

PIC16(L)F1615/9

14/20-Pin, 8-Bit Flash Microcontroller

Description

PIC16(L)F1615/9 microcontrollers deliver on-chip features that are unique to the design for embedded control of small motors and general purpose applications in 14/20-pin count packages. Features like 10-bit A/D, CCP, 24-bit SMT and Zero-Cross Detection offer an excellent solution to the variety of applications. The product family also has a CRC+ memory scan and Windowed WDT to support safety-critical systems in home appliances, white goods and other end equipment.

Core Features

- C Compiler Optimized RISC Architecture
- Only 49 Instructions
- · Operating Speed:
- DC 32 MHz clock input
- 125 ns minimum instruction cycle
- Interrupt Capability
- 16-Level Deep Hardware Stack
- One 8-Bit Timer
- Four 16-bit Timers
- Low Current Power-on Reset (POR)
- Configurable Power-up Timer (PWRT)
- · Brown-out Reset (BOR) with Selectable Trip Point
- Windowed Watchdog Timer (WWDT):
 - Variable prescaler selection
 - Variable window size selection
 - All sources configurable in hardware or software

Memory

- 8 KW Flash Program Memory
- 1024 Bytes Data SRAM
- · Direct, Indirect and Relative Addressing modes
- High-Endurance Flash Data Memory (HEF):
 - 128 B of nonvolatile data storage
 - 100K erase/write cycles

Operating Characteristics

- Operating Voltage Range:
 - 1.8V to 3.6V (PIC16LF1615/9)
 - 2.3V to 5.5V (PIC16F1615/9)
- Temperature Range:
 - Industrial: -40°C to 85°C
 - Extended: -40°C to 125°C

eXtreme Low-Power (XLP) Features

- Sleep mode: 50 nA @ 1.8V, typical
- Watchdog Timer: 500 nA @ 1.8V, typical
- · Operating Current:
 - 8 uA @ 32 kHz, 1.8V, typical
 - 32 uA/MHz @ 1.8V, typical

Digital Peripherals

- Configurable Logic Cell (CLC):
 - Four CLCs
 - Integrated combinational and sequential logic
- · Complementary Waveform Generator (CWG):
 - Rising and falling edge dead-band control
 - Full-bridge, half-bridge, 1-channel drive
 - Multiple signal sources
- Two Capture/Compare/PWM (CCP) modules
- PWM: Two 10-bit Pulse-Width Modulators
- Two Signal Measurement Timers (SMT):
 - 24-bit timer/counter with prescaler
 - Multiple gate and clock inputs
- Angular Timer:
 - Single pulse
- Multiple pulses with missing pulse recovery
- 8-Bit Timers (TMR2+HLT/4/6):
 - Up to 3 Timer2/4/6 with Hardware Limit Timer (HLT)
 - Monitors Fault Conditions: Stall, Stop, etc.
 - Multiple modes
 - 8-bit timer/counter with prescaler
 - 8-bit period register and postscaler
 - Asynchronous H/W Reset sources
- Math Accelerator with Proportional-Integral-Derivative (PID):
 - Four operation modes
 - Add and multiply
 - Simple multiplier
 - Multiply and Accumulate
 - Programmable PID controller
- Cyclic Redundancy Check with Memory Scan (CRC/SCAN):
 - Software configurable
- · Serial Communications:
 - Enhanced USART (EUSART)
 - SPI, I²C, RS-232, RS-485, LIN compatible
 - Auto-Baud Detect, Auto-Wake-up on start

- Up to 17 I/O Pins and One Input-only Pin:
 - Individually programmable pull-ups
 - Slew rate control
 - Interrupt-on-change with edge-select
- Two High Current Drive pins
- Peripheral Pin Select (PPS):
 - Enables pin mapping of digital I/O

Intelligent Analog Peripherals

- 10-Bit Analog-to-Digital Converter (ADC):
 - Up to 12 external channels
 - Conversion available during Sleep
- Two Comparators (COMP):
 - Low-Power/High-Speed mode
 - Up to three external inverting inputs
 - Fixed Voltage Reference at non-inverting input(s)
 - Comparator outputs externally accessible
- 8-Bit Digital-to-Analog Converter (DAC):
 - 8-bit resolution, rail-to-rail
 - Positive Reference Selection
- Voltage Reference:
 - Fixed Voltage Reference (FVR): 1.024V, 2.048V and 4.096V output levels
- Zero-Cross Detect (ZCD):
 - Detect when AC signal on pin crosses ground
- Two High-Current Drive Pins:
 - 100mA @ 5V

Clocking Structure

- 16 MHz Internal Oscillator:
 - ±1% at calibration
 - Selectable frequency range from 32 MHz to 31 kHz
- 31 kHz Low-Power Internal Oscillator
- 4x Phase-Locked Loop (PLL):
 - For up to 32 MHz internal operation
- External Oscillator Block with:
 - Three external clock modes up to 32 MHz
 - One crystal resonator mode up to 32 MHz
- Fail-Safe Clock Monitor:
- Allows for safe shutdown if peripheral clock stops
- Two-Speed Oscillator Start-up
- Oscillator Start-up Timer (OST)

TABLE 1:

PIC12/16(L)F161X FAMILY TYPES

Device	Data Sheet Index	Program Memory Flash (W)	Program Memory Flash (kB)	Data SRAM (bytes)	High Endurance Flash (bytes)	I/O Pins	8-bit Timer with HLT	16-bit Timer	Angular Timer	Windowed Watchdog Timer	24-bit SMT	Comparators	10-bit ADC (ch)	Zero-Cross Detect	CCP/10-bit PWM	CWG	CLC	CRC with Memory Scan	Math Accelerator with PID	High-Current I/O 100mA	Sdd	EUSART
PIC12(L)F1612	(A)	2048	3.5	256	256	6	4	1	0	Y	1	1	4	1	2/0	1	0	Y	0	0	Ν	0
PIC16(L)F1613	(A)	2048	3.5	256	256	12	4	1	0	Y	2	2	8	1	2/0	1	0	Y	0	0	Ν	0
PIC16(L)F1614	(B)	4096	7	512	128	12	4	3	1	Y	2	2	8	1	2/2	1	2	Y	1	2	Υ	1
PIC16(L)F1615	(C)	8192	14	1024	1024	12	4	3	1	Υ	2	2	8	1	2/2	1	4	Υ	1	2	Y	1
PIC16(L)F1618	(B)	4096	7	512	128	18	4	3	1	Y	2	2	12	1	2/2	1	2	Y	1	2	Υ	1
PIC16(L)F1619	(C)	8192	14	1024	1024	18	4	3	1	Y	2	2	12	1	2/2	1	4	Y	1	2	Υ	1

Debugging Methods: (I) – Integrated on Chip; E – using Emulation Product Note 1:

Data Sheet Index:

В.

DS40001737 Α.

PIC12(L)F1612/16(L)F1613 Data Sheet, 8/14-Pin, 8-bit Flash Microcontrollers

DS40001769 PIC16(L)F1614/8 Data Sheet, 14/20-Pin, 8-bit Flash Microcontrollers

DS40001770 PIC16(L)F1615/9 Data Sheet, 14/20-Pin, 8-bit Flash Microcontrollers C.

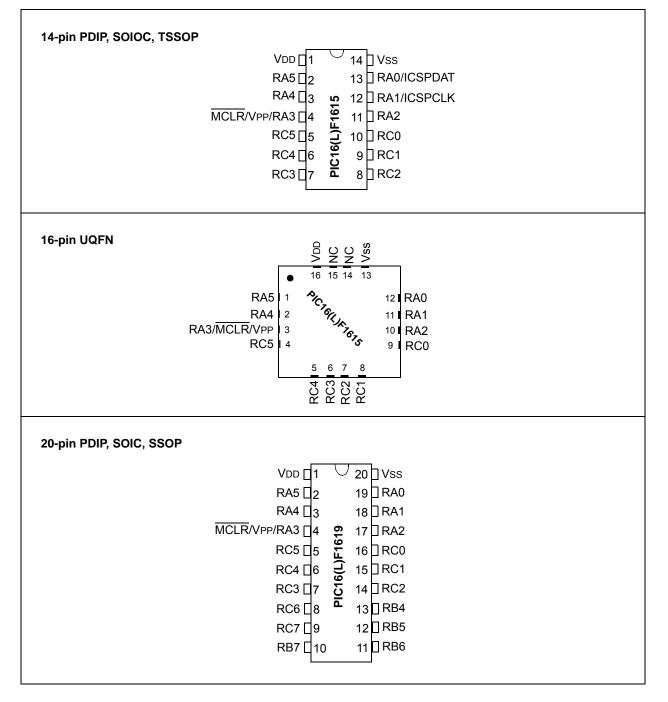
For other small form-factor package availability and marking information, please visit Note: http://www.microchip.com/packaging or contact your local sales office.

TABLE 2:PACKAGES

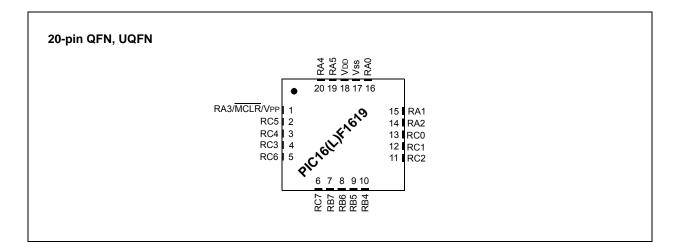
Packages	PDIP	SOIC	DFN	UDFN	TSSOP	QFN	UQFN	SSOP
PIC16(L)F1615	•	•			•	•		
PIC16(L)F1619	•	•				•	•	•

Note: Pin details are subject to change.

PIN DIAGRAMS



© 2014-2017 Microchip Technology Inc.



0
201
4-2
ŏ
7
Microchip
Technology
Inc.

TADLI	- • ·						(
I/O	14-Pin PDIP, SOIC, TSSOP	16-Pin UQFN	A/D	Reference	Comparator	Timers	CCP	CWG	ZCD	СГС	EUSART	SMT	Angular Timer	dSSM	MWG
RA0	13	12	AN0	DAC10UT1	C1IN+	_	_	—	-	—	_	-	-	_	—
RA1	12	11	AN1	VREF+	C1IN0- C2IN0-			—		_			l		—
RA2	11	10	AN2	—		T0CKI ⁽¹⁾		CWG1IN ⁽¹⁾	ZCD1IN	—					—
RA3	4	3	-	—	—	T6IN ⁽¹⁾	_	_	_	_	_	SMTWIN2 ⁽¹⁾	_	_	_
RA4	3	2	AN3	_	_	T1G ⁽¹⁾	_	_	_	—	_	SMTSIG1 ⁽¹⁾	—	_	_
RA5	2	1	-	-	_	T1CKI ⁽¹⁾ T2IN ⁽¹⁾	_	_	_	—	_	SMTWIN1 ⁽¹⁾	_	_	-
RC0	10	9	AN4	_	C2IN+	T5CKI ⁽¹⁾	_	_	_	_	_	_	_	SCK ^(1,3)	_
RC1	9	8	AN5	-	C1IN1- C2IN1-	T4IN ⁽¹⁾	_	_	_	_	_	SMTSIG2 ⁽¹⁾	_	SDI(1)	—
RC2	8	7	AN6	-	C1IN2- C2IN2-	—	-	-	-	—	_	-	-	-	-
RC3	7	6	AN7	-	C1IN3- C2IN3-	T5G ⁽¹⁾	CCP2 ⁽¹⁾	_	_	CLCIN0 ⁽¹⁾	_	_	ATCC1 ⁽¹⁾	<u>SS</u> (1)	-
RC4	6	5	—	—	—	T3G ⁽¹⁾	_	_	_	CLCIN1 ⁽¹⁾	CK ⁽¹⁾	_	ATCC2 ⁽¹⁾	_	_
RC5	5	4	_	—		T3CKI ⁽¹⁾	CCP1 ⁽¹⁾	_		—	RX ^(1,3)	_	ATIN ⁽¹⁾ ATCC3 ⁽¹⁾		—
Vdd	1	16	_	_			_	_	_	_	-	_	_	_	_
Vss	14	13	_	_				_		_		_		_	_
	_	—	—	_	C1OUT		CCP1	CWG1A	ZCD10UT	CLC10UT	DT ⁽³⁾	-		SDO	PWM3OL
OUT ⁽²⁾	-	—	_	—	C2OUT		CCP2	CWG1B	-	CLC2OUT	СК	_		SCK ⁽³⁾	PWM4OL
	_	_	_	_	-		-	CWG1C	-	_	ΤX	_	_	_	—
	—	—	—	—	—	-	_	CWG1D	-	—	_	—	—	_	—

PIN ALLOCATION TABLES

TABLE 3: 14/16-PIN ALLOCATION TABLE (PIC16(L)F1615)

1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. 2: 3:

These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

Note

0
201
4-20
7
-
Microch
ğ
ij
۲
Tech
Б
<u> </u>
g
\leq
nc.

TABLE 4:

SSOP

Q	20-Pin PDIP, SOIC, \$	20-Pin UQFN	A/D	Reference	Comparator	Timers	ССР	CWG	ZCD	СГС	EUSART	SMT	Angular Timer	dSSM	MWG
RA0	19	16	AN0	DAC10UT	C1IN+	—	_	—	_	—	_	—	_	—	_
RA1	18	15	AN1	VREF+	C1IN0- C2IN0-		-	-	-	—		_	-	-	
RA2	17	14	AN2	—	-	T0CKI ⁽¹⁾	-	CWG1IN ⁽¹⁾	ZCD1IN	—	—	—	-	-	—
RA3	4	1	-	_	_	T6IN ⁽¹⁾	_	—	_	_		SMTWIN2 ⁽¹⁾	_	_	
RA4	3	20	AN3	_	_	T1G ⁽¹⁾	_	_		_	_	SMTSIG1(1)	_	_	_
RA5	2	19	-	_	_	T1CKI ⁽¹⁾ T2IN ⁽¹⁾	_	—	_	CLCIN3 ⁽¹⁾		SMTWIN1 ⁽¹⁾	_	_	
RB4	13	10	AN10	_	_	_	_	_	_	—	_	_	_	SDI(1)	_
RB5	12	9	AN11	_				—	_	—	RX ^(1,3)	_		—	
RB6	11	8	Ι	_	-	-	-	_		—		_	-	SCK ^(1,3)	
RB7	10	7		-				_		—	CK ⁽¹⁾			—	
RC0	16	13	AN4	_	C2IN+	T5CKI ⁽¹⁾	_	—		_		_	_	_	
RC1	15	12	AN5		C1IN1- C2IN1-	T4IN ⁽¹⁾	-	-	-	CLCIN2 ⁽²⁾		SMTSIG2 ⁽¹⁾	-	-	
RC2	14	11	AN6	-	C1IN2- C2IN2-		_	_		_		_		-	
RC3	7	4	AN7	—	C1IN3- C2IN3-	T5G ⁽¹⁾	CCP2 ⁽¹⁾	-	—	CLCIN0 ⁽¹⁾	—	_	ATCC ⁽¹⁾	-	_
RC4	6	3	_	_	_	T3G ⁽¹⁾	_	_	_	CLCIN1 ⁽¹⁾	_	_	_	_	_
RC5	5	2		_	_	T3CKI ⁽¹⁾	CCP1 ⁽¹⁾	_		—		_	ATIN ⁽¹⁾	_	
RC6	8	5	AN8					_		—				<u>SS</u> (1)	
RC7	9	6	AN9	_	_	_	_	—	_	—	_	_	_	—	_
VDD	1	18	—	_	_	_	_	—	_	—	_	_	_	—	_
Vss	20	17	_	_	_	_	_	_	_		_	_	_	—	_
Note	1: C	Default	periphe	ral input. Inpu	it can be m	oved to any	other pin v	vith the PPS ir	put selection	n registers.					

