<6:0>) -	$(f < 6:0 >) \rightarrow dest < 7:1 >$	Status Affected:	Z	
Status Affected:	C, Z	Description:	The contents of register f is moved to a destination dependent upon the	
Description:	cription: 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 registe	C	er f 📕 Words:	1	
		Cycles:	1	
		Example:	MOVF FSR, 0	
LSRF	Logical Right Shift		After Instruction W = value in FSR register	
Syntax:	[<i>label</i>]LSRF f{,d}		Z = 1	

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'.

►C 0→ register f

ΜΟΥΙΨ	Move INDFn to W
Syntax:	[<i>label</i>] MOVIW ++FSRn [<i>label</i>] MOVIWFSRn [<i>label</i>] MOVIW FSRn++ [<i>label</i>] MOVIW FSRn [<i>label</i>] MOVIW k[FSRn]
Operands:	n ∈ [0,1] mm ∈ [00,01, 10, 11] -32 ≤ k ≤ 31
Operation:	$\begin{split} &\text{INDFn} \rightarrow \text{W} \\ &\text{Effective address is determined by} \\ &\text{FSR + 1 (preincrement)} \\ &\text{FSR - 1 (predecrement)} \\ &\text{FSR + k (relative offset)} \\ &\text{After the Move, the FSR value will be either:} \\ &\text{FSR + 1 (all increments)} \\ &\text{FSR - 1 (all decrements)} \\ &\text{Unchanged} \end{split}$
Status Affected:	Z

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.

Syntax:	[<i>label</i>] MOVLB k
Operands:	$0 \leq k \leq 15$
Operation:	$k \rightarrow BSR$
Status Affected:	None
Description:	The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP	Move literal to PCLATH	
Syntax:	[<i>label</i>] MOVLP k	
Operands:	$0 \le k \le 127$	
Operation:	$k \rightarrow PCLATH$	
Status Affected:	None	
Description:	The 7-bit literal 'k' is loaded into the PCLATH register.	
MOVLW	Move literal to W	
Syntax:	[<i>label</i>] MOVLW k	
Operands:	$0 \leq k \leq 255$	
Operation:	$k \rightarrow (W)$	
Status Affected:	None	
Description:	The 8-bit literal 'k' is loaded into W reg- ister. The "don't cares" will assemble as '0's.	
Words:	1	
Cycles:	1	
Example:	MOVLW 0x5A	
	After Instruction W = 0x5A	
MOVWF	Move W to f	
Syntax:	[<i>label</i>] MOVWF f	
Operands:	$0 \leq f \leq 127$	
Operation:	$(W) \rightarrow (f)$	
Status Affected:	None	
Description:	Move data from W register to register 'f'.	
Words:	1	
Cycles:	1	
Example:	MOVWF OPTION_REG	
	Before Instruction OPTION REG = 0xFF	

W = 0x4F After Instruction OPTION_REG = 0x4F W = 0x4F

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

Status Affected:

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

None

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.
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 RI flag of the PCON register.	
Status Affected:	None	
Description:	This instruction provides a way to execute a hardware Reset by soft- ware.	
RETFIE	Return from Interrupt	
------------------	---	--
Syntax:	[label] RETFIE	
Operands:	None	
Operation:	$\begin{array}{l} TOS \to PC, \\ 1 \to GIE \end{array}$	
Status Affected:	None	
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.	
Words:	1	
Cycles:	2	
Example:	RETFIE	
	After Interrupt PC = TOS GIE = 1	

RETURN	Return from Subroutine
Syntax:	[label] RETURN
Operands:	None
Operation:	$TOS \rightarrow PC$
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.

RETLW	Return with literal in W		Detete Left f through Correct	
Syntax:	[<i>label</i>] RETLW k	RLF	Rotate Left f through Carry	
Operands:	$0 \le k \le 255$	Syntax:	[<i>label</i>] RLF f,d	
Operation:	$k \rightarrow (W);$ TOS \rightarrow PC	Operands:	$0 \le f \le 127$ $d \in [0,1]$	
Status Affected:	None	Operation:	See description below	
Description:	The W register is loaded with the 8-bit	Status Affected:	C	
	literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.		The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is	
Words:	1		stored back in register T.	
Cycles:	2			
Example:	CALL TABLE;W contains table	Words:	1	
	;offset value;W now has table value	Cycles:	1	
TABLE	•	Example:	RLF REG1,0	
	<pre>ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;</pre>		Before Instruction REG1 = 1110 0110 C = 0 After Instruction REG1 = 1110 0110 W = 1100 1100 C = 1	
	After Instruction			
	W = value of k8			

PIC16LF1904/6/7

RRF	Rotate Right f through Carry
Syntax:	[<i>label</i>] RRF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
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:	[<i>label</i>] SUBLW k	
Operands:	$0 \leq k \leq 255$	
Operation:	$k - (W) \rightarrow (W)$	
Status Affected:	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

DC = 1

W<3:0> > k<3:0>

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

SLEEP	Enter Sleep mode
Syntax:	[label] SLEEP
Operands:	None
Operation:	$\begin{array}{l} 00h \rightarrow WDT, \\ 0 \rightarrow \underline{WDT} \text{ prescaler}, \\ 1 \rightarrow \underline{TO}, \\ 0 \rightarrow \overline{PD} \end{array}$
Status Affected:	TO, PD
Description:	The power-down Status bit, $\overline{\text{PD}}$ is cleared. Time-out Status bit, $\overline{\text{TO}}$ 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	IBWF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in \texttt{[0,1]} \end{array}$	
Operation:	$(f) - (W) \to (d$	estination)
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'.