1: 2: 3:

20-PIN ALLOCATION TABLE (PIC16(L)F1619)

Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

DS40001770D-page 7

TABLE 4: 20-PIN ALLOCATION TABLE (PIC16(L)F1619)

IADEE															
01	20-Pin PDIP, SOIC, SSOP	20-Pin UQFN	A/D	Reference	Comparator	Timers	ССР	CWG	ZCD	CLC	EUSART	SMT	Angular Timer	dssm	PWM
	C10UT - CCP1 CWG1A ZCD10UT CLC10UT DT ⁽³⁾ SDO PWM30U														
OUT ⁽²⁾															
001-	_		_		_	_	_	CWG1C	_	CLC3OUT	TX	_	_	_	_
	CWG1D CLC40UT														
Note	ote 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.														

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.

3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE OF CONTENTS

Device Overview	11
Enhanced Mid-Range CPU	
Memory Organization	
Device Configuration	
Oscillator Module	
Resets	
Interrupts	
Power-Down Mode (Sleep)	
Windowed Watchdog Timer (WDT)	
Flash Program Memory Control	
Cyclic Redundancy Check (CRC) Module	
I/O Ports	
Peripheral Pin Select (PPS) Module	170
Interrupt-On-Change	
Fixed Voltage Reference (FVR)	
Temperature Indicator Module	
Analog-to-Digital Converter (ADC) Module	
8-bit Digital-to-Analog Converter (DAC1) Module	
Comparator Module	
Zero-Cross Detection (ZCD) Module	
Timer0 Module	
Timer1/3/5 Module with Gate Control	
Timer2/4/6 Module	
Master Synchronous Serial Port (MSSP) Module	
Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	
Capture/Compare/PWM Modules	
Pulse-Width Modulation (PWM) Module	
Complementary Waveform Generator (CWG) Module	
Signal Measurement Timer (SMT)	
Configurable Logic Cell (CLC)	
Angular Timer (AT) Module	
Math Accelerator with Proportional-Integral-Derivative (PID) Module	
In-Circuit Serial Programming™ (ICSP™)	
Instruction Set Summary	
Electrical Specifications	
DC and AC Characteristics Graphs and Charts	
Development Support	
Packaging Information	
Data Sheet Revision History	

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** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. 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., DS30000A is version A of document DS30000).

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 PIC16(L)F1615/9 are described within this data sheet. The block diagram of these devices are shown in Figure 1-1, the available peripherals are shown in Table 1-1, and the pin out descriptions are shown in Tables 1-2 and 1-3.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

PeripheralSP SP	JOININIAN	-		,
Complementary Wave Generator (CWG) • • Cyclic Redundancy Check (CRC) • • Digital-to-Analog Converter (DAC) • • Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART) • • Fixed Voltage Reference (FVR) • • • Temperature Indicator • • • • Windowed Watchdog Timer (WDT) • • • • • Zero-Cross Detection (ZCD) •	Peripheral		PIC16(L)F1615	PIC16(L)F1619
Cyclic Redundancy Check (CRC) • • Digital-to-Analog Converter (DAC) • • Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART) • • Fixed Voltage Reference (FVR) • • • Temperature Indicator • • • Windowed Watchdog Timer (WDT) • • • Zero-Cross Detection (ZCD) • • • Capture/Compare/PWM (CCP) Modules • • • Comparators CCP1 • • • Comparators C1 • • • • Configurable Logic Cell (CLC) • • • • • • Master Synchronous Serial Ports • <td>Analog-to-Digital Converter (A</td> <td>ADC)</td> <td>•</td> <td>•</td>	Analog-to-Digital Converter (A	ADC)	•	•
Digital-to-Analog Converter (DAC) • • Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART) • • Fixed Voltage Reference (FVR) • • • Temperature Indicator • • • Windowed Watchdog Timer (WDT) • • • Zero-Cross Detection (ZCD) • • • Capture/Compare/PWM (CCP) Modules • • • Comparators CCP1 • • Configurable Logic Cell (CLC) • • • CLC1 • • • • CLC2 • • • • • Configurable Logic Cell (CLC) •	Complementary Wave Generation	ator (CWG)	•	•
Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART) • • Fixed Voltage Reference (FVR) • • Temperature Indicator • • Windowed Watchdog Timer (WDT) • • Zero-Cross Detection (ZCD) • • Capture/Compare/PWM (CCP) Modules • • CCP2 • • Comparators C1 • Configurable Logic Cell (CLC) • • CLC1 • • CLC2 • • CLC3 • • CLC3 • • CLC3 • • CLC3 • • CLC4 • • Master Synchronous Serial Ports • • Pulse-Width Modulator (PWM) • • Signal Measurement Timer (SMT) • • Signal Measurement Timer (SMT) • • Timers • • • Timer3 <	Cyclic Redundancy Check (C	RC)	٠	•
Synchronous/Asynchronous Receiver/Transmitter (EUSART)Image: Comparison of the synchronous of the	Digital-to-Analog Converter (I	•	•	
Temperature Indicator • • Windowed Watchdog Timer (WDT) • • Zero-Cross Detection (ZCD) • • Capture/Compare/PWM (CCP) Modules • • CCP1 • • CCP2 • • Comparators C1 • Configurable Logic Cell (CLC) • • ClC2 • • ClC2 • • Configurable Logic Cell (CLC) • • CLC3 • • CLC3 • • CLC3 • • CLC4 • • Master Synchronous Serial Ports • • Pulse-Width Modulator (PWM) • • Signal Measurement Timer (SMT) • • Signal Measurement Timer (SMT) • • Timers • • • Timer3 • • •	Synchronous/Asynchronous	•	•	
Windowed Watchdog Timer (WDT) • • Zero-Cross Detection (ZCD) • • Capture/Compare/PWM (CCP) Modules • • CCP1 • • CCP2 • • Comparators CCP1 • Comparators • • Configurable Logic Cell (CLC) • • CLC1 • • CLC2 • • CLC2 • • CLC2 • • CLC2 • • CLC3 • • CLC3 • • CLC4 • • Master Synchronous Serial Ports • • Pulse-Width Modulator (PWM) • • Signal Measurement Timer (SMT) • • Signal Measurement Timer (SMT1 • • Timers • • • Timer1 • • •	Fixed Voltage Reference (FV	•	•	
Zero-Cross Detection (ZCD) Image: Compare/PWM (CCP) Modules CCP1 Image: CCP2 Ima	•		•	•
Capture/Compare/PWM (CCP) Modules CCP1 • CCP2 • C1 • C2 • CLC1 • CLC2 • CLC3 • CLC3 • CLC3 • CLC4 • Master Synchronous Serial Ports • Pulse-Width Modulator (PWM) • PWM3 • Signal Measurement Timer (SMT) • Signal Measurement Timer (SMT1 • SMT1 • • Timer3 • • Timer4 • •	Windowed Watchdog Timer (WDT)	•	•
CCP1 • CCP2 • COmparators • C1 • C2 • Configurable Logic Cell (CLC) • CLC1 • CLC2 • CLC3 • CLC3 • CLC3 • CLC4 • Master Synchronous Serial Ports • Master Synchronous Serial Ports • Pulse-Width Modulator (PWM) • Signal Measurement Timer (SMT) • Signal Measurement Timer (SMT) • Timers • Timer3 • Timer3 • Timer3 •	Zero-Cross Detection (ZCD)		•	•
CCP2•ComparatorsC1•C1•C2•C2•Configurable Logic Cell (CLC)CLC1•CLC2•CLC3•CLC3•CLC4•CLC4•CLC4•Master Synchronous Serial PortsMaster Synchronous Serial PortsPulse-Width Modulator (PWM)Pulse-Width Modulator (PWM4PWM3•Signal Measurement Timer (SMT)SMT1•SMT1•SMT2•TimersTimer1•Timer3•Timer3•Timer3•Timer4•Timer5•	Capture/Compare/PWM (CCF	P) Modules		
Comparators C1 • C1 • • • C2 • • • Configurable Logic Cell (CLC) • • • CLC1 • • • • CLC2 • • • • • CLC3 • <		CCP1	•	•
C1 • C2 • Configurable Logic Cell (CLC) • CLC1 • CLC2 • CLC3 • CLC3 • CLC4 • Master Synchronous Serial Ports MSSP1 Master Synchronous Serial Ports • Pulse-Width Modulator (PWM) • Signal Measurement Timer (SMT) • Signal Measurement Timer (SMT) • SMT1 • SMT2 • Timers • Timer1 • Timer3 • Timer3 •		CCP2	•	•
C2•Configurable Logic Cell (CLC)CLC1•CLC2•CLC3•CLC3•CLC4•Master Synchronous Serial PortsMaster Synchronous Serial PortsPulse-Width Modulator (PWW)PWM3•PWM4•Signal Measurement Timer (SMT)SMT1•SMT1•Timers•Timer1•Timer2•Timer3•Timer3•Timer4•Timer5•	Comparators			
Configurable Logic Cell (CLC) CLC1 • CLC2 • CLC3 • CLC4 • Master Synchronous Serial Ports • Master Synchronous Serial Ports • Pulse-Width Modulator (PWM) • PUBSe-Width Modulator (PWM4 • Signal Measurement Timer (SMT) • Signal Measurement Timer (SMT1 • SMT1 • SMT2 • Timers • Timer1 • Timer3 • Timer4 •		C1	•	•
CLC1 • CLC2 • CLC3 • CLC3 • CLC4 • Master Synchronous Serial Ports MSSP1 Master Synchronous Serial Ports MSSP1 Pulse-Width Modulator (PWM) • PWM3 • PWM4 • Signal Measurement Timer (SMT) • SMT1 • SMT2 • Timers • Timer1 • Timer2 • Timer3 • Timer5 •		C2	•	•
CLC2•CLC3•CLC3•CLC4•Master Synchronous Serial PortsMSSP1•Pulse-Width Modulator (PWW)PWM3•PWM4•PWM4•Signal Measurement Timer (SMT)SMT1•SMT2•Timers•Timer0•Timer1•Timer3•Timer3•Timer4•Timer5•	Configurable Logic Cell (CLC)		
CLC3•CLC4••Master Synchronous Serial PortsMSSP1•Pulse-Width Modulator (PWM)••PWM3••PWM4••Signal Measurement Timer (SMT)••Signal Measurement Timer (SMT)••TimersSMT1••Timers•••Timer1•••Timer2•••Timer3•••Timer4•••		CLC1	•	•
CLC4 • Master Synchronous Serial Ports MSSP1 • Pulse-Width Modulator (PWW) • • PWM3 • • PWM4 • • Signal Measurement Timer (SMT) • • Signal Measurement Timer (SMT) • • Signal Measurement Timer (SMT) • • Timers SMT1 • • Timers • • • Timer1 • • • Timer2 • • • Timer3 • • •		CLC2	•	•
Master Synchronous Serial Ports MSSP1 • Pulse-Width Modulator (PWM) • PWM3 • • PWM4 • • Signal Measurement Timer (SMT) • • SMT1 • • SMT2 • • Timers • • Timer1 • • Timer2 • • Timer3 • • Timer4 • •		CLC3	•	•
MSSP1 • • Pulse-Width Modulator (PWM) PWM3 • • PWM4 • • • Signal Measurement Timer (SMT) SMT1 • • Signal Measurement Timer (SMT) • • • Timers SMT1 • • Timers • • • Timer1 • • • Timer2 • • • Timer3 • • • Timer5 • • •		CLC4	•	•
Pulse-Width Modulator (PWM) PWM3 • PWM3 • • PWM4 • • Signal Measurement Timer (SMT) • • SMT1 • • SMT2 • • Timers • • Timer1 • • Timer2 • • Timer3 • • Timer4 • •	Master Synchronous Serial P	orts		
PWM3 • • PWM4 • • Signal Measurement Timer (SMT) • • SMT1 • • SMT2 • • Timers • • Timer0 • • Timer1 • • Timer3 • • Timer3 • • Timer4 • •		MSSP1	٠	•
PWM4 • • Signal Measurement Timer (SMT) SMT1 • • SMT1 • • • • SMT2 •	Pulse-Width Modulator (PWM	1)		
Signal Measurement Timer (SMT) SMT1 •		PWM3	٠	•
SMT1 • • SMT2 • • Timers Timer0 • • Timer1 • • • Timer2 • • • Timer3 • • • Timer4 • • •		PWM4	٠	•
SMT2 • • Timers Timer0 • • Timer1 • • • Timer2 • • • Timer3 • • • Timer4 • • •	Signal Measurement Timer (S	SMT)		
Timers Timer0 • Timer1 • • Timer2 • • Timer3 • • Timer4 • •		SMT1	•	•
Timer0 • Timer1 • Timer2 • Timer3 • Timer4 • Timer5 •		SMT2	•	•
Timer1 • • Timer2 • • Timer3 • • Timer4 • • Timer5 • •	Timers			
Timer2•Timer3•Timer4•Timer5•		Timer0	•	•
Timer3•Timer4•Timer5•		Timer1	٠	•
Timer4•Timer5•		Timer2	•	•
Timer5 • •		Timer3	•	•
		Timer4	•	•
Timer6 • •		Timer5	٠	٠
		Timer6	•	•

1.1 Register and Bit Naming Conventions

1.1.1 REGISTER NAMES

When there are multiple instances of the same peripheral in a device, the peripheral control registers will be depicted as the concatenation of a peripheral identifier, peripheral instance, and control identifier. The control registers section will show just one instance of all the register names with an 'x' in the place of the peripheral instance number. This naming convention may also be applied to peripherals when there is only one instance of that peripheral in the device to maintain compatibility with other devices in the family that contain more than one.

1.1.2 BIT NAMES

There are two variants for bit names:

- Short name: Bit function abbreviation
- Long name: Peripheral abbreviation + short name

1.1.2.1 Short Bit Names

Short bit names are an abbreviation for the bit function. For example, some peripherals are enabled with the EN bit. The bit names shown in the registers are the short name variant.

Short bit names are useful when accessing bits in C programs. The general format for accessing bits by the short name is *RegisterName*bits.*ShortName*. For example, the enable bit, EN, in the COG1CON0 register can be set in C programs with the instruction COG1CON0bits.EN = 1.

Short names are generally not useful in assembly programs because the same name may be used by different peripherals in different bit positions. When this occurs, during the include file generation, all instances of that short bit name are appended with an underscore plus the name of the register in which the bit resides to avoid naming contentions.

1.1.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral, thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.

Long bit names are useful in both C and assembly programs. For example, in C the COG1CON0 enable bit can be set with the G1EN = 1 instruction. In assembly, this bit can be set with the BSF COG1CON0, G1EN instruction.

1.1.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the mode control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

COG1CON0bits.MD = 0x5;

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name MD2 and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode:

Example 1:

MOVLW ~(1<<G1MD1) ANDWF COG1CON0,F MOVLW 1<<G1MD2 | 1<<G1MD0 IORWF COG1CON0,F

Example 2:

BSF	COG1CON0,G1MD2
BCF	COG1CON0,G1MD1
BSF	COG1CON0,G1MD0

1.1.3 REGISTER AND BIT NAMING EXCEPTIONS

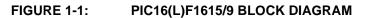
1.1.3.1 Status, Interrupt, and Mirror Bits

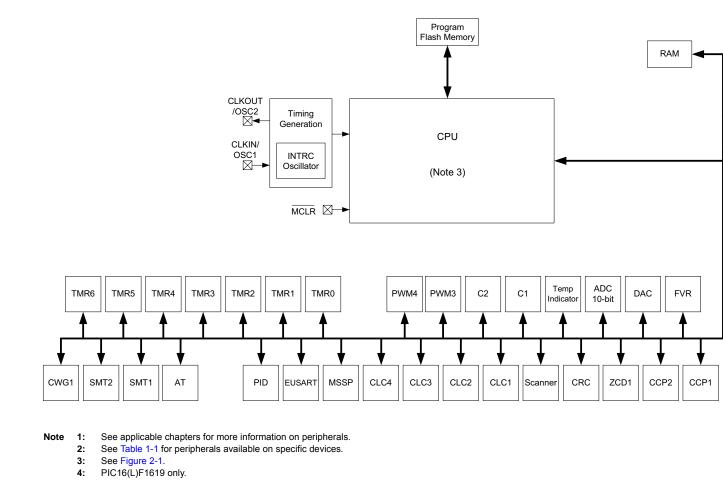
Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

1.1.3.2 Legacy Peripherals

There are some peripherals that do not strictly adhere to these naming conventions. Peripherals that have existed for many years and are present in almost every device are the exceptions. These exceptions were necessary to limit the adverse impact of the new conventions on legacy code. Peripherals that do adhere to the new convention will include a table in the registers section indicating the long name prefix for each peripheral instance. Peripherals that fall into the exception category will not have this table. These peripherals include, but are not limited to, the following:

- EUSART
- MSSP





4:

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT1/	RA0	TTL/ST	CMOS/OD	General purpose I/O.
ICSPDAT	AN0	AN		ADC Channel input.
	C1IN+	AN	_	Comparator positive input.
	DAC1OUT1	_	AN	Digital-to-Analog Converter output.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/AN1/VREF+/C1IN0-/C2IN0-/	RA1	TTL/ST	CMOS/OD	General purpose I/O.
ICSPCLK	AN1	AN	_	ADC Channel input.
	VREF+	AN	_	Voltage Reference input.
	C1IN0-	AN	_	Comparator negative input.
	C2IN0-	AN	CMOS/OD	Comparator negative input.
	ICSPCLK	ST	_	ICSP Programming Clock.
RA2/AN2/T0CKI ⁽¹⁾ /CWG1IN ⁽¹⁾	RA2	TTL/ST	CMOS/OD	General purpose I/O.
ZCD1IN/INT	AN2	AN	_	ADC Channel input.
	TOCKI	TTL/ST	_	Timer0 clock input.
	CWG1IN	TTL/ST	_	CWG complementary input.
	ZCD1IN	AN	_	Zero-Cross Detect input.
	INT	TTL/ST	_	External interrupt.
RA3/VPP/T6IN ⁽¹⁾ /SMTWIN2 ⁽¹⁾ /	RA3	TTL/ST	_	General purpose input with IOC and WPU.
MCLR	VPP	HV	_	Programming voltage.
	T6IN	TTL/ST	_	Timer6 input.
	SMTWIN2	TTL/ST	_	SMT2 window input.
	MCLR	TTL/ST	_	Master Clear with internal pull-up.
RA4/AN3/T1G ⁽¹⁾ /SMTSIG1 ⁽¹⁾ /	RA4	TTL/ST	CMOS/OD	General purpose I/O.
CLKOUT	AN3	AN	_	ADC Channel input.
	T1G	TTL/ST	_	Timer1 Gate input.
	SMTSIG1	TTL/ST	_	SMT1 signal input.
	CLKOUT	_	CMOS	Fosc/4 output.
RA5/CLKIN/T1CKI ⁽¹⁾ /T2IN ⁽¹⁾ /	RA5	TTL/ST	CMOS/OD	General purpose I/O.
SMTWIN1 ⁽¹⁾	CLKIN	CMOS	_	External clock input (EC mode).
	T1CKI	TTL/ST	_	Timer1 clock input.
	T2IN	TTL/ST	_	Timer2 input.
	SMTWIN1	TTL/ST	_	SMT1 window input.
RC0/AN4/C2IN+/T5CKI(1)/	RC0	TTL/ST	CMOS/OD	General purpose I/O.
SCK ⁽¹⁾	AN4	AN	_	ADC Channel input.
	C2IN+	AN	_	Comparator positive input.
	T5CKI	TTL/ST	—	Timer5 clock input.
	SCK	ST	CMOS	SPI clock.

TABLE 1-2: PIC16(L)F1615 PINOUT DESCRIPTION

Legend:AN= Analog input or outputCMOS= CMOS compatible input or outputOD=Open-DrainTTL= TTL compatible inputST= Schmitt Trigger input with CMOS levels I^2C =Schmitt Trigger input with I^2C HV= High VoltageXTAL= CrystalLevels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-1.

Name	Function	Input Type	Output Type	Description
RC1/AN5/C1IN1-/C2IN1-/	RC1	TTL/ST	CMOS/OD	General purpose I/O.
T4IN ⁽¹⁾ /SMTSIG2 ⁽¹⁾ /SDI ⁽¹⁾	AN5	AN	_	ADC Channel input.
	C1IN1-	AN	_	Comparator negative input.
	C2IN1-	AN	_	Comparator negative input.
	T4IN	TTL/ST	_	Timer4 input.
	SMTSIG2	TTL/ST	_	SMT2 signal input.
	SDI	CMOS	_	SPI data input.
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	General purpose I/O.
	AN6	AN	_	ADC Channel input.
	C1IN2-	AN	_	Comparator negative input.
	C2IN2-	AN	_	Comparator negative input.
RC3/AN7/C1IN3-/C2IN3-/T5G ⁽¹⁾ /	RC3	TTL/ST	_	General purpose input with IOC and WPU.
CCP2 ⁽¹⁾ /CLCIN0 ⁽¹⁾ /ATCC ⁽¹⁾ /SS	AN7	AN	_	ADC Channel input.
	C1IN3-	AN	_	Comparator negative input.
	C2IN3-	AN	_	Comparator negative input.
	T5G	ST	_	Timer5 Gate input.
	CCP2	TTL/ST	CMOS/OD	Capture/Compare/PWM2.
	CLCIN0	ST	_	Configurable Logic Cell source input.
	ATCC	ST		Angular Timer Capture/Compare input.
	SS	ST	_	Slave Select input.
RC4/T3G ⁽¹⁾ /CLCIN1 ⁽¹⁾ /CK ⁽¹⁾ /	RC4	TTL/ST	CMOS/OD	General purpose I/O.
HIC4	T3G	TTL/ST	_	Timer3 Gate input.
	CLCIN1	ST	_	Configurable Logic Cell source input.
	СК	ST	CMOS	EUSART synchronous clock.
	HIC4	TTL	CMOS	High current I/O.
RC5/T3CKI ⁽¹⁾ /CCP1 ⁽¹⁾ /RX ⁽¹⁾ /	RC5	TTL/ST	CMOS/OD	General purpose I/O.
ATIN ⁽¹⁾ /HIC5	T3CKI	TTL/ST		Timer3 clock input.
	CCP1	TTL/ST	CMOS/OD	Capture/Compare/PWM1.
	RX	ST	—	EUSART asynchronous input.
	ATIN	TTL/ST	—	Angular Timer clock input.
	HIC5	TTL	CMOS	High current I/O.

TABLE 1-2. FICTO(L)FTOTO FINOUT DESCRIPTION (CONTINUED	TABLE 1-2:	PIC16(L)F1615 PINOUT DESCRIPTION (CONTINUED)
--	------------	--

 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

 HV
 = High Voltage
 XTAL
 =
 Crystal
 Ievels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-1.

Name	Function	Input Type	Output Type	Description
OUT ⁽²⁾	C10UT	_	CMOS	Comparator output.
	C2OUT	—	CMOS	Comparator output.
	CCP1	—	CMOS	Capture/Compare/PWM1 output.
	CCP2	—	CMOS	Capture/Compare/PWM2 output.
	PWM3OUT	_	CMOS	PWM3 output.
	PWM4OUT	—	CMOS	PWM4 output.
	CWG1A	—	CMOS	Complementary Output Generator Output A.
	CWG1B	_	CMOS	Complementary Output Generator Output B.
	CWG1C	_	CMOS	Complementary Output Generator Output C.
	CWG1D	—	CMOS	Complementary Output Generator Output D.
	SDO	_	CMOS	SPI data output.
	SCK	—	CMOS	SPI clock output.
	TX/CK	_	CMOS	EUSART asynchronous TX data/synchronous clock output.
	DT	—	CMOS	EUSART synchronous data output.E
	CLC1OUT	—	CMOS	Configurable Logic Cell 1 source output.
	CLC2OUT	_	CMOS	Configurable Logic Cell 2 source output.
	ZCD1OUT	_	CMOS	Zero-Cross Detect output.
Vdd	VDD	Power	_	Positive supply.
Vss	Vss	Power	_	Ground reference.

TABLE 1-2: PIC16(L)F1615 PINOUT DESCRIPTION (CONTINUED)

Schmitt Trigger input with I²C TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C =

HV = High Voltage XTAL = Crystal

levels Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-1.

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT/	RA0	TTL/ST	CMOS/OD	General purpose I/O.
ICSPDAT	AN0	AN		ADC Channel input.
	C1IN+	AN	_	Comparator positive input.
	DAC1OUT	—	AN	Digital-to-Analog Converter output.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/AN1/VREF+/C1IN0-/C2IN0-/	RA1	TTL/ST	CMOS/OD	General purpose I/O.
ICSPCLK	AN1	AN		ADC Channel input.
	VREF+	AN	_	Voltage Reference input.
	C1IN0-	AN	-	Comparator negative input.
	C2IN0-	AN	CMOS/OD	Comparator negative input.
	ICSPCLK	ST		ICSP Programming Clock.
RA2/AN2/T0CKI ⁽¹⁾ /CWG1IN ⁽¹⁾ /	RA2	TTL/ST	CMOS/OD	General purpose I/O.
ZCD1IN/INT	AN2	AN	_	ADC Channel input.
	TOCKI	TTL/ST	_	Timer0 clock input.
	CWG1IN	TTL/ST	_	CWG complementary input.
	ZCD1IN	AN	_	Zero-Cross Detect input.
	INT	TTL/ST		External interrupt.
RA3/VPP/T6IN ⁽¹⁾ /SMTWIN2 ⁽¹⁾ /	RA3	TTL/ST		General purpose input with IOC and WPU.
MCLR	Vpp	HV	_	Programming voltage.
	T6IN	TTL/ST		Timer6 input.
	SMTWIN2	TTL/ST	_	SMT2 window input.
	MCLR	TTL/ST		Master Clear with internal pull-up.
RA4/AN3/T1G ⁽¹⁾ /SMTSIG1 ⁽¹⁾ /	RA4	TTL/ST	CMOS/OD	General purpose I/O.
CLKOUT	AN3	AN		ADC Channel input.
	T1G	TTL/ST	_	Timer1 Gate input.
	SMTSIG1	TTL/ST		SMT1 signal input.
	CLKOUT	_	CMOS	Fosc/4 output.
RA5/CLKIN/T1CKI ⁽¹⁾ /T2IN ⁽¹⁾ /	RA5	TTL/ST	CMOS/OD	General purpose I/O.
CLCIN3 ⁽¹⁾ /SMTWIN1	CLKIN	CMOS		External clock input (EC mode).
	T1CKI	TTL/ST	_	Timer1 clock input.
	T2IN	TTL/ST	_	Timer2 input.
	CLCIN3	ST		Configurable Logic Cell source input.
	SMTWIN1	TTL/ST	_	SMT1 window input.
RB4/AN10/SDI ⁽¹⁾	RB4	TTL/ST	CMOS/OD	General purpose I/O.
	AN10	AN	_	ADC Channel input.
	SDI	CMOS	_	SPI data input.
RB5/AN11/RX ^(1, 3)	RB5	TTL/ST	CMOS/OD	General purpose I/O.
	AN11	AN	_	ADC Channel input.
	RX	ST		EUSART asynchronous input.
Legend: AN = Analog input or o			l manatible innu	

TABLE 1-3: PIC16(L)F1619 PINOUT DESCRIPTION

Legend:AN= Analog input or outputCMOS= CMOS compatible input or outputOD=Open-DrainTTL= TTL compatible inputST=Schmitt Trigger input with CMOS levels l^2C =Schmitt Trigger

HV = High Voltage XTAL = Crystal

Schmitt Trigger input with I²C levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-1.