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 register '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:	[label] XORWF f,d	
Operands:	$\begin{array}{l} 0\leq f\leq 127\\ d\in [0,1] \end{array}$	
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'.	

PIC16LF1904/6/7

NOTES:

22.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings^(†)

Ambient temperature under bias	40°C to +125°C
Storage temperature	65°C to +150°C
Voltage on pins with respect to Vss	
on VDD	-0.3V to +4.0V
on MCLR	-0.3V to +9.0V
on all other pins	0.3V to (VDD + 0.3V)
Total power dissipation ⁽¹⁾	
Maximum current	
out of Vss pin	
-40°C \leq TA \leq +85°C for industrial	300 mA
-40°C \leq TA \leq +125°C for extended	95 mA
into VDD pin	
-40°C \leq TA \leq +85°C for industrial	250 mA
-40°C \leq TA \leq +125°C for extended	
Clamp current, IK (VPIN < 0 or VPIN > VDD)	± 20 mA
Maximum output current	
sunk by any I/O pin	
sourced by any I/O pin	
Note 1: Power dissipation is calculated as follows: PDIS = VDD x {IDD $-\sum$ IOH} + \sum {($VDD - VOH) \times IOH + \Sigma(VOI \times IOL)$
+ NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may ca	ause permanent damage to the

device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

22.1 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $VDDMIN \le VDD \le VDDMAX$ Operating Temperature: $TA_MIN \le TA \le TA_MAX$

VDD — Op	perating Supply Voltage ⁽²⁾	
	VDDMIN (Fosc \leq 16 MHz)	+1.8V
	VDDMIN (Fosc \leq 20 MHz)	+2.3V
	VDDMAX	+3.6V
Та — Оре	erating Ambient Temperature Range	
Indu	dustrial Temperature	
	Та_міл	40°C
	Та_мах	+85°C
Exte	tended Temperature	
	Та_міл	40°C
	Та_мах	+125°C
Note 1:	Maximum current rating requires even load distribution across I/O pins. Maximum of limited by the device package power dissipation characterizations, see Section TA Considerations.	current rating may be ABLE 22-6: "Thermal
2.	See Parameter D001 DS Characteristics: Supply Voltage	
Ζ.	oce i arameter booi, bo characteristics. Oupply Voltage.	

PIC16LF1904/6/7





FIGURE 22-2: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



22.2 DC Characteristics

TABLE 22-1: SUPPLY VOLTAGE

PIC16LF1904/6/7				Standard Operating Conditions (unless otherwise stated)					
Param. No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions		
D001	Vdd		1.8 2.3	_ _	3.6 3.6	V V	$Fosc \le 16 \text{ MHz}$ $Fosc \le 20 \text{ MHz} (EC mode only)$		
D002*	Vdr	RAM Data Retention Voltage ⁽¹⁾	1.5	—	_	V	Device in Sleep mode		
D002A*	VPOR*	Power-on Reset Release Voltage	1.54	1.64	1.74	V			
D002B*	VPORR*	Power-on Reset Rearm Voltage	—	1.7	—	V	Device in Sleep mode		
D003	VADFVR	Fixed Voltage Reference Voltage for ADC, Initial Accuracy	6 7 7 8		4 4 6 6	%	1.024V, VDD ≥ 1.8V, 85°C 1.024V, VDD ≥ 1.8V, 125°C 2.048V, VDD ≥ 2.5V, 85°C 2.048V, VDD ≥ 2.5V, 125°C		
D003A	VCDAFVR	Fixed Voltage Reference Voltage for Comparator and DAC, Initial Accuracy	7 8 8 9		5 5 7 7	%	1.024V, VDD ≥ 1.8V, 85°C 1.024V, VDD ≥ 1.8V, 125°C 2.048V, VDD ≥ 2.5V, 85°C 2.048V, VDD ≥ 2.5V, 125°C		
D003B	VLCDFVR	Fixed Voltage Reference Voltage for LCD Bias, Initial Accuracy	9 9.5	_	9 9	%	$\begin{array}{l} 3.072 \text{V}, \ \text{VDD} \geq 3.6 \text{V}, \ 85^{\circ}\text{C} \\ 3.072 \text{V}, \ \text{VDD} \geq 3.6 \text{V}, \ 125^{\circ}\text{C} \end{array}$		
D003C*	TCVFVR	Temperature Coefficient, Fixed Voltage Reference	—	-130	—	ppm/°C			
D003D*	ΔVFVR/ ΔVIN	Line Regulation, Fixed Voltage Reference	—	0.270	_	%/V			
D004*	SVDD	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	_	V/ms	See Section 5.1 "Power-on Reset (POR)" for details.		

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

FIGURE 22-3: POR AND POR REARM WITH SLOW RISING VDD



TABLE 22-2: SUPPLY CURRENT (IDD)^(1,2)

PIC16LF1	904/6/7	Standard	Standard Operating Conditions (unless otherwise stated)						
Param	Device						Conditions		
No.	Characteristics	Min.	Typ†	Max.	Units	Vdd	Note		
	Supply Current (IDD) ^{(1,}	2)							
D010		_	58	75	μA	1.8	Fosc = 1 MHz		
		_	115	140	μA	3.0	EC Oscillator mode		
			133	176	μΑ	3.6	High Fower mode		
D011		_	130	200	μA	1.8	Fosc = 4 MHz		
		_	245	300	μΑ	3.0	EC Oscillator mode		
		—	290	350	μA	3.6			
D012		_	218	275	μΑ	1.8	Fosc = 500 kHz		
		—	283	375	μA	3.0	HFINTOSC mode		
		_	314	395	μΑ	3.6			
D013		_	233	325	μΑ	1.8	Fosc = 1 MHz		
		—	309	425	μA	3.0	HFINTOSC mode		
		_	347	475	μA	3.6			
D014		_	305	360	μA	1.8	Fosc = 4 MHz		
		_	433	520	μA	3.0	HFINTOSC mode		
		_	500	600	μΑ	3.6			
D015		_	395	480	μA	1.8	Fosc = 8 MHz		
		_	600	720	μΑ	3.0	HFINTOSC mode		
		_	700	850	μA	3.6	-		
D016		_	567	670	μA	1.8	Fosc = 16 MHz		
		_	915	1100	μA	3.0	HFINTOSC mode		
		_	1087	1300	μA	3.6	-		
D017		_	2.7	7.2	μA	1.8	Fosc = 31 kHz		
		_	4.5	9.7	μA	3.0	LFINTOSC mode		
		_	5.2	12.0	μA	3.6	$-40^{\circ}C \le IA \le +125^{\circ}C$		
D017A		_	2.7	6.5	μA	1.8	Fosc = 31 kHz		
		_	4.5	9.0	μA	3.0	LFINTOSC mode		
		_	5.2	11.0	μA	3.6	$-40^{\circ}C \le TA \le +85^{\circ}C$		
D018		1_	2.4	6.7	μA	1.8	Fosc = 32 kHz		
		—	4.2	9.2	μA	3.0	EC Oscillator mode, Low-Power mode		
		_	4.8	11.5	μΑ	3.6	$-40^{\circ}C \le TA \le +125^{\circ}C$		
D018A		_	2.4	6.0	μA	1.8	Fosc = 32 kHz		
		_	4.2	8.5	μA	3.0	EC Oscillator mode, Low-Power mode		
		<u> </u>	4.8	10.5	μA	3.6	$-40^{\circ}C \le TA \le +85^{\circ}C$		
		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 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: FVR and BOR are disabled.

PIC16LF19	04/6/7	Standar	d Operat	ing Cond	itions (unl	ess othe	rwise stat	ed)
Param	Dovice Characteristics	Min	Tunt	Max.	Max.	Unito	Conditions	
No.	Device Characteristics	WIII.	турт	+85°C	+125°C	Units	Vdd	Note
	Power-down Base Current	(IPD) ⁽²⁾						
D023		_	0.15	1.0	3.0	μA	1.8	WDT, BOR, FVR, and T1OSC
			0.16	2.0	4.0	μA	3.0	disabled, all Peripherals Inactive
		—	0.65	3.0	5.0	μA	3.6	
D024			0.27	2.0	4.0	μA	1.8	WDT Current (Note 1)
			0.56	3.0	5.0	μA	3.0	
			0.75	4.0	6.0	μA	3.6	
D025			17.5	31	35	μA	1.8	FVR current
			17.7	33	38	μA	3.0	
			17.8	35	41	μA	3.6	
D026			0.15	2.30	3.56	μA	3.0	LPBOR current
			0.21	3.40	4.70	μA	3.6	
D027			7.0	10	12	μA	3.0	BOR Current
			7.5	12	14	μA	3.6	
D028			0.50	2.0	4.0	μA	1.8	T1OSC Current
			0.60	3.0	5.0	μA	3.0	
		—	0.70	4.0	6.0	μA	3.6	
D029			0.40	2.0	4.0	μA	1.8	ADC Current (Note 1, Note 3),
			0.70	3.0	5.0	μA	3.0	no conversion in progress
		—	0.90	4.0	6.0	μA	3.6	
D030			—	250		μA	1.8	ADC Current (Note 1, Note 3),
			_	250	_	μA	3.0	conversion in progress
				250		μA	3.6	
D031	LCD Bias Ladder							
	Low power	_	1	2	6	μA	1.8	
	Medium Power	_	10	13	21	μA	3.0	
	High Power	—	100	111	120	μA	3.6	

TABLE 22-3: POWER-DOWN CURRENTS (IPD)

* 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 peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral △ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

3: A/D oscillator source is FRC.

TABLE 22-4: I/O PORTS

	DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)					
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions		
	VIL	Input Low Voltage							
		I/O PORT:				_			
D032		with TTL buffer	—	_	0.15 VDD	V	$1.8V \leq V\text{DD} \leq 3.6V$		
D033		with Schmitt Trigger buffer	—	_	0.2 Vdd	V	$1.8V \leq V\text{DD} \leq 3.6V$		
D034		MCLR, OSC1	—	_	0.2 Vdd	V			
	Vih	Input High Voltage							
		I/O ports:							
D040		with TTL buffer	0.25 VDD + 0.8	_	—	V	$1.8V \le V\text{DD} \le 3.6V$		
D041		with Schmitt Trigger buffer	0.8 VDD	_		V	$1.8V \leq V\text{DD} \leq 3.6V$		
D042		MCLR	0.8 VDD	_		V			
	lı∟	Input Leakage Current ⁽²⁾							
D060		I/O ports	_	± 5	± 125	nA	Vss ≤ VPIN ≤ VDD, Pin at high-impedance @ 85°C		
				± 5	± 1000	nA	125°C		
D061		MCLR ⁽³⁾	—	± 50	± 200	nA	$Vss \le VPIN \le VDD @ 85^{\circ}C$		
	IPUR	Weak Pull-up Current							
D070*			25	100	200	μΑ	VDD = 3.3V, VPIN = VSS		
	Vol	Output Low Voltage							
D080		I/O ports	_		0.6	V	IOL = 6mA, VDD = 3.3V IOL = 1.8mA, VDD = 1.8V		
	Voн	Output High Voltage							
D090		I/O ports	Vdd - 0.7	_	_	V	IOH = 3mA, VDD = 3.3V IOH = 1mA, VDD = 1.8V		
		Capacitive Loading Specs on	Output Pins						
D101*	CIO	All I/O pins	—		50	pF			

* 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: Negative current is defined as current sourced by the pin.