3: These I²C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 1-3:	PIC16(L)F1619 PINOUT DESCRIPTION ((CONTINUED)	1
IADEE I-J.		CONTINUED	,

Name	Function	Input Type	Output Type	Description	
RB6/SCK ^(1, 3)	RB6	TTL/ST	CMOS/OD	General purpose I/O.	
	SCK	ST	CMOS	SPI clock.	
RB7/CK ⁽¹⁾	RB7	TTL/ST	CMOS/OD	General purpose I/O.	
	СК	ST	CMOS	EUSART synchronous clock.	
RC0/AN4/C2IN+/T5CKI ⁽¹⁾	RC0	TTL/ST	CMOS/OD	General purpose I/O.	
	AN4	AN	_	ADC Channel input.	
	C2IN+	AN	_	Comparator positive input.	
	T5CKI	TTL/ST	_	Timer5 clock input.	
RC1/AN5/C1IN1-/C2IN1-/	RC1	TTL/ST	CMOS/OD	General purpose I/O.	
T4IN ⁽¹⁾ /CLCIN2 ⁽²⁾ /SMTSIG2 ⁽¹⁾	AN5	AN	_	ADC Channel input.	
	C1IN1-	AN	_	Comparator negative input.	
	C2IN1-	AN	_	Comparator negative input.	
	T4IN	TTL/ST	_	Timer4 input.	
	CLCIN2	ST	_	Configurable Logic Cell source input.	
	SMTSIG2	TTL/ST	_	SMT2 signal input.	
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	General purpose I/O.	
	AN6	AN	_	ADC Channel input.	
	C1IN2-	AN	_	Comparator negative input.	
	C2IN2-	AN	_	Comparator negative input.	
RC3/AN7/C1IN3-/C2IN3-/T5G ⁽¹⁾ / CCP2 ⁽¹⁾ /CLCIN0 ⁽¹⁾ /ATCC ⁽¹⁾	RC3	TTL/ST	_	General purpose input with IOC and WPU.	
	AN7	AN	—	ADC Channel input.	
	C1IN3-	AN	—	Comparator negative input.	
	C2IN3-	AN	—	Comparator negative input.	
	T5G	ST	—	Timer5 Gate input.	
	CCP2	ST	CMOS	Capture/Compare/PWM2.	
	CLCIN0	ST	—	Configurable Logic Cell source input.	
	ATCC	ST	—	Angular Timer Capture/Compare input.	
RC4/T3G ⁽¹⁾ /CLCIN1 ⁽¹⁾ /HIC4	RC4	TTL/ST	CMOS/OD	General purpose I/O.	
	T3G	ST	_	Timer3 Gate input.	
	CLCIN1	ST	—	Configurable Logic Cell source input.	
	HIC4	TTL	CMOS	High current I/O.	
RC5/T3CKI ⁽¹⁾ /CCP1 ⁽¹⁾ /ATIN ⁽¹⁾ /	RC5	TTL/ST	CMOS/OD	General purpose I/O.	
HIC5	T3CKI	TTL/ST	—	Timer3 clock input.	
	CCP1	TTL/ST	CMOS/OD	Capture/Compare/PWM1.	
	ATIN	TTL/ST	_	Angular Timer clock input.	
	HIC5	TTL	CMOS	High current I/O.	
RC6/AN8/SS ⁽¹⁾	RC6	TTL/ST	CMOS/OD	General purpose I/O.	
	AN8	AN	—	ADC Channel input.	
	SS	ST	_	SPI slave select.	
RC7/AN9	RC7	TTL/ST	CMOS/OD	General purpose I/O.	
	AN9	AN	_	ADC Channel input.	

HV = High Voltage XTAL = Crystal

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-1.

3: These I²C functions are bidirectional. The output pin selections must be the same as the input pin selections.

levels

Name	Function	Input Type	Output Type	Description
OUT ⁽²⁾	C1OUT	—	CMOS	Comparator output.
	C2OUT	_	CMOS	Comparator output.
	CCP1	_	CMOS	Capture/Compare/PWM1 output.
	CCP2	—	CMOS	Capture/Compare/PWM2 output.
	PWM3OUT	—	CMOS	PWM3 output.
	PWM4OUT	—	CMOS	PWM4 output.
	CWG1A	—	CMOS	Complementary Output Generator Output A.
	CWG1B	_	CMOS	Complementary Output Generator Output B.
	CWG1C	_	CMOS	Complementary Output Generator Output C.
	CWG1D	—	CMOS	Complementary Output Generator Output D.
	SDO	_	CMOS	SPI data output.
	SCK	_	CMOS	SPI clock output.
	TX/CK	_	CMOS	EUSART asynchronous TX data/synchronous clock output.
	DT	—	CMOS	EUSART synchronous data output.
	CLC1OUT	—	CMOS	Configurable Logic Cell 1 source output.
	CLC2OUT	_	CMOS	Configurable Logic Cell 2 source output.
	CLC3OUT	—	CMOS	Configurable Logic Cell 3 source output.
	CLC4OUT	—	CMOS	Configurable Logic Cell 4 source output.
	ZCD10UT	—	CMOS	Zero-Cross Detect output.
Vdd	Vdd	Power		Positive supply.
Vss	Vss	Power	_	Ground reference.

TABLE 1-3: PIC16(L)F1619 PINOUT DESCRIPTION (CONTINUED)

Drain og input TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C Schmitt Trigger input with I²C HV = High Voltage levels

XTAL = Crystal

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-1.

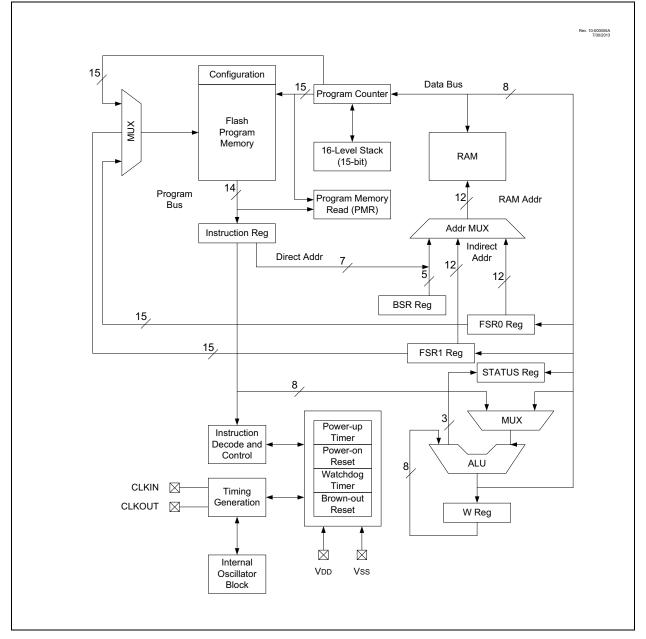
3: These I^2C functions are bidirectional. The output pin selections must be the same as the input pin selections.

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 a hardware 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 **Section 3.5** "**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 3.6 "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 34.0** "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

3.1 **Program Memory Organization**

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented. 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 Figure 3-1).

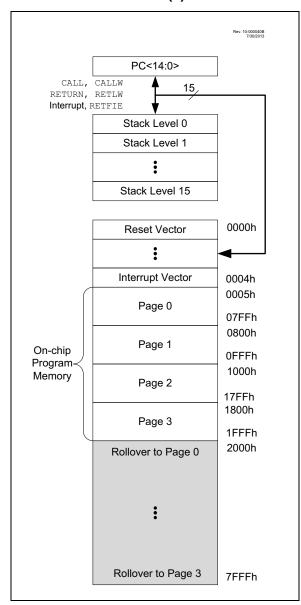
3.2 High-Endurance Flash

This device has a 128-byte section of high-endurance Program Flash Memory (PFM) in lieu of data EEPROM. This area is especially well suited for nonvolatile data storage that is expected to be updated frequently over the life of the end product. See Section 10.2 "Flash Program Memory Overview" for more information on writing data to PFM. See Section 3.2.1.2 "Indirect Read with FSR" for more information about using the FSR registers to read byte data stored in PFM.

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range ⁽¹⁾
PIC16(L)F1615/9	8,192	1FFFh	1F80h-1FFFh

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

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1615/9



3.2.1 READING PROGRAM MEMORY AS DATA

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

3.2.1.1 RETLW Instruction

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 CO	DDE
MOVLW D.	ATA_INDEX
call constant	ts
; THE CONSTR	ANT 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.

3.2.1.2 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 operator will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2:	ACCESSING PROGRAM
	MEMORY VIA FSR

DW DATA0 ;First constant DW DATA1 ;Second constant DW DATA2 DW DATA3 my_function ; LOTS OF CODE MOVLW DATA_INDEX ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS,C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	constants		
DW DATA2 DW DATA3 my_function ; LOTS OF CODE MOVLW DATA_INDEX ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	DW DATA	.0	;First constant
DW DATA3 my_function ; LOTS OF CODE MOVLW DATA_INDEX ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	DW DATA	1	;Second constant
<pre>my_function ;LOTS OF CODE MOVLW DATA_INDEX ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets</pre>	DW DATA	2	
<pre>; LOTS OF CODE MOVLW DATA_INDEX ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]</pre>	DW DATA	.3	
MOVLW DATA_INDEX ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	my_function	on	
ADDLW LOW constants MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	;… LOTS	OF CODE	
MOVWF FSR1L MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	MOVLW	DATA_INDEX	
MOVLW HIGH constants;MSb sets automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	ADDLW	LOW constants	
automatically MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	MOVWF	FSR1L	
MOVWF FSR1H BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]	MOVLW	HIGH constants	s;MSb sets
BTFSC STATUS, C ;carry from ADDLW? INCF FSR1h, f ;yes MOVIW 0[FSR1]			automatically
INCF FSR1h, f ;yes MOVIW 0[FSR1]	MOVWF	FSR1H	
MOVIW 0[FSR1]	BTFSC	STATUS, C	<pre>;carry from ADDLW?</pre>
	INCF	FSR1h, f	;yes
	MOVIW	0[FSR1]	
;THE PROGRAM MEMORY IS IN W	; THE PROG	RAM MEMORY IS	IN W

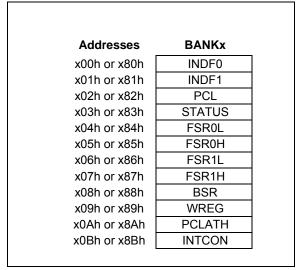
3.3 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-2):

- 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

TABLE 3-1: CORE REGISTERS



file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.6 "Indirect Addressing**" for more information.

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

3.3.1 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/x80h through x0Bh/x8Bh). These registers are listed below in Table 3-1. For detailed

3.3.1.1 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.

REGISTER 3-1: STATUS: STATUS REGISTER

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 34.0 "Instruction Set Summary").

Note 1: The <u>C</u> and <u>DC</u> bits operate as Borrow and Digit Borrow out bits, respectively, in subtraction.

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 b	bit	W = Writable I	bit	U = Unimpler	mented bit, read	as '0'	
u = Bit is uncha	inged	x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other Resets			ther Resets
'1' = Bit is set		'0' = Bit is clea	ared	q = Value dep	pends on condit	ion	

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

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.

3.3.2 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.

3.3.3 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).

3.3.3.1 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 3.6.2 "Linear Data Memory"** for more information.

3.3.4 COMMON RAM

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

3.3.5 DEVICE MEMORY MAPS

The memory maps are shown in Table 3-2 through Table 3-12.

FIGURE 3-2:

BANKED MEMORY PARTITIONING

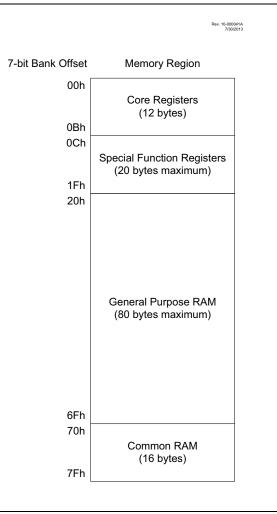


TABLE 3-2: PIC16(L)F1615 MEMORY MAP, BANK 0-7

	BANK 0		BANK 1		BANK 2		BANK 3	_	BANK 4		BANK 5	_	BANK
000h		080h		100h		180h		200h		280h		300h	
	Core Registers (Table 3-1)		Core Regist (Table 3-1										
000	(Table 3-1)		(Table 5-T)	1004	(Table 3-T)	1006	(Table 5-1)	000	(Table 5-T)	0006	(Table 5-1)	2004	(Table 3-
00Bh 00Ch	PORTA	08Bh 08Ch	TRISA	10Bh 10Ch	LATA	18Bh 18Ch	ANSELA	20Bh 20Ch	WPUA	28Bh 28Ch	ODCONA	30Bh 30Ch	SLRCON
00Ch 00Dh	PURTA	08Dh	I RISA	10Ch	LATA	18Dh	ANSELA	20Ch 20Dh	WPUA	280h	ODCONA	30Dh	SLRCON
00Dh 00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Dh	WPUC	28Eh	ODCONC	30Eh	
00Eh	_	08Fh	_	10En		18Fh		20Eh	_	28Fh	-	30Fh	-
010h	PIR1	090h	PIE1	110h	_	190h	_	210h	_	290h	_	310h	_
011h	PIR2	091h	PIE2	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	
012h	PIR3	092h	PIE3	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—
013h	PIR4	093h	PIE4	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—
014h	PIR5	094h	PIE5	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON	295h	—	315h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—
017h	TMR1H	097h	_	117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	_	218h	—	298h	CCPR2L	318h	—
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RC1REG	219h	_	299h	CCPR2H	319h	
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	_
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	—	29Dh	—	31Dh	_
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	_	19Eh	TX1STA	21Eh	—	29Eh	CCPTMRS	31Eh	_
01Fh	T2RST	09Fh	ADCON2	11Fh	_	19Fh	BAUD1CON	21Fh	—	29Fh	—	31Fh	_
020h		0A0h		120h		1A0h		220h		2A0h		320h	
			General		General								
			Purpose		Purpose								
	General		Register 80 Bytes		Register 80 Bytes								
	Purpose Register		oo Dytes		00 Dytes		00 Dytes		oo bytes		00 Dytes		oo bytes
	96 Bytes	0EFh						005		0551		36Fh	
		0EFN		16Fh 170h		1EFh 1F0h		26Fh 270h		2EFh 2F0h		370h	
		01 011	Common RAM	17011	Common RAM		Common RAM	2,011	Common RAM	21 011	Common RAM	07011	Common R
			(Accesses		(Accesse								
07Fh		0FFh	70h – 7Fh)	17Fh	70h – 7Fh)	1FFh	70h – 7Fh)	27Fh	70h – 7Fh)	2FFh	70h – 7Fh)	37Fh	70h – 7Fł

Legend:

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

TABLE 3-3: PIC16(L)F1619 MEMORY MAP, BANK 0-7

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK
000h		080h		100h		180h		200h		280h		300h	
	Core Registers		Core Registers		Core Registers		Core Registers		Core Registers		Core Registers		Core Regist
	(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCON
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCON
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCON
00Fh	_	08Fh		10Fh	_	18Fh	_	20Fh	—	28Fh	—	30Fh	_
010h	PIR1	090h	PIE1	110h	—	190h	—	210h	—	290h	—	310h	
011h	PIR2	091h	PIE2	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	CCP1RL	311h	_
012h	PIR3	092h	PIE3	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	CCP1RH	312h	—
013h	PIR4	093h	PIE4	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	
014h	PIR5	094h	PIE5	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON	295h	_	315h	_
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	_	316h	—
017h	TMR1H	097h		117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	_	218h	_	298h	CCP2RL	318h	_
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RC1REG	219h	—	299h	CCP2RH	319h	_
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	_
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	_	29Bh	CCP2CAP	31Bh	_
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	SP1BRGH	21Ch	_	29Ch	_	31Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh		19Dh	RC1STA	21Dh		29Dh		31Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh		19Eh	TX1STA	21Eh		29Eh	CCPTMRS	31Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	_	19Fh	BAUD1CON	21Fh	—	29Fh	—	31Fh	—
020h		0A0h		120h		1A0h		220h		2A0h		320h	
			General		General								
	General		Purpose		Purpose								
	Purpose		Register		Register								
	Register 96 Bytes		80 Bytes		80 Bytes		80 Bytes		80 Bytes		80 Bytes		80 Bytes
	30 Dytes	0EFh		16Fh		1EFh		26Fh		2EFh		36Fh	
		0F0h	Common RAM	170h	Common RAM	1F0h	Common RAM	270h	Common RAM	2F0h	Common RAM	370h	Common R
			(Accesses		(Accesse								
07Fh		0FFh	70h – 7Fh)	17Fh	70h – 7Fh)	1FFh	70h – 7Fh)	27Fh	70h – 7Fh)	2FFh	70h – 7Fh)	37Fh	70h – 7Fł

Legend:

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

TABLE 3-4: PIC16(L)F1615/9 MEMORY MAP, BANK 8-15

	-	(,	-	,	-	-						
	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 1
400h		480h		500h		580h		600h		680h		700h	
	Core Registers		Core Registers		Core Registers		Core Registers		Core Registers		Core Registers		Core Regis
	(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-1)		(Table 3-7
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh	
40Ch	—	48Ch	—	50Ch		58Ch	PID1SETL	60Ch	PID1Z2L	68Ch	—	70Ch	
40Dh	—	48Dh	—	50Dh	_	58Dh	PID1SETH	60Dh	PID1Z2H	68Dh	—	70Dh	_
40Eh	HDRVENC	48Eh	—	50Eh		58Eh	PID1INL	60Eh	PID1Z2U	68Eh	—	70Eh	
40Fh	—	48Fh	—	50Fh	_	58Fh	PID1INH	60Fh	PID1ACCLL	68Fh	—	70Fh	_
410h	—	490h	—	510h	—	590h	PID1K1L	610h	PID1ACCLH	690h	—	710h	-
411h	—	491h	—	511h	_	591h	PID1K1H	611h	PID1ACCHL	691h	CWG1DBR	711h	WDTCON
412h	—	492h	—	512h	—	592h	PID1K2L	612h	PID1ACCHH	692h	CWG1DBF	712h	WDTCON
413h	TMR4	493h	TMR3L	513h	_	593h	PID1K2H	613h	PID1ACCUL	693h	CWG1AS0	713h	WDTPSI
414h	PR4	494h	TMR3H	514h	—	594h	PID1K3L	614h	PID1CON	694h	CWG1AS1	714h	WDTPSH
415h	T4CON	495h	T3CON	515h	_	595h	PID1K3H	615h	—	695h	CWG10C0N0	715h	WDTTM
416h	T4HLT	496h	T3GCON	516h		596h	PID10UTLL	616h		696h	CWG1CON0	716h	
417h	T4CLKCON	497h	—	517h		597h	PID10UTLH	617h	PWM3DCL	697h	CWG1CON1	717h	
418h	T4RST	498h	_	518h	_	598h	PID10UTHL	618h	PWM3DCH	698h	—	718h	SCANLAD
419h	—	499h	—	519h	_	599h	PID10UTHH	619h	PWM3CON	699h	CWG1CLKCON	719h	SCANLAD
41Ah	TMR6	49Ah	TMR5L	51Ah	_	59Ah	PID10UTUL	61Ah	PWM4DCL	69Ah	CWG1ISM	71Ah	SCANHAD
41Bh	PR6	49Bh	TMR5H	51Bh	—	59Bh	PID1Z1L	61Bh	PWM4DCH	69Bh	—	71Bh	SCANHAD
41Ch	T6CON	49Ch	T5CON	51Ch	_	59Ch	PID1Z1H	61Ch	PWM4CON	69Ch	—	71Ch	SCANCO
41Dh	T6HLT	49Dh	T5GCON	51Dh	_	59Dh	PID1Z1U	61Dh	—	69Dh	—	71Dh	SCANTRI
41Eh	T6CLKCON	49Eh	—	51Eh	_	59Eh	_	61Eh	—	69Eh	—	71Eh	_
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	_	61Fh	_	69Fh	—	71Fh	
420h		4A0h		520h		5A0h		620h	General Purpose	6A0h		720h	
	General		General		General		General		Register				
	Purpose		Purpose		Purpose		Purpose	64Fh	48 Bytes		Unimplemented		Unimpleme
	Register		Register		Register		Register	640h			Read as '0'		Read as '
	80 Bytes		80 Bytes		80 Bytes		80 Bytes		Unimplemented				
	, ,		,		,		, ,		Read as '0'				
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh	
470h		4F0h		570h		5F0h		670h		6F0h		770h	
	Accesses		Accesses		Accesses		Accesses		Accesses		Accesses		Accesse
47Fh	70h – 7Fh	4FFh	70h – 7Fh	57Fh	70h – 7Fh	5FFh	70h – 7Fh	67Fh	70h – 7Fh	6FFh	70h – 7Fh	77Fh	70h – 7F
47FN		4660		57 FN		SPEN		0/FI		OFFI		//F0	

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

© 2014-2017 Microchip Technology Inc.

TABLE 3-5: PIC16(L)F1615/9 MEMORY MAP, BANK 16-23

	BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 2
800h	Core Registers (Table 3-1)	880h	Core Registers (Table 3-1)	900h	Core Registers (Table 3-1)	980h	Core Registers (Table 3-1)	A00h	Core Registers (Table 3-1)	A80h	Core Registers (Table 3-1)	B00h	Core Regist (Table 3-1
80Bh		88Bh		90Bh		98Bh		A0Bh		A8Bh		B0Bh	
80Ch	AT1RESL	88Ch	AT1CLK	90Ch		98Ch		A0Ch		A8Ch		B0Ch	
80Dh	AT1RESH	88Dh	AT1SIG										
80Eh	AT1MISSL	88Eh	AT1CSEL1										
80Fh	AT1MISSH	88Fh	AT1CC1L										
810h	AT1PERL	890h	AT1CC1H										
811h	AT1PERH	891h	AT1CCON1										
812h	AT1PHSL	892h	AT1CSEL2										
813h	AT1PHSH	893h	AT1CC2L										
814h	AT1CON0	894h	AT1CC2H										
815h	AT1CON1	895h	AT1CCON2		Unimplemented		Unimplemented		Unimplemented		Unimplemented		Unimplemer
816h	AT1IR0	896h	AT1CSEL2		Read as '0'		Read as '0'		Read as '0'		Read as '0'		Read as '
817h	AT1IE0	897h	AT1CC3L										
818h	AT1IR1	898h	AT1CC3H										
819h	AT1IE1	899h	AT1CCON3										
81Ah	AT1STPTL	89Ah											
81Bh	AT1STPTH												
81Ch	AT1ERRL												
81Dh	AT1ERRH												
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh	
870h		8F0h		90F11 970h		9F0h		A0FII A70h		AF0h		B70h	
07011	Accesses	01 011	Accesses	37011	Accesses	51 011	Accesses	7,01	Accesses		Accesses	5,01	Accesses
87Fh	70h – 7Fh	8FFh	70h – 7Fh	97Fh	70h – 7Fh	9FFh	70h – 7Fh	A7Fh	70h – 7Fh	AFFh	70h – 7Fh	B7Fh	70h – 7F

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

TABLE 3-6: PIC16(L)F1615/9 MEMORY MAP, BANK 24-31

	BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 3
C00h	Core Registers (Table 3-1)	C80h	Core Registers (Table 3-1)	D00h	Core Registers (Table 3-1)	D80h	Core Registers (Table 3-1)	E00h	Core Registers (Table 3-1)	E80h	Core Registers (Table 3-1)	F00h	Core Regis (Table 3-
C0Bh		C8Bh		D0Bh		D8Bh		E0Bh		E8Bh		F0Bh	
C0Ch	—	C8Ch	_	D0Ch	_	D8Ch		E0Ch		E8Ch		F0Ch	
C0Dh	_	C8Dh	_	D0Dh	_	D8Dh		E0Dh		E8Dh		F0Dh	
C0Eh	—	C8Eh	_	D0Eh	_	D8Eh		E0Eh		E8Eh		F0Eh	
C0Fh	—	C8Fh	_	D0Fh	_	D8Fh		E0Fh		E8Fh		F0Fh	
C10h	—	C90h	_	D10h	_	D90h		E10h		E90h		F10h	
C11h	—	C91h	_	D11h	_	D91h		E11h		E91h		F11h	
C12h	—	C92h	—	D12h	—	D92h		E12h		E92h		F12h	
C13h	—	C93h	—	D13h	—	D93h		E13h		E93h		F13h	
C14h	—	C94h	—	D14h	—	D94h		E14h		E94h		F14h	
C15h	_	C95h	—	D15h	_	D95h		E15h		E95h		F15h	
C16h	_	C96h	—	D16h	—	D96h		E16h		E96h		F16h	
C17h	_	C97h	—	D17h	—	D97h	See Table 3-7 for	E17h	See Table 3-8 for	E97h	See Table 3-9	F17h	See Table
C18h	_	C98h	—	D18h	—	D98h	register mapping	E18h	register mapping	E98h	and Table 3-10	F18h	for register
C19h	_	C99h	—	D19h	_	D99h	details	E19h	details	E99h	for register mapping details	F19h	ping deta
C1Ah	_	C9Ah	—	D1Ah	_	D9Ah		E1Ah		E9Ah	mapping details	F1Ah	
C1Bh	_	C9Bh	—	D1Bh	_	D9Bh		E1Bh		E9Bh		F1Bh	
C1Ch	_	C9Ch	—	D1Ch	—	D9Ch		E1Ch		E9Ch		F1Ch	
C1Dh	_	C9Dh	—	D1Dh	—	D9Dh		E1Dh		E9Dh		F1Dh	
C1Eh	_	C9Eh	—	D1Eh	—	D9Eh		E1Eh		E9Eh		F1Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh		E1Fh		E9Fh		F1Fh	
C20h		CA0h		D20h		DA0h		E20h		EA0h		F20h	
	Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'								
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh	
C70h		CF0h		D70h		DF0h		E70h		EF0h		F70h	
	Accesses 70h – 7Fh		Accesse 70h – 7F										
CFFh		CFFh		D7Fh		DFFh		E7Fh		EFFh		F7Fh	

Legend:

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

TABLE 3-7:PIC16(L)F1615/9 MEMORY
MAP, BANK 27

		Bank 27
	D8Ch	
	D8Ch D8Dh	SMT1TMRH
	D8Eh	SMT1TMRU
	D8En	SMT1CPRL
	Dorn D90h	SMT1CPRH
	D9011 D91h	SMT1CPRU
	D9111 D92h	SMT1CPWL
	D9211 D93h	SMT1CPWH
	D9311 D94h	SMT1CPWU
	D9411 D95h	SMT1PRL
	D9511 D96h	SMT1PRH
	D9011 D97h	SMT1PRU
	D9711 D98h	SMT1CON0
	D980 D99h	SMT1CON1
	D99h D9Ah	SMT1STAT
	D9An D9Bh	SMT1CLK
	D9BN D9Ch	SMT1SIG
	D9Ch D9Dh	SMT1010
	D9Dh D9Eh	SMT1WIN
	D9En D9Fh	SMT2TMRH
	DA0h	SMT2TMRU
	DA01	SMT2CPRL
	DA III DA2h	SMT2CPRH
	DA2h DA3h	SMT2CPRU
	DA3h DA4h	SMT2CPWL
	DA4n DA5h	SMT2CPWH
	DASh DA6h	SMT2CPWU
	DA01 DA7h	SMT2PRL
	DA/II DA8h	SMT2PRH
	DA9h	SMT2PRU
	DAAh	SMT2CON0
	DABh	SMT2CON1
	DACh	SMT2STAT
	DADh	SMT2CLK
	DAEh	SMT2SIG
	DAFh	SMT2WIN
	DB0h	
		_
	DEFh	
Legend:		Jnimplemented data memory locations,
	read	i as '0'.

TABLE 3-8:PIC16(L)F1615/9 MEMORY
MAP, BANK 28

		Bank 28	
		Dank 20	l
	E0Ch E0Dh		
	E0Dh E0Eh		
	E0Fh	PPSLOCK	
	E10h	INTPPS	
	E11h	TOCKIPPS	
	E12h	T1CKIPPS	
	E13h	T1GPPS	
	E14h	CCP1PPS	
	E15h	CCP2PPS	
	E16h	ATINPPS	
	E17h	CWGINPPS	
	E18h	T2PPS	
	E19h	T3CKIPPS	
	E1Ah	T3GPPS	
	E1Bh	T4PPS	
	E1Ch	T5CKIPPS	
	E1Dh	T5GPPS	
	E1Eh	T6PPS	
	E1Fh	ATCC1PPS	
	E20h	SSPCLKPPS	
	E21h	SSPDATPPS	
	E22h	SSPSSPPS	
	E23h	ATCC2PPS	
	E24h	RXPPS	
	E25h	CKPPS	
	E26h	SMT1SIGPPS	
	E27h	SMT1WINPPS	
	E28h	CLCIN0PPS	
	E29h	CLCIN1PPS	
	E2Ah	CLCIN2PPS	
	E2Bh	CLCIN3PPS	
	E2Ch	SMT2SIGPPS	
	E2Dh	SMT2WINPPS	
	E2Eh	ATCC3PPS	
	E2Fh		
		_	
	E6Fh		
	-		I
Legend:		Unimplemented data me d as '0'.	mory lo

TABLE 3-9: PIC16(L)F1615 MEMORY MAP, BANK 29

		Bank 29	
I	E8Ch		
1	E8Dh		
	E8Eh		
	E8Fh		
	E90h	RA0PPS	
	E91h	RA1PPS	
	E92h	RA2PPS	
	E93h	_	
	E94h	RA4PPS	
	E95h	RA5PPS	
	E96h		
	E97h		
	E98h		
	E99h		
	E9Ah		
	E9Bh		
1	E9Ch		
1	E9Dh		
	E9Eh		
	E9Fh		
	EA0h	RC0PPS	
	EA1h	RC1PPS	
	EA2h	RC2PPS	
	EA3h	RC3PPS	
	EA4h	RC4PPS	
1	EA5h	RC5PPS	
	EA6h		
		_	
	EEFh		
			I
Legend:		Inimplemented data me	mory locations,
	read	i as '0'.	