DC CHA	DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)					
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions		
		Program Memory Programming Specifications							
D110	VIHH	Voltage on MCLR/VPP/RE3 pin	8.0	_	9.0	V	(Note 2)		
D111	IDDP	Supply Current during Programming	—	_	10	mA			
D112		VDD for Bulk Erase	2.7	—	VDD max.	V			
D113	Vpew	VDD for Write or Row Erase	Vdd min.	—	VDD max.	V			
D114	IPPPGM	Current on MCLR/VPP during Erase/Write	—	—	1.0	mA			
D115	IDDPGM	Current on VDD during Erase/Write	—		5.0	mA			
		Program Flash Memory							
D121	Eр	Cell Endurance	1K	10K	_	E/W	-40°C to +85°C (Note 1)		
D122	Vpr	VDD for Read	VDD min.	—	V _{DD} max.	V			
D123	Tiw	Self-timed Write Cycle Time	_	2	2.5	ms			
D124	TRETD	Characteristic Retention	40	_	_	Year	Provided no other specifications are violated		

TABLE 22-5: MEMORY PROGRAMMING REQUIREMENTS

† 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: Self-write and Block Erase.

2: Required only if single-supply programming is disabled.

TABLE 22-6: THERMAL CONSIDERATIONS

Standar	Standard Operating Conditions (unless otherwise stated)									
Param No.	Sym.	Characteristic	Тур.	Units	Conditions					
TH01	θJA	Thermal Resistance Junction to Ambient	60	°C/W	28-pin SPDIP package					
			80	°C/W	28-pin SOIC package					
			90	°C/W	28-pin SSOP package					
			27.5	°C/W	28-pin UQFN 4x4mm package					
			47.2	°C/W	40-pin PDIP package					
			41.0	°C/W	40-pin UQFN 5x5mm package					
			46.0	°C/W	44-pin TQFP package					
TH02	θJC	Thermal Resistance Junction to Case	31.4	°C/W	28-pin SPDIP package					
			24	°C/W	28-pin SOIC package					
			24	°C/W	28-pin SSOP package					
			24	°C/W	28-pin UQFN 4x4mm package					
			24.7	°C/W	40-pin PDIP package					
			50.5	°C/W	40-pin UQFN 5x5mm package					
			14.5	°C/W	44-pin TQFP package					
TH03	TJMAX	Maximum Junction Temperature	150	°C						
TH04	PD	Power Dissipation	_	W	PD = PINTERNAL + PI/O					
TH05	PINTERNAL	Internal Power Dissipation		W	PINTERNAL = IDD x VDD ⁽¹⁾					
TH06	Pi/o	I/O Power Dissipation	_	W	$PI/O = \Sigma (IOL * VOL) + \Sigma (IOH * (VDD - VOH))$					
TH07	Pder	Derated Power	—	W	Pder = PDmax (Τj - Τa)/θja ⁽²⁾					

Standard Operating Conditions (unless otherwise stated)

Note 1: IDD is current to run the chip alone without driving any load on the output pins.

2: TA = Ambient Temperature

3: T_J = Junction Temperature

22.3 AC Characteristics

The timing parameter symbols have been created with one of the following formats:

- 1. TppS2ppS
- 2. TppS

Т			
F	Frequency	Т	Time
Lowerc	ase letters (pp) and their meanings:		
рр			
сс	CCP1	osc	OSC1
ck	CLKOUT	rd	RD
cs	CS	rw	RD or WR
di	SDI	sc	SCK
do	SDO	SS	SS
dt	Data in	tO	ТОСКІ
io	I/O PORT	t1	T1CKI
mc	MCLR	wr	WR
Upperc	ase letters and their meanings:		
S			
F	Fall	Р	Period
Н	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 22-4: LOAD CONDITIONS



TABLE 22-7: CLOCK OSCILLATOR TIMING REQUIREMENTS

Standard	Operating	Conditions	unless	otherwise stated)	
otaniaana	oporating		(41110000	0	

Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
OS01	Fosc	External CLKIN Frequency ⁽¹⁾	DC	_	0.5	MHz	External Clock (ECL)
			DC	—	4	MHz	External Clock (ECM)
			DC	—	20	MHz	External Clock (ECH)
OS02	Tosc	External CLKIN Period ⁽¹⁾	50	_	8	ns	External Clock (EC)
OS03	TCY	Instruction Cycle Time ⁽¹⁾	200	TCY	DC	ns	Tcy = 4/Fosc

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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

TABLE 22-8: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	Sym.	Characteristic	Freq. Tolerance	Min.	Тур†	Max.	Units	Conditions
OS08	HFosc	Internal Calibrated HFINTOSC Frequency ⁽²⁾	±8% ±6.5%		16 16		MHz MHz	$0^{\circ}C \le TA \le +85^{\circ}C$ VDD = 3.0V @ +25°C
OS10A*	TIOSC ST	HFINTOSC 16 MHz Oscillator Wake-up from Sleep Start-up Time	_		5 5	15 15	μs μs	VDD = 2.0V, -40°C to +85°C VDD = 3.0V, -40°C to +85°C

* 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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: To ensure these oscillator frequency tolerances, VDD and Vss must be capacitively decoupled as close to the device as possible. 0.1 μ F and 0.01 μ F values in parallel are recommended.



TABLE 22-9: CL	KOUT AND I/O	TIMING PARAMETERS
----------------	--------------	-------------------

Standar	rd Operating	g Conditions (unless otherwise stated)				-	
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
OS11	TosH2ckL	Fosc↑ to CLKOUT↓ ⁽¹⁾			70	ns	VDD = 3.3-5.0V
OS12	TosH2ckH	Fosc↑ to CLKOUT↑ ⁽¹⁾	—		72	ns	VDD = 3.3-5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid ⁽¹⁾	_		20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑ ⁽¹⁾	Tosc + 200 ns	_	_	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	VDD = 3.3-5.0V
OS16	TosH2iol	Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	_	_	ns	VDD = 3.3-5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—		ns	
OS18	TioR	Port output rise time	—	40	72	ns	VDD = 1.8V
			—	15	32		VDD = 3.3-5.0V
OS19	TioF	Port output fall time	_	28	55	ns	VDD = 1.8V
			—	15	30		VDD = 3.3-5.0V
OS20*	Tinp	INT pin input high or low time	25	_	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	_		ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

Note 1: Measurements are taken in EC mode where CLKOUT output is 4 x Tosc.

PIC16LF1904/6/7













TABLE 22-10: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS

Standa	Standard Operating Conditions (unless otherwise stated)								
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions		
30	ТмсL	MCLR Pulse Width (low)	2 5	_		μS μS	VDD = 3.0V, -40°C to +85°C VDD = 3.0V		
31	FWDTLP	Low Frequency Internal Oscillator Frequency	19	33	52	kHz			
32	Tost	Oscillator Start-up Timer Period ⁽¹⁾		1024		Tosc	(Note 2)		
33*	TPWRT	Power-up Timer Period, PWRTE = 0		2048		Tosc	Clocked by LFINTOSC		
34*	Tioz	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	_	—	2.0	μS			
35	VBOR	Brown-out Reset Voltage: BORV = 0 BORV = 1	2.55 1.80	2.70 1.90	2.85 2.05	V V			
35A*	VHYST	Brown-out Reset Hysteresis	25 —	50 —	75 100	mV mV	-40°C to +85°C -40°C to +125°C		
35B*	TBORDC	Brown-out Reset DC Response Time	1	3	5 10	μS μS	$VDD \le VBOR$, -40°C to +85°C $VDD \le VBOR$		
35C	TBORAC	Brown-out Reset AC Response Time	_	100		ns	Transient Response immunity for a noise spike that goes from VDD to VSS and back with 10 ns rise and fall times. Guidance only.		
36	TFVRS	Fixed Voltage Reference Turn-on Time	—	—	5	μS	Turn on to specified stability		
37	Vlpbor	Low-Power Brown-out Reset Voltage	1.85	1.95	2.10	V	-40°C to +85°C		
38*	VZPHYST	Zero-Power Brown-out Reset Hysteresis	0	25	60	mV	-40°C to +85°C		
39*	TZPBPW	Zero-Power Brown-out Reset AC Response Time for BOR detection	10		500	nVs	VDD \leq VBOR, -40°C to +85°C		

* 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:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.
 - **2:** Period of the slower clock.
 - 3: To ensure these voltage tolerances, VDD and Vss must be capacitively decoupled as close to the device as possible. 0.1 μ F and 0.01 μ F values in parallel are recommended.

PIC16LF1904/6/7





TABLE 22-11:	TIMER0 AND TIMER1	EXTERNAL CLOCK	KREQUIREMENTS
--------------	-------------------	----------------	----------------------

Standar	Standard Operating Conditions(unless otherwise stated)								
Param No.	Sym.		Characteristic	•	Min.	Тур†	Max.	Units	Conditions
40*	T⊤0H	T0CKI High F	Pulse Width	No Prescaler	0.5 Tcy + 20	—	_	ns	
				With Prescaler	10	—		ns	
41*	T⊤0L	T0CKI Low F	ulse Width	No Prescaler	0.5 Tcy + 20	—		ns	
				With Prescaler	10	—		ns	
42*	T⊤0P	T0CKI Period	1		Greater of: 20 or <u>Tcy + 40</u> N	—	—	ns	N = prescale value (2, 4,, 256)
45*	T⊤1H	T1CKI High	Synchronous, N	lo Prescaler	0.5 Tcy + 20	—	_	ns	
		Time	Synchronous, with Prescaler		15	—	_	ns	
			Asynchronous		30	—	—	ns	
46*	T⊤1L	T1CKI Low	Synchronous, N	lo Prescaler	0.5 Tcy + 20	—	_	ns	
		Time	Synchronous, v	vith Prescaler	15	_	_	ns	
			Asynchronous		30	_	_	ns	
47*	T⊤1P	T1CKI Input Period	Synchronous		Greater of: 30 or <u>Tcy + 40</u> N	—	_	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous		60	_		ns	
48	FT1	Timer1 Oscill (oscillator en	ator Input Frequency Range abled by setting bit T1OSCEN)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	Delay from E Increment	xternal Clock Ec	lge to Timer	2 Tosc	—	7 Tosc	—	Timers in Sync mode

* 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.