TABLE 3-10:PIC16(L)F1619 MEMORY MAP,
BANK 29

		Bank 29	
	E8Ch		1
	E8Dh		
	E8Eh		
	E8Fh		
	E90h	RA0PPS	
	E91h	RA1PPS	
	E92h	RA2PPS	
	E93h		
	E94h	RA4PPS	
	E95h	RA5PPS	
	E96h		
	E97h		
	E98h		
	E99h		
	E9Ah		
	E9Bh		
	E9Ch	RB4PPS	
	E9Dh	RB5PPS	
	E9Eh	RB6PPS	
	E9Fh	RB7PPS	
	EA0h	RC0PPS	
	EA1h	RC1PPS	
	EA2h	RC2PPS	
	EA3h	RC3PPS	
	EA4h	RC4PPS	
	EA5h	RC5PPS	
	EA6h	RC6PPS	
	EA7h	RC7PPS	
	EA8h		
		_	
	EEFh		
			1
Legend:		Unimplemented data me id as '0'.	emory locations,

TABLE 3-11:PIC16(L)F1615/9 MEMORYMAP, BANK 30

	•	
		Bank 30
	F0Ch	
	F0Dh	
	F0Eh	
	F0Fh	CLCDATA
	F10h	CLC1CON
	F11h	CLC1POL
	F12h	CLC1SEL0
	F13h	CLC1SEL1
	F14h	CLC1SEL2
	F15h	CLC1SEL3
	F16h	CLC1GLS0
	F17h	CLC1GLS1
	F1711 F18h	CLC1GLS2
		CLC1GLS3
	F19h	CLC2CON
	F1Ah	
	F1Bh	CLC2POL
	F1Ch	CLC2SEL0
	F1Dh	CLC2SEL1
	F1Eh	CLC2SEL2
	F1Fh	CLC2SEL3
	F20h	CLC2GLS0
	F21h	CLC2GLS1
	F22h	CLC2GLS2
	F23h	CLC2GLS3
	F24h	CLC3CON
	F25h	CLC3POL
	F26h	CLC3SEL0
	F27h	CLC3SEL1
	F28h	CLC3SEL2 CLC3SEL3
	F29h F2Ah	CLC3GLS0
	F2An F2Bh	CLC3GLS0
	F2Ch	CLC3GLS1
	F2Dh	CLC3GLS3
	F2Eh	CLC4CON
	F2Fh	CLC4POL
	F30h	CLC4SEL0
	F31h	CLC4SEL1
	F32h	CLC4SEL2
	F33h	CLC4SEL3
	F34h	CLC4GLS0
	F35h	CLC4GLS1
	F36h	CLC4GLS2
	F37h	CLC4GLS3
	F38h	
		—
	F6Fh	
Legend:		Unimplemented data memory locations, ad as '0'.
1		

TABLE 3-12: PIC16(L)F1615/9 MEMORY MAP, BANK 31

		Bank 31	
F	F8Ch		
		Unimplemented Read as '0'	
I	FE3h		
I	FE4h	STATUS_SHAD	
I	FE5h	WREG_SHAD	
I	FE6h	BSR_SHAD	
I	FE7h	PCLATH_SHAD	
I	FE8h	FSR0L_SHAD	
-	FE9h	FSR0H_SHAD	
-	FEAh	FSR1L_SHAD	
-	FEBh	FSR1H_SHAD	
-	FECh		
-	FEDh	STKPTR	
-	FEEh	TOSL	
I	FEFh	TOSH	
Legend:		Jnimplemented data me d as '0'.	emory locations,

3.3.6 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-13 can be addressed from any Bank.

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		this location ical register)		xxxx xxxx	uuuu uuuu					
x01h or x81h	INDF1		this location ical register)		XXXX XXXX	uuuu uuuu					
x02h or x82h	PCL	Program C	ounter (PC) I	Least Signifi	cant Byte					0000 0000	0000 0000
x03h or x83h	STATUS	_			TO	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 Hig	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 Hię	gh Pointer					0000 0000	0000 0000
x08h or x88h	BSR	_	_	_			BSR<4:0>			0 0000	0 0000
x09h or x89h	WREG	Working Re	Working Register								uuuu uuuu
x0Ahor x8Ah	PCLATH	_	Write Buffer	for the upp		-000 0000	-000 0000				
x0Bhor x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000

TABLE 3-13: CORE FUNCTION REGISTERS SUMMARY

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

TABLE 3-14: SPECIAL FUNCTION REGISTER SUMMARY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D								
PORTA	—	_	RA5	RA4	RA3	RA2	RA1	RA0
PORTB ⁽⁴⁾	RB7	RB6	RB5	RB4	—	—	_	_
PORTC	RC7 ⁽⁴⁾	RC6 ⁽⁴⁾	RC5	RC4	RC3	RC2	RC1	RC0
—	Unimplemented	l						
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIR2	OSFIF	C2IF	C1IF	_	BCL1IF	TMR6IF	TMR4IF	CCP2IF
PIR3	—	_	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF
TMR0	Holding Registe	er for the 8-bit Tin	ner0 Count					
TMR1L	Holding Registe	er for the Least S	ignificant Byte of	the 16-bit TMR	1 Count			
TMR1H	Holding Registe	er for the Most Si	gnificant Byte of	the 16-bit TMR1	Count			
T1CON	TMR1C	S<1:0>	T1CKP	S<1:0>	_	T1SYNC		TMR10N
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T <u>1GGO</u> / DONE	T1GVAL	T1GSS	S<1:0>
TMR2	Timer2 Module	Register						
PR2	Timer2 Period F	Register						
T2CON	ON		CKPS<2:0>	> OUTPS<3:0>				
T2HLT	PSYNC	CKPOL	CKSYNC		MODE<4:0>			
T2CLKCON	—				— CS<3:0>			
T2RST						RSEL	<3:0>	
	D PORTA PORTB ⁽⁴⁾ PORTC — PIR1 PIR2 PIR3 PIR4 PIR5 TMR0 TMR1L TMR1H T1GCON TMR2 PR2 T2CON T2HLT T2CLKCON	PORTA — PORTB ⁽⁴⁾ RB7 PORTC RC7 ⁽⁴⁾ — Unimplemented PIR1 TMR1GIF PIR2 OSFIF PIR3 — PIR4 SCANIF PIR5 TMR3GIF TMR1L Holding Registe TMR1H Holding Registe T1CON TMR1C T1GCON TMR1GE TMR2 Timer2 Module PR2 Timer2 Period F T2CON ON T2CLKCON —	PORTA — — PORTB ⁽⁴⁾ RB7 RB6 PORTC RC7 ⁽⁴⁾ RC6 ⁽⁴⁾ — Unimplemented PIR1 TMR1GIF ADIF PIR2 OSFIF C2IF PIR3 — — PIR4 SCANIF CRCIF PIR5 TMR3GIF TMR3IF TMR0 Holding Register for the 8-bit Tir TMR1L Holding Register for the Least S TMR1H Holding Register for the Most Si T1CON TMR1GE T1GPOL TMR2 Timer2 Module Register PR2 Timer2 Period Register T2CON ON T2HLT PSYNC CKPOL T2CLKCON — —	PORTA—RA5PORTB(4)RB7RB6RB5PORTCRC7(4)RC6(4)RC5—UnimplementedRC1FPIR1TMR1GIFADIFRCIFPIR2OSFIFC2IFC1IFPIR3——CWGIFPIR4SCANIFCRCIFSMT2PWAIFPIR5TMR3GIFTMR3IFTMR5GIFTMR0Holding Register for the 8-bit Timer0 CountTMR1LHolding Register for the Most Significant Byte ofT1CONTMR1CS<1:0>T1CKPT1GCONTMR1GET1GPOLT1GTMTMR2Timer2 Module RegisterT1GTMT2CONONCKPS<2:0>T2HLTPSYNCCKPOLCKSYNCT2CLKCON———	PORTA—RA5RA4PORTB(4)RB7RB6RB5RB4PORTCRC7(4)RC6(4)RC5RC4—UnimplementedRC1FTXIFPIR1TMR1GIFADIFRCIFTXIFPIR2OSFIFC2IFC1IF—PIR3——CWGIFZCDIFPIR4SCANIFCRCIFSMT2PWAIFSMT2PRAIFPIR5TMR3GIFTMR3IFTMR5GIFTMR5IFTMR0Holding Register for the 8-bit Timer0 CountTMR5IFTMR1LHolding Register for the Most Significant Byte of the 16-bit TMR1TICONTMR1CS<1:0>T1CKPS<1:0>TIGCONTMR1GET1GPOLT1GTMT1GSPMTMR2Timer2 Module RegisterT1GTMT1GSPMTMR2Timer2 Period RegisterTT1GTMT1GSPMT2CONONCKPOLCKSYNC	PORTA—RA5RA4RA3PORTBRB7RB6RB5RB4—PORTCRC7(4)RC6(4)RC5RC4RC3—UnimplementedRC1FTXIFSSP1IFPIR1TMR1GIFADIFRCIFTXIFSSP1IFPIR2OSFIFC2IFC1IF—BCL1IFPIR3——CWGIFZCDIFCLC4IFPIR4SCANIFCRCIFSMT2PWAIFSMT2PRAIFSMT2IFPIR5TMR3GIFTMR3IFTMR5GIFTMR5IF—TMR0Holding Register for the 8-bit Timer0 CountTMR1LHolding Register for the Most Significant Byte of the 16-bit TMR1 CountT1CONTMR1HHolding Register for the Most Significant Byte of the 16-bit TMR1 CountT1GGO/	PORTA—RA5RA4RA3RA2PORTB(4)RB7RB6RB5RB4——PORTCRC7(4)RC6(4)RC5RC4RC3RC2Unimplemented————PIR1TMR1GIFADIFRCIFTXIFSSP1IFCCP1IFPIR2OSFIFC2IFC1IF—BCL1IFTMR6IFPIR3——CWGIFZCDIFCLC4IFCLC3IFPIR4SCANIFCRCIFSMT2PWAIFSMT2PRAIFSMT2IFSMT1PWAIFPIR5TMR3GIFTMR3IFTMR5GIFTMR5IF—AT1IFTMR0Holding Register for the 8-bit Timer0 CountTMR5IF—T1SYNCTMR1LHolding Register for the Most Significant Byte of the 16-bit TMR1 CountT1SYNCT1GVALTMR2Timer2 Module RegisterT1GPOLT1GTMT1GSPMT1GGO/ DONET1GVALTMR2Timer2 Period RegisterT1GCNOUTPOUTPT2HLTPSYNCCKPOLCKSYNCMODE<4:0>T2CLKCON—————CS	PORTA — — RA5 RA4 RA3 RA2 RA1 PORTB(4) RB7 RB6 RB5 RB4 — — — — PORTC RC7(4) RC6(4) RC5 RC4 RC3 RC2 RC1 — Unimplemented — — — — — — PIR1 TMR1GIF ADIF RCIF TXIF SSP1IF CCP1IF TMR2IF PIR2 OSFIF C2IF C1IF — BCL1IF TMR6IF TMR2IF PIR3 — — CWGIF ZCDIF CLC4IF CLC3IF CLC2IF PIR4 SCANIF CRCIF SMT2PRAIF SMT2IF SMT1PWAIF SMT1PRAIF PIR5 TMR30IF TMR3IF TMR3F TMR5GIF TMR1G AT11F PID1EIF TMR0 Holding Register for the 8-bit Timer0 Count T TMR1L Holding Register for the Most Significant Byte of the 16-bit TMR1 Count T11GN T1GSPM

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as Note1:PIC16F1615/9 only.2:Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

					`	/				
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Bank	1									
08Ch	TRISA	—	—	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	
08Dh	TRISB ⁽⁴⁾	TRISB7	TRISB6	TRISB5	TRISB4	_	—	_	—	
08Eh	TRISC	TRISC7 ⁽⁴⁾	TRISC6 ⁽⁴⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	
08Fh	—	Unimplemented	1							
090h	—	Unimplemented	1							
090h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	
091h	PIE2	OSFIE	C2IE	C1IE	TMR4IE	CCP2IE				
092h	PIE3		—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	
093h	PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	
094h	PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	_	AT1IE	PID1EIE	PID1DIE	
095h	OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		
096h	PCON	STKOVF	STKUNF	WDTWV	RWDT	RMCLR	RI	POR	BOR	
097h	—	Unimplemented	1							
098h	OSCTUNE	—	_			TUN	<5:0>			
099h	OSCCON	SPLLEN		IRCF	<3:0>		—	SCS	<1:0>	
09Ah	OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	
09Bh	ADRESL	ADC Result Re	gister Low							
09Ch	ADRESH	ADC Result Re	gister High							
09Dh	ADCON0		CHS<4:0> GO/DONE AI					ADON		
09Eh	ADCON1	ADFM	ADCS<2:0> — — ADPREF<1:0>					F<1:0>		
09Fh	ADCON2		TRIGSEL<4:0> — — —						—	
1										

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as **Note 1:** PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

					<u> </u>				
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank	2								
10Ch	LATA			LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
10Dh	LATB ⁽⁴⁾	LATB7	LATB6	LATB5	LATB4				
10Eh	LATC	LATC7 ⁽⁴⁾	LATC6 ⁽⁴⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
10Fh	_	Unimplemented	t T						
110h	<u> </u>	Unimplemented	1 <u> </u>						
111h	CM1CON0	C10N	C1OUT	_	C1POL		C1SP	C1HYS	C1SYNC
112h	CM1CON1	C1INTP	C1INTN	C1PC	H<1:0>			C1NCH<2:0>	
113h	CM2CON0 ⁽⁴⁾	C2ON	C2OUT		C2POL		C2SP	C2HYS	C2SYNC
114h	CM2CON1 ⁽⁴⁾	C2INTP	C2INTN	C2PC	H<1:0>			C2NCH<2:0>	
115h	CMOUT			—	—	—	—	MC2OUT	MC1OUT
116h	BORCON	SBOREN	BORFS		<u> </u>				BORRDY
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFV	/R<1:0>	ADFVI	R<1:0>
118h	DAC1CON0	DAC1EN	_	DAC10E1	'	DAC1P	SS<1:0>	—	
119h	DAC1CON1				DAC1F	R<7:0>			
11Ah		Unimplemented	E						
11Bh		Unimplemented	T						
11Ch	ZCD1CON	ZCD1EN		ZCD10UT	ZCD1POL			ZCD1INTP	ZCD1INTN
11Dh		Unimplemented	t t						
11Eh	_	Unimplemented	E						
11Fh	—	Unimplemented	L						

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as **Note 1:** PIC16F1615/9 only.

2: Unimplemented, read as '1'.