TABLE 22-12: PIC16LF1904/6/7 A/D CONVERTER (ADC) CHARACTERISTICS:

0	perating	Conditions	unless	otherwise	stated)	
-	porating	o o mantio no i	(41110000	011101 11100	otatoa,	

VDD = 3	.0V, TA	= 25°C					
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
AD01	NR	Resolution	—	_	10	bit	
AD02	EIL	Integral Error		±1	±1.7	LSb	VREF = 3.0V
AD03	Edl	Differential Error	_	±1	±1	LSb	No missing codes VREF = 3.0V
AD04	EOFF	Offset Error		±1	±2	LSb	VREF = 3.0V
AD05	Egn	Gain Error	_	±1	±1.5	LSb	VREF = 3.0V
AD06	VREF	Reference Voltage ⁽³⁾	1.8	_	Vdd	V	VREF = (VRPOS - VRNEG)
AD07	VAIN	Full-Scale Range	Vss	_	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	_		10	kΩ	Can go higher if external 0.01µF capacitor is present on input pin.

* 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: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: ADC VREF is from external VREF, VDD pin or FVREF, whichever is selected as reference input.

4: When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

TABLE 22-13: PIC16LF1902/3 A/D CONVERSION REQUIREMENTS

Standar	Standard Operating Conditions (unless otherwise stated)									
Param. No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions			
AD130*	TAD	ADC Clock Period (TADC)	1.0	—	6.0	μS	Fosc-based			
		ADC Internal FRC Oscillator Period (TFRC)	1.0	2.0	6.0	μS	ADCS<2:0> = x11 (ADC FRC mode)			
AD131	TCNV	Conversion Time (not including Acquisition Time) ⁽¹⁾	—	11	—	TAD	Set GO/DONE bit to conversion complete			
AD132*	TACQ	Acquisition Time		5.0	Ι	μS				
AD133*	THCD	Holding Capacitor Disconnect Time		1/2 TAD 1/2 TAD + 1TCY	_		Fosc-based ADCS<2:0> = x11 (ADC FRC mode)			

* 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 ADRES register may be read on the following TCY cycle.









23.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

Graphs and charts are not available at this time.

24.0 DEVELOPMENT SUPPORT

The PIC[®] microcontrollers (MCU) and dsPIC[®] digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- · Integrated Development Environment
- MPLAB[®] X IDE Software
- · Compilers/Assemblers/Linkers
 - MPLAB XC Compiler
 - MPASM[™] Assembler
 - MPLINK[™] Object Linker/ MPLIB[™] Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- · Simulators
 - MPLAB X SIM Software Simulator
- · Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
 - MPLAB ICD 3
 - PICkit™ 3
- Device Programmers
- MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

24.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows[®], Linux and Mac $OS^{®}$ X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- · Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- · Call graph window
- Project-Based Workspaces:
- · Multiple projects
- Multiple tools
- · Multiple configurations
- · Simultaneous debugging sessions

File History and Bug Tracking:

- · Local file history feature
- Built-in support for Bugzilla issue tracker

24.2 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

24.3 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

24.4 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

24.5 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

24.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

24.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

24.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a highspeed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

24.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a fullspeed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming[™] (ICSP[™]).

24.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

24.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM[™] and dsPICDEM[™] demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ[®] security ICs, CAN, IrDA[®], PowerSmart battery management, SEEVAL[®] evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

24.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent[®] and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika[®]

25.0 PACKAGING INFORMATION

25.1 Package Marking Information



Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Package Marking Information (Continued)



25.2 Package Details

The following sections give the technical details of the packages.

28-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

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



	Units		INCHES	
Dimension	n Limits	MIN	NOM	MAX
Number of Pins	Ν		28	
Pitch	е		.100 BSC	
Top to Seating Plane	Α	-	-	.250
Molded Package Thickness	A2	.125	-	.195
Base to Seating Plane	A1	.015	-	-
Shoulder to Shoulder Width	E	.590	-	.625
Molded Package Width	E1	.485	-	.580
Overall Length	D	1.380	-	1.565
Tip to Seating Plane	L	.115	-	.200
Lead Thickness	С	.008	-	.015
Upper Lead Width	b1	.030	-	.070
Lower Lead Width	b	.014	_	.022
Overall Row Spacing §	eB	-	_	.700

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-079B

28-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 C04-052C Sheet 1 of 2

28-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





	MILLIMETERS				
Dimension	Limits	MIN	NOM	MAX	
Number of Pins	Z		28		
Pitch	е		1.27 BSC		
Overall Height	A	-	-	2.65	
Molded Package Thickness	A2	2.05	-	-	
Standoff §	A1	0.10	-	0.30	
Overall Width	E		10.30 BSC		
Molded Package Width	E1		7.50 BSC		
Overall Length	D		17.90 BSC		
Chamfer (Optional)	h	0.25	-	0.75	
Foot Length	L	0.40	-	1.27	
Footprint	L1		1.40 REF		
Lead Angle	Θ	0°	-	I	
Foot Angle	φ	0°	-	8°	
Lead Thickness	С	0.18 - 0.33			
Lead Width	b	0.31 - 0.51			
Mold Draft Angle Top	α	5° - 15°			
Mold Draft Angle Bottom	β	5°	-	15°	

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M BSC: Basic Dimension. Theoretically exact value shown without tolerances. REF: Reference Dimension, usually without tolerance, for information purposes only.
- 5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

28-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



RECOMMENDED LAND PATTERN

	N	IILLIMETER	S	
Dimension	Limits	MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	С	9.40		
Contact Pad Width (X28)	Х			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	stance Between Pads G			

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A



For the most current package drawings, please see the Microchip Packaging Specification located at

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

http://www.microchip.com/packaging

	Units	MILLIMETERS				
	Dimension Limits	MIN NOM MAX				
Number of Pins	N		28	•		
Pitch	е		0.65 BSC			
Overall Height	A	-	-	2.00		
Molded Package Thickness	A2	1.65	1.75	1.85		
Standoff	A1	0.05	-	-		
Overall Width	E	7.40	7.80	8.20		
Molded Package Width	E1	5.00	5.30	5.60		
Overall Length	D	9.90	10.20	10.50		
Foot Length	L	0.55	0.75	0.95		
Footprint	L1		1.25 REF			
Lead Thickness	С	0.09	-	0.25		
Foot Angle	ф	0°	4°	8°		
Lead Width	b	0.22	-	0.38		

Notes:

Note:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- 3. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



	Units	Units MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	С		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



Microchip Technology Drawing C04-152A Sheet 1 of 2

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



	MILLIMETERS			
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	е	0.40 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.127 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated.

- 3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances. REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-152A Sheet 2 of 2
28-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) - 4x4 mm Body [UQFN] With 0.40 mm Contact Length





	MILLIMETERS			
Dimension	MIN	NOM	MAX	
Contact Pitch	E	E 0.40 BSC		
Optional Center Pad Width	W2			2.35
Optional Center Pad Length	T2			2.35
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X28)	X1			0.20
Contact Pad Length (X28)	Y1			0.80
Distance Between Pads	0.20			

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2152A

40-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

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



	Units	INCHES			
Dimensio	n Limits	MIN	NOM	MAX	
Number of Pins	Ν		40		
Pitch	е	.100 BSC			
Top to Seating Plane	Α	-	-	.250	
Molded Package Thickness	A2	.125	-	.195	
Base to Seating Plane	A1	.015	-	—	
Shoulder to Shoulder Width	E	.590	-	.625	
Molded Package Width	E1	.485	-	.580	
Overall Length	D	1.980	-	2.095	
Tip to Seating Plane	L	.115	-	.200	
Lead Thickness	С	.008	-	.015	
Upper Lead Width	b1	.030	-	.070	
Lower Lead Width	b	.014	_	.023	
Overall Row Spacing §	eB	_	_	.700	

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. § Significant Characteristic.

3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.

4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-016B



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

Microchip Technology Drawing C04-076C Sheet 1 of 2

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

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



	MILLIMETERS				
Dimension Limits		MIN	NOM	MAX	
Number of Leads	44				
Lead Pitch	е		0.80 BSC		
Overall Height	Α	-	-	1.20	
Standoff	A1	0.05	-	0.15	
Molded Package Thickness	A2	0.95	1.00	1.05	
Overall Width	12.00 BSC				
Molded Package Width	E1	10.00 BSC			
Overall Length	D	12.00 BSC			
Molded Package Length	D1	10.00 BSC			
Lead Width	b	0.30	0.37	0.45	
Lead Thickness	С	0.09	-	0.20	
Lead Length	L	0.45	0.60	0.75	
Footprint	L1	1.00 REF			
Foot Angle	θ	0°	3.5°	7°	

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Exact shape of each corner is optional.

3. Dimensioning and tolerancing per ASME Y14.5M

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

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076C Sheet 2 of 2

44-Lead Plastic Thin Quad Flatpack (PT) - 10X10X1 mm Body, 2.00 mm Footprint [TQFP]

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



RECOMMENDED LAND PATTERN

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

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2076B

40-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) – 5x5x0.5 mm Body [UQFN]

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



Microchip Technology Drawing C04-156A Sheet 1 of 2

40-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) – 5x5x0.5 mm Body [UQFN]

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



	MILLIMETERS			
Dimensior	MIN	NOM	MAX	
Number of Pins	N	40		
Pitch	е	0.40 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.127 REF		
Overall Width	E	5.00 BSC		
Exposed Pad Width	E2	3.60	3.70	3.80
Overall Length	D	5.00 BSC		
Exposed Pad Length	D2	3.60	3.70	3.80
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated.

3. Dimensioning and tolerancing per ASME Y14.5M.

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

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-156A Sheet 2 of 2

40-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) - 5x5 mm Body [UQFN]

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



	MILLIMETERS				
Dimension	MIN	NOM	MAX		
Contact Pitch	E		0.40 BSC		
Optional Center Pad Width	W2			3.80	
Optional Center Pad Length	T2			3.80	
Contact Pad Spacing	C1		5.00		
Contact Pad Spacing	C2		5.00		
Contact Pad Width (X40)	X1			0.20	
Contact Pad Length (X40)	Y1			0.75	
Distance Between Pads	G	0.20			

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2156B

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (03/2011)

Original release.

Revision B (08/2014)

Added Tables 9-3 and 9-4.

Updated PIC16LF190X Family Types Table.

Updated Equation 15-1; Example 3-2; Figures 1 through 5, 6-4, 15-1, 15-4, 19-7 22-2, 22-8; Registers 4-1, 4-2, 11-17, 15-1; Sections 4.1, 6.2.2.2, 10.0, 13.0, 15.1.3, 18.1.1.2, 18.1.1.3, 18.1.1.7, 18.1.2.9, 18.1.2.10, 18.2, 18.4.1.2, 19.1, 19.4.5, 22.0, 22.1, 22.5, 22.6, 22.7, 22-8, 22-10; Tables 1, 3-1, 3-3, 3-5, 5-1, 9-2, 10-3, 10-4, 17-2, 19-1, 22-2, 22-3, 22-4, 22-5, 22-6, 22-7, 22-8, 22-12.

Updated Package Marking Information.