PIC16(L)F1615 only. 3:

						-			
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank	3								
18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0
18Dh	ANSELB ⁽⁴⁾	—	—	ANSB5	ANSB4	—	—	—	—
18Eh	ANSELC	ANSC7 ⁽⁴⁾	ANSC6 ⁽⁴⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0
18Fh	—	Unimplemented	I						
190h	—	Unimplemented	nplemented						
191h	PMADRL	Flash Program	ash Program Memory Address Register Low Byte						
192h	PMADRH	(2)	—(2) Flash Program Memory Address Register High Byte						
193h	PMDATL	Flash Program	lash Program Memory Read Data Register Low Byte						
194h	PMDATH	—	—	Flash Program	Memory Read D	ata Register Hig	h Byte		
195h	PMCON1	(2)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
196h	PMCON2	Flash Program	Memory Control	Register 2					
197h	VREGCON ⁽¹⁾	—	—	—	—	—	—	VREGPM	Reserved
198h	—	Unimplemented	I						
199h	RC1REG	EUSART Recei	ve Data Register	ſ					
19Ah	TX1REG	EUSART Trans	mit Data Registe	r					
19Bh	SP1BRGL	Baud Rate Gen	erator Data Regi	ister Low					
19Ch	SP1BRGH	Baud Rate Gen	Baud Rate Generator Data Register High						
19Dh	RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
19Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
19Fh	BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16		WUE	ABDEN

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) **TABLE 3-14:**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Bank	4									
20Ch	WPUA	—	_	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	
20Dh	WPUB ⁽⁴⁾	WPUB7	WPUB6	WPUB5	WPUB4	—	_	_	—	
20Eh	WPUC	WPUC7 ⁽⁴⁾	WPUC6 ⁽⁴⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	
20Fh	—	Unimplemented								
210h	—	Unimplemented								
211h	SSP1BUF		Synchronous Serial Port Receive Buffer/Transmit Register							
212h	SSP1ADD				ADD	<7:0>				
213h	SSP1MSK				MSK	<7:0>				
214h	SSP1STAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	
215h	SSP1CON1	WCOL	SSPOV	SSPEN	СКР		SSPN	/<3:0>		
216h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	
217h	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	
218h to 21Fh	_	Unimplemented						· · ·		

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

Unimplemented, read as '1'. 2:

PIC16(L)F1615 only. 3:

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 5	5								
	ODCONA	[<u> </u>		ODA5	ODA4		ODA2	ODA1	ODA0
28Dh	ODCONB ⁽⁴⁾	ODB7	ODB6	ODB5	ODB4			_	_
28Eh	ODCONC	ODC7 ⁽⁴⁾	ODC6 ⁽⁴⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0
28Fh	<u> </u>	Unimplemented	1						
290h	<u> </u>	Unimplemented							
-	CCP1RL		are/PWM 1 Regis	, ,					
-	CCP1RH	Capture/Compa	are/PWM 1 Regis	ster (MSB)					_
293h	CCP1CON	EN	<u> </u>	OUT	FMT		MODE	E<3:0>	
294h	CCP1CAP	_	<u> </u>	—	<u> </u>			CTS<2:0>	
295h 297h	—	Unimplemented	Inimplemented						
	CCP2RL	Capture/Compa	are/PWM 2 Regis	ster (LSB)					
299h	CCP2RH	Capture/Compa	are/PWM 2 Regis	ter (MSB)	<u> </u>				
	CCP2CON	EN	<u> </u>	OUT	FMT		MODE	E<3:0>	
29Bh	CCP2CAP				-	-		CTS<2:0>	
29Ch	—	Unimplemented	1						
29Dh		Unimplemented	1						
	CCPTMRS	-	EL<1:0>	P3TSE	EL<1:0>	C2TSE	EL<1:0>	C1TSE	EL<1:0>
29Fh	<u> </u>	Unimplemented	1						
Bank 6	-	+	•	•	.	<u> </u>		+	+
	SLRCONA	<u> </u>	<u> </u>	SLRA5	SLRA4		SLRA2	SLRA1	SLRA0
	SLRCONB ⁽⁴⁾	SLRB7	SLRB6	SLRB5	SLRB4	—	—	-	—
30Eh	SLRCONC	SLRC7 ⁽⁴⁾	SLRC6 ⁽⁴⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
30Fh 	—	Unimplemented							
Legend	: x = unknown, u		g = value depenr	Js on condition,	- = unimplement	.ed, r = reserved	. Shaded locatio	ons are unimplem	ented, read as

Note 1: PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

						/			
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank	7								
38Ch	INLVLA			INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
38Dh	INLVLB ⁽⁴⁾	INLVLB7	INLVLB6	INLVLB5	INLVLB4				_
38Eh	INLVLC	INLVLC7 ⁽⁴⁾	INLVLC6 ⁽⁴⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
30Fh	—	Unimplemented	t						
390h	—	Unimplemented	L						
391h	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
392h	IOCAN		_	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
393h	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
394h	IOCBP ⁽⁴⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	_	—	—
395h	IOCBN ⁽⁴⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	_	—	—	—
396h	IOCBF ⁽⁴⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4		_	_	—
397h	IOCCP	IOCCP7 ⁽⁴⁾	IOCCP6 ⁽⁴⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
398h	IOCCN	IOCCN7 ⁽⁴⁾	IOCCN6 ⁽⁴⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
399h	IOCCF	IOCCF7 ⁽⁴⁾	IOCCF6 ⁽⁴⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
39Ah					·				
to	—	Unimplemented	1						
39Fh	$d_{x} = unknown$ $u = unchanged$ $a = value depends on condition z = unimplemented x = reserved. Shaded locations are unimplemented, read as$								
haana l	$\cdot \cdot \cdot \cdot \cdot = \dots \cdot $	11 = 1000000000000000000000000000000000	$\alpha = value denenc$	de on condition	_ = unimplomont	od v = rocorvod	Shadad locatio	ne ara unimplam	ac hear hatea

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank	8								
40Ch	—	Unimplemented							
40Dh	—	Unimplemented							
40Eh	HIDRVC	_	_	HIDC5	HIDC4	—	—	—	—
40Fh to 412h	_	Unimplemented							
413h	TMR4	Timer4 Module	Register						
414h	PR4	Timer4 Period F	mer4 Period Register						
415h	T4CON	ON		CKPS<2:0>			OUTP	S<3:0>	
416h	T4HLT	PSYNC	CKPOL	CKSYNC			MODE<4:0>		
417h	T4CLKCON	—	—	—	—		CS<	:3:0>	
418h	T4RST	—		—	—		RSEL	<3:0>	
419h	—	Unimplemented							
41Ah	TMR6	Timer6 Module	Register						
41Bh	PR6	Timer6 Period F	Register						
41Ch	T6CON	ON		CKPS<2:0>			OUTP	S<3:0>	
41Dh	T6HLT	PSYNC	CKPOL	CKSYNC			MODE<4:0>		
41Eh	T6CLKCON	—	_	_	_		CS<	:3:0>	
41Fh	T6RST	_	_		_		RSEL	<3:0>	

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

Note 1: PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
Bank	9	-									
48Ch to 492h	_	Unimplemented	l								
493h	TMR3L	Timer3 Module	Register								
494h	TMR3H	Timer3 Module	lodule Register								
495h	T3CON	TMR3C	S<1:0>	T3CKP	S<1:0>	_	T3SYNC	—	TMR3CON		
496h	T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ DONE	T3GVAL	T3GS	S<1:0>		
497h to 499h	_	Unimplemented	nplemented								
49Ah	TMR5L	Timer5 Module	Register								
49Bh	TMR5H	Timer5 Module	Register								
49Ch	T5CON	TMR5C	S<1:0>	T5CKP	S<1:0>	—	T5SYNC	—	TMR5CON		
49Dh	T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T <u>5GGO</u> / DONE	T5GVAL	T5GS	S<1:0>		
49Eh	—	Unimplemented	l								
49Fh	—	Unimplemented									
Bank	10										
50Ch to 51Fh	_	Unimplemented									

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as **Note 1:** PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Bank '	11									
58Ch	PID1SELT				SET	<7:0>				
58Dh	PID1SETH				SET<	<15:8>				
58Eh	PID1INL				IN<	7:0>				
58Fh	PID1INH				IN<1	15:8>				
590h	PID1K1L		K1<7:0>							
591h	PID1K1H				K1<1	15:8>				
592h	PID1K2L				K2<	:7:0>				
593h	PID1K2H				K2<1	15:8>				
594h	PID1K3L				K3<	:7:0>				
595h	PID1K3H				K3<1	15:8>				
596h	PID1OUTLL				OUT<	<7:0>				
597h	PID1OUTLH				OUT<	:15:8>				
598h	PID1OUTHL				OUT </td <td>23:16></td> <td></td> <td></td> <td></td>	23:16>				
599h	PID1OUTHH				OUT<:	31:24>				
59Ah	PID1OUTU	—	—	—	—		OUT<	:35:32>		
59Bh	PID1Z1L				Z1<7	7:0>				
59Ch	PID1Z1H				Z1<1	15:8>				
59Dh	PID1Z1U	—	—	—	—	—	—	—	Z116	
59Eh	—	Unimplemented	1							
59Fh	—	Unimplemented	1							

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
Bank	12										
60Ch	PID1Z2L				Z2<	7:0>					
60Dh	PID1Z2H				Z2<1	15:8>					
60Eh	PID1Z2U	—	—	—	—	—	—	—	Z216		
60Fh	PID1ACCLL				ACC	<7:0>					
610h	PID1ACCLH		ACC<15:8>								
611h	PID1ACCHL		ACC<23:16>								
612h	PID1ACCHH		ACC<31:24>								
613h	PID1ACCU	—	— — — — ACC<34:32>								
614h	PID1CON	EN	BUSY	—	—	—		MODE<2:0>			
615h	_	Unimplemented	ł								
616h	—	Unimplemented	ł								
617h	PWM3DCL	DC<	<1:0>	—	—	—	—	—	—		
618h	PWM3DCH				DC<	<9:2>					
619h	PWM3CON	EN	—	OUT	POL	_	—	—	—		
61Ah	PWM4DCL	DC<	<1:0>	—	—	—	—	—	—		
61Bh	PWM4DCH		DC<9:2>								
61Ch	PWM4CON	EN	—	OUT	POL	_	—	—	—		
61Dh to 61Fh	_	Unimplemented	1								

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Bank 1	13	·	·	·	·		· <u>······</u> ·			
68Ch to 690h	_	Unimplemented	1							
691h	CWG1DBR	<u> </u>	—			DBR	<5:0>			
692h	CWG1DBF	<u> </u>	—			DBF	<5:0>			
693h	CWG1AS0	SHUTDOWN	REN	LSBD	LSBD<1:0> LSAC<1:0> — —					
694h	CWG1AS1		TMR6AS	TMR4AS	TMR2AS		C2AS ⁽⁴⁾	C1AS	INAS	
695h	CWG10C0N0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	
696h	CWG1CON0	EN	LD					MODE<2:0>		
697h	CWG1CON1	<u> </u>	—	IN	—	POLD	POLC	POLB	POLA	
698h	—	Unimplemented	1							
699h	CWG1CLKCON	<u> </u>	_		—		_		CS	
69Ah	CWG1ISM		IS<3:0>							
69Bh to 6EFh	_	Unimplemented	implemented							

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) **TABLE 3-14:**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank	14								
70Ch to 710h	_	Unimplemented	1						
711h	WDTCON0		—			WDTPS<4:0>			SEN
712h	WDTCON1	—		WDTCS<2:0>		—		WINDOW<2:0>	
713h	WDTPSL				PSCN	T<7:0>			
714h	WDTPSH				PSCNT	<15:8>			
715h	WDTTMR			WDTTMR<4:0>		STATE	PSCNT	<17:16>	
716h	—	Unimplemented	l						
717h	—	Unimplemented	I						
718h	SCANLADRL				LADR	8<7:0>			
719h	SCANLADRH				LADR	<15:8>			
71Ah	SCANHADRL				HADF	R<7:0>			
71Bh	SCANHADRH				HADR	<15:8>			
71Ch	SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE	<1:0>
71Dh	SCANTRIG						TSEL	<3:0>	
71Eh	_	Unimplemented							
71Fh		Unimplemented							

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as **Note 1:** PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Banks	s 15								
78Ch to 790h	_	Unimplemented							
791h	CRCDATL				DAT	<7:0>			
792h	CRCDATH				DAT<	15:8>			
793h	CRCACCL				ACC	<7:0>			
794h	CRCACCH				ACC<	:15:8>			
795h	CRCSHIFTL				SHIFT	<7:0>			
796h	CRCSHIFTH				SHIFT	<15:8>			
797h	CRCXORL				XOR<7:1>				—
798h	CRCXORH				XOR<	:15:8>			
799h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL
79Ah	CRCCON1		DLEN	<3:0>			PLEN	<3:0>	
79Bh to 79Fh	_	Unimplemented							

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
Bank '	16										
80Ch	AT1RESL				RES	<7:0>					
80Dh	AT1RESH	—	—	—	—		—	RES	<9:8>		
80Eh	AT1MISSL				MISS	6<7:0>					
80Fh	AT1MISSH				MISS	<15:8>					
810h	AT1PERL				PER	<7:0>					
811h	AT1PERH	POV				PER<14:8>					
812h	AT1PHSL		PHS<7:0>								
813h	AT1PHSH	—						PHS	<9:8>		
814h	AT1CON0	EN	PREC	PS<	<1:0>	POL	_	APMOD	MODE		
815h	AT1CON1	—	PHP	—	PRP	—	MPP	ACCS	VALID		
816h	AT1IR0	—	—	_	—	_	PHSIF	MISSIF	PERIF		
817h	AT1IE0		_	—	—		PHSIE	MISSIE	PERIE		
818h	AT1IR1	<u> </u>			—	—	CC3IF	CC2IF	CC1IF		
819h	AT1IE1	<u> </u>		—	—	—	CC3IE	CC2IE	CC1IE		
81Ah	AT1STPTL				STPT	Γ<7:0>					
81Bh	AT1STPTH	—				STPT<14:8>					
81Ch	AT1ERRL				ERR	<7:0>					
81Dh	AT1ERRH		ERR<15:8>								
81Eh	—	Unimplemented	plemented								
81Fh	—	Unimplemented	l l								