Removed Section 18.1.2.3: Receive Data Polarity and Section 18.4.1.4: Data Polarity.

Data sheet went from Preliminary to Final data sheet.

Revision C (01/2016)

Added 'Memory' section and updated the 'PIC16LF190X Family Types' table; Other minor corrections.

Revision D (05/2016)

Minor changes to Figure 22-2, Table 22-8, and Table 22-10, in Electrical Specifications chapter; Updated Packaging: 28L SOIC, 44L TQFP, 28L UQFN, 40L UQFN. Removed Table 19-7, LCD Worksheet.

THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- Product Support Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- General Technical Support Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- Business of Microchip Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- · Local Sales Office
- Field Application Engineer (FAE)
- · Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X] ⁽¹⁾ ₋ Ț Tape and Reel Option	X Temperature Range	/XX Package	XXX Pattern	Ex a)	cample PIC Tape Indu UQF	25: 16LF1904T - I/MV 301 e and Reel, istrial temperature, FN package,
Device:	PIC16LF1904, F	PIC16LF1906, P	IC16LF1907		b)	QTF PIC Indu PDI	² pattern #301 16LF1906 - I/P Istrial temperature P package
Tape and Reel Option:	Blank = Stand T = Tape a	ard packaging (ti and Reel ⁽¹⁾	ube or tray)		c)	PIC Exte SSC	16LF1906 - E/SS ended temperature,)P package
Temperature Range:	$ = -40^{\circ}(2)$ E = -40^{\circ}(2)	C to +85°C (C to +125°C (Industrial) Extended)				
Package:	MV = UQF P = PDIP PT = TQFI SO = SOIC SS = SSO	N (4x4x0.5) > (44-pin) P			No	ote 1:	Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check
Pattern:	QTP, SQTP, Co (blank otherwise	de or Special Re e)	quirements				with your Microchip Sales Office for package availability with the Tape and Reel option.

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

- · Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

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

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEEL0Q® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV = ISO/TS 16949=

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2011-2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0545-0



Worldwide Sales and Service

AMERICAS

Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/ support

Web Address: www.microchip.com

Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455

Austin, TX Tel: 512-257-3370

Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

Cleveland Independence, OH Tel: 216-447-0464 Fax: 216-447-0643

Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924

Detroit Novi, MI Tel: 248-848-4000

Houston, TX Tel: 281-894-5983

Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453

Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608

New York, NY Tel: 631-435-6000

San Jose, CA Tel: 408-735-9110

Canada - Toronto Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office Suites 3707-14, 37th Floor Tower 6, The Gateway

Harbour City, Kowloon Hong Kong Tel: 852-2943-5100 Fax: 852-2401-3431

Australia - Sydney Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing Tel: 86-10-8569-7000 Fax: 86-10-8528-2104

China - Chengdu Tel: 86-28-8665-5511 Fax: 86-28-8665-7889

China - Chongqing Tel: 86-23-8980-9588 Fax: 86-23-8980-9500

China - Dongguan Tel: 86-769-8702-9880

China - Hangzhou Tel: 86-571-8792-8115 Fax: 86-571-8792-8116

China - Hong Kong SAR Tel: 852-2943-5100 Fax: 852-2401-3431

China - Nanjing Tel: 86-25-8473-2460 Fax: 86-25-8473-2470

China - Qingdao Tel: 86-532-8502-7355 Fax: 86-532-8502-7205

China - Shanghai Tel: 86-21-5407-5533 Fax: 86-21-5407-5066

China - Shenyang Tel: 86-24-2334-2829 Fax: 86-24-2334-2393

China - Shenzhen Tel: 86-755-8864-2200 Fax: 86-755-8203-1760

China - Wuhan Tel: 86-27-5980-5300 Fax: 86-27-5980-5118

China - Xian Tel: 86-29-8833-7252 Fax: 86-29-8833-7256 ASIA/PACIFIC China - Xiamen

Tel: 86-592-2388138 Fax: 86-592-2388130

China - Zhuhai Tel: 86-756-3210040 Fax: 86-756-3210049

India - Bangalore Tel: 91-80-3090-4444 Fax: 91-80-3090-4123

India - New Delhi Tel: 91-11-4160-8631 Fax: 91-11-4160-8632

India - Pune Tel: 91-20-3019-1500

Japan - Osaka Tel: 81-6-6152-7160 Fax: 81-6-6152-9310

Japan - Tokyo Tel: 81-3-6880- 3770 Fax: 81-3-6880-3771

Korea - Daegu Tel: 82-53-744-4301 Fax: 82-53-744-4302

Korea - Seoul Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

Malaysia - Kuala Lumpur Tel: 60-3-6201-9857 Fax: 60-3-6201-9859

Malaysia - Penang Tel: 60-4-227-8870 Fax: 60-4-227-4068

Philippines - Manila Tel: 63-2-634-9065 Fax: 63-2-634-9069

Singapore Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan - Hsin Chu Tel: 886-3-5778-366 Fax: 886-3-5770-955

Taiwan - Kaohsiung Tel: 886-7-213-7828

Taiwan - Taipei Tel: 886-2-2508-8600 Fax: 886-2-2508-0102

Thailand - Bangkok Tel: 66-2-694-1351 Fax: 66-2-694-1350

EUROPE

Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393

Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829

France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany - Dusseldorf Tel: 49-2129-3766400

Germany - Karlsruhe Tel: 49-721-625370

Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44

Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781

Italy - Venice Tel: 39-049-7625286

Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340

Poland - Warsaw Tel: 48-22-3325737

Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

Sweden - Stockholm Tel: 46-8-5090-4654

UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820

07/14/15