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank 1	17								
88Ch	AT1CLK	—	—	—	—	—	—	—	CS0
88Dh	AT1SIG				—			SSEL<2:0>	
88Eh	AT1CSEL1		—	—	—	<u> </u>		CP1S<2:0>	
88Fh	AT1CC1L				CC1	<7:0>			
890h	AT1CC1H	—	—	—	—		—	CC1 [,]	<9:8>
891h	AT1CCON1	CC1EN		_	CC1POL	CAP1P			CC1MODE
892h	AT1CSEL2	'	—	—	<u> </u>			CP2S<2:0>	
893h	AT1CC2L				CC2·	<7:0>			
894h	AT1CC2H			—	_	—		CC2	<9:8>
895h	AT1CCON2	CC2EN		—	CC2POL	CAP2P		—	CC2MODE
896h	AT1CSEL3	'	—	—	—			CP3S<2:0>	
897h	AT1CC1L				CC3	<7:0>			
898h	AT1CC1H	—	—	—	—	—	—	CC3 [,]	<9:8>
899h	AT1CCON1	CC3EN		—	CC3POL	CAP3P		—	CC3MODE
89Ah to 89Fh	—	Unimplemented							
Bank 1	18-26								
x0Ch/ x8Ch x1Fh/	_	Unimplemented							
x9Fh					·				

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as**Note 1:**PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0			
Banks	\$ 27											
D80h to D8Bh	_	Unimplemented										
D8Ch	SMT1TMRL				SMT1TM	MR<7:0>						
D8Dh	SMT1TMRH				SMT1TM	/IR<15:8>						
D8Eh	SMT1TMRU				SMT1TM	IR<23:16>						
D8Fh	SMT1CPRL				SMT1CF	PR<7:0>						
D90h	SMT1CPRH				SMT1CF	PR<15:8>						
D91h	SMT1CPRU				SMT1CP	PR<23:16>						
D92h	SMT1CPWL				SMT1CF	PW<7:0>						
D93h	SMT1CPWH				SMT1CP	PW<15:8>						
D94h	SMT1CPWU			SMT1CPW<23:16>								
D95h	SMT1PRL		SMT1PR<7:0>									
D96h	SMT1PRH				SMT1PI	R<15:8>						
D97h	SMT1PRU				SMT1PF	R<23:16>						
D98h	SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1P	S<1:0>			
D99h	SMT1CON1	SMT1GO	REPEAT	—	—		MODE	E<3:0>				
D9Ah	SMT1STAT	CPRUP	CPWUP	RST	_		TS	WS	AS			
D9Bh	SMT1CLK	—	—		—	_		CSEL<2:0>				
D9Ch	SMT1SIG		<u> </u>	_			SSEL<4:0>					
D9Dh	SMT1WIN						WSEL<4:0>					
D9Eh	SMT2TMRL				SMT2TN	MR<7:0>						
D9Fh	SMT2TMRH				SMT2TN	/IR<15:8>						
DA0h	SMT2TMRU				SMT2TM	IR<23:16>						
DA1h	SMT2CPRL		SMT2CPR<7:0>									
DA2h	SMT2CPRH		SMT2CPR<15:8>									
DA3h	SMT2CPRU				SMT2CP	PR<23:16>						
DA4h	SMT2CPWL				SMT2CF	PW<7:0>						

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

 Note
 1:
 PIC16F1615/9 only.

 2:
 Unimplemented, read as '1'.

PIC16(L)F1615 only. 3:

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank	27 (Continued)								
DA5h	SMT2CPWH				SMT2CP	W<15:8>			
DA6h	SMT2CPWU				SMT2CP	N<23:16>			
DA7h	SMT2PRL				SMT2P	PR<7:0>			
DA8h	SMT2PRH				SMT2PI	R<15:8>			
DA9h	SMT2PRU				SMT2PF	<23:16>			
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT2F	'S<1:0>
DABh	SMT2CON1	SMT2GO	REPEAT	_	_		MODE	=<3:0>	
DACh	SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS
DADh	SMT2CLK		—	—	—	—		CSEL<2:0>	
DAEh	SMT2SIG		_	—			SSEL<4:0>		
DAFh	SMT2WIN	_	_	_			WSEL<4:0>		

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as **Note 1:** PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) **TABLE 3-14:**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0				
Banks	s 28												
E0Ch to E0Eh	—	Unimplemented											
E0Fh	PPSLOCK	_	_		—	—		—	PPSLOCKED				
E10h	INTPPS	—		—			INTPPS<4:0>						
E11h	T0CKIPPS	—		—			T0CKIPPS<4:0>	>					
E12h	T1CKIPPS	_		_			T1CKIPPS<4:0>	•					
E13h	T1GPPS	_		_	T1GPPS<4:0>								
E14h	CCP1PPS	—		—			CCP1PPS<4:0>	•					
E15h	CCP2PPS	_		_			CCP2PPS<4:0>	•					
E16h	ATINPPS	_		_			ATINPPS<4:0>						
E17h	CWGINPPS	—		—		(CWGINPPS<4:0	>					
E18h	T2PPS	_		_			T2PPS<4:0>						
E19h	T3CKIPPS	_		_			T3CKIPPS<4:0>	•					
E1Ah	T3GPPS	—		—			T3GPPS<4:0>						
E1Bh	T4PPS	_		_			T4PPS<4:0>						
E1Ch	T5CKIPPS	_		_			T5CKIPPS<4:0>	•					
E1Dh	T5GPPS	_		_			T5GPPS<4:0>						
E1Eh	T6PPS	_		_			T6PPS<4:0>						
E1Fh	ATCC1PPS		-	—			ATCC1PPS<4:02	>					
E20h	SSPCLKPPS(3)	_		_		S	SPCLKPPS<4:0)>					
E20h	SSPCLKPPS(4)	_		_		S	SPCLKPPS<4:0)>					
E21h	SSPDATPPS ⁽³⁾	—	_	_	SSPDATPPS<4:0>								
E21h	SSPDATPPS ⁽⁴⁾	—	_	—	SSPDATPPS<4:0>								
E22h	SSPSSPPS ⁽³⁾	—	_	_	SSPSSPPS<4:0>								
E22h	SSPSSPPS ⁽⁴⁾	—	_	_	SSPSSPPS<4:0>								
E23h	ATCC2PPS	_	_				ATCC2PPS<4:02	>					

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

Note 1: PIC16F1615/9 only. Unimplemented, read as '1'. 2:

3: PIC16(L)F1615 only.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
Banks	s 28 (Continued)	1									
E24h	RXPPS ⁽³⁾	—	-	—			RXPPS<4:0>				
E24h	RXPPS ⁽⁴⁾		_	_			RXPPS<4:0>				
E25h	CKPPS ⁽³⁾	_	_	_			CKPPS<4:0>				
E25h	CKPPS ⁽⁴⁾	_	_	_	CKPPS<4:0> SMT1SIGPPS<4:0>						
E26h	SMT1SIGPPS	_	_			S	MT1SIGPPS<4:	0>			
E27h	SMT1WINPPS	—	_			SI	MT1WINPPS<4:	0>			
E28h	CLCIN0PPS	—	—	—		(CLCIN0PPS<4:0	>			
E29h	CLCIN1PPS		_	_		(CLCIN1PPS<4:0	>			
E2Ah	CLCIN2PPS	_	_	_		(CLCIN2PPS<4:0	>			
E2Bh	CLCIN3PPS	—	_	—		(CLCIN3PPS<4:0	>			
E2Ch	SMT2SIGPPS	_	_	_		S	MT2SIGPPS<4:	0>			
E2Dh	SMT2WINPPS	—	_	_		SI	MT2WINPPS<4:	0>			
E2Eh	ATCC3PPS		_	—	ATCC3PPS<4:0>						
E2Fh to	_	Unimplemented	d								

to — E6Fh

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as Note 1: PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) **TABLE 3-14:**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Banks	s 29			· <u> </u>	·		<u> </u>	·	
E8Ch to E8Fh	—	Unimplemented	1						
E90h	RA0PPS	—	—	—			RA0PPS<4:0>		
E91h	RA1PPS						RA1PPS<4:0>		
E92h	RA2PPS	—					RA2PPS<4:0>		
E93h	—	Unimplemented	t						
E94h	RA4PPS		—	—			RA4PPS<4:0>		
E95h	RA5PPS	—	—	—			RA5PPS<4:0>		
E96h to E9Bh	_	Unimplemented							
E9Ch	RB4PPS ⁽⁴⁾		_	—			RB4PPS<4:0>		
E9Dh	RB5PPS ⁽⁴⁾	_	_				RB5PPS<4:0>		
E9Eh	RB6PPS ⁽⁴⁾	—	—	—			RB6PPS<4:0>		
E9Fh	RB7PPS ⁽⁴⁾	—	—	—			RB7PPS<4:0>		
EA0h	RC0PPS		_	-			RC0PPS<4:0>		
EA1h	RC1PPS	—	—	—			RC1PPS<4:0>		
EA2h	RC2PPS	—	—	—			RC2PPS<4:0>		
EA3h	RC3PPS	—	—	—			RC3PPS<4:0>		
EA4h	RC4PPS		<u> </u>				RC4PPS<4:0>		
EA5h	RC5PPS	—	—	—			RC5PPS<4:0>		
EA6h	RC6PPS ⁽⁴⁾		—				RC6PPS<4:0>		
EA7h	RC7PPS ⁽⁴⁾		—	—			RC7PPS<4:0>		
EA8h to EEFh		Unimplemented		de an condition	_ unimplement				

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as Note 1: PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED) **TABLE 3-14:**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0			
Banks	s 30											
F0Ch to F0Eh	_	Unimplemented										
F0Fh	CLCDATA	_		_	_	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT			
F10h	CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN		LC1MODE<2:0>				
F11h	CLC1POL	LC1POL				LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL			
F12h	CLC1SEL0		—			LC1D1	S<5:0>					
F13h	CLC1SEL1	—	—			LC1D2	2S<5:0>					
F14h	CLC1SEL2	—	—			LC1D3	LC1D3S<5:0>					
F15h	CLC1SEL3	_	—			LC1D4	S<5:0>					
F16h	CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N					
F17h	CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D2N LC1G2D1T LC1G				
F18h	CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N			
F19h	CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N			
F1Ah	CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN		LC2MODE<2:0>	,			
F1Bh	CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL			
F1Ch	CLC2SEL0	_	_			LC2D1	S<5:0>					
F1Dh	CLC2SEL1	—	—			LC2D2	2S<5:0>					
F1Eh	CLC2SEL2	—	—			LC2D3	S<5:0>					
F1Fh	CLC2SEL3					LC2D4	S<5:0>	<u> </u>				
F20h	CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N			
F21h	CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N			
F22h	CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N			
F23h	CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N			

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as Note 1:

PIC16F1615/9 only. Unimplemented, read as '1'. 2:

3:

PIC16(L)F1615 only. PIC16(L)F1619 only. 4:

					\	- /			
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Banks	s 30 (Continued)								
F24h	CLC3CON	LC3EN	_	LC3OUT	LC3INTP	LC3INTN		LC3MODE<2:0>	
F25h	CLC3POL	LC3POL	_	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL
F26h	CLC3SEL0	—				LC3D1	S<5:0>		
F27h	CLC3SEL1	—	_			LC3D2	S<5:0>		
F28h	CLC3SEL2	—	_			LC3D3	S<5:0>		
F29h	CLC3SEL3	—				LC3D4	S<5:0>		
F2Ah	CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N
F2Bh	CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N
F2Ch	CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N
F2Dh	CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N
F2Eh	CLC4CON	LC4EN	_	LC4OUT	LC4INTP	LC4INTN		LC4MODE<2:0>	•
F2Fh	CLC4POL	LC4POL	_	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL
F30h	CLC4SEL0	—	_			LC4D1	S<5:0>		
F31h	CLC4SEL1	—	_			LC4D2	S<5:0>		
F32h	CLC4SEL2	—	_			LC4D3	S<5:0>		
F33h	CLC4SEL3	—	_			LC4D4	S<5:0>		
F34h	CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N
F35h	CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N
F36h	CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N
F37h	CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N
F38h to	_	Unimplemented	í						

F6Fh

Legend: Note 1: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

PIC16(L)F1619 only. 4:

PIC16(L)F1615/9

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	l
Bank	31			1					1
F8Ch	_	Unimplemented	t						
— FE3h									
FE4h	STATUS_	_		_	_	_	Z_SHAD	DC_SHAD	C_
	SHAD								
FE5h	WREG_	Working Regist	er Shadow	•					
	SHAD								
FE6h	BSR_	—	—	—	Bank Select Re	gister Shadow			
	SHAD								
FE7h	PCLATH_	—	Program Counter	er Latch High R	egister Shadow				
	SHAD								
FE8h	FSR0L_	Indirect Data M	emory Address 0) Low Pointer SI	nadow				
	SHAD								
FE9h	FSR0H_	Indirect Data M	emory Address 0) High Pointer S	hadow				
	SHAD								
FEAh	FSR1L_	Indirect Data M	emory Address 1	Low Pointer SI	nadow				
	SHAD								
FEBh	FSR1H_	Indirect Data M	emory Address 1	High Pointer S	hadow				
	SHAD								
FECh	—	Unimplemented	Ł						
FEDh	STKPTR	—	—	—	Current Stack P	ointer			
FEEh	TOSL	Top-of-Stack Lo	ow byte						
FEFh	тоѕн		Top-of-Stack Hid	ah byte					

TARI E 3-14. SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

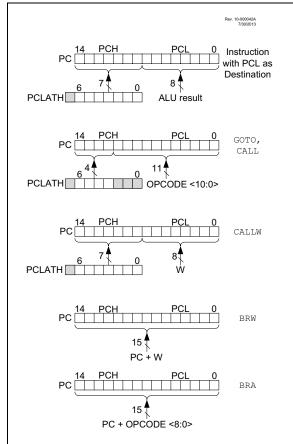
Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, Note1:PIC16F1615/9 only.2:Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

3.4 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 3-3 shows the five situations for the loading of the PC.

FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS



3.4.1 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.

3.4.2 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).

3.4.3 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.

3.4.4 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.

3.5 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figures 3-4 through 3-7). 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 Words). 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 1: 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.

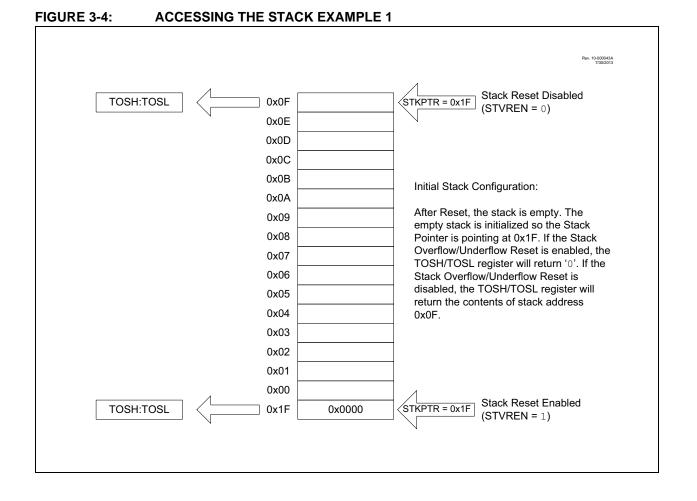
3.5.1 ACCESSING THE STACK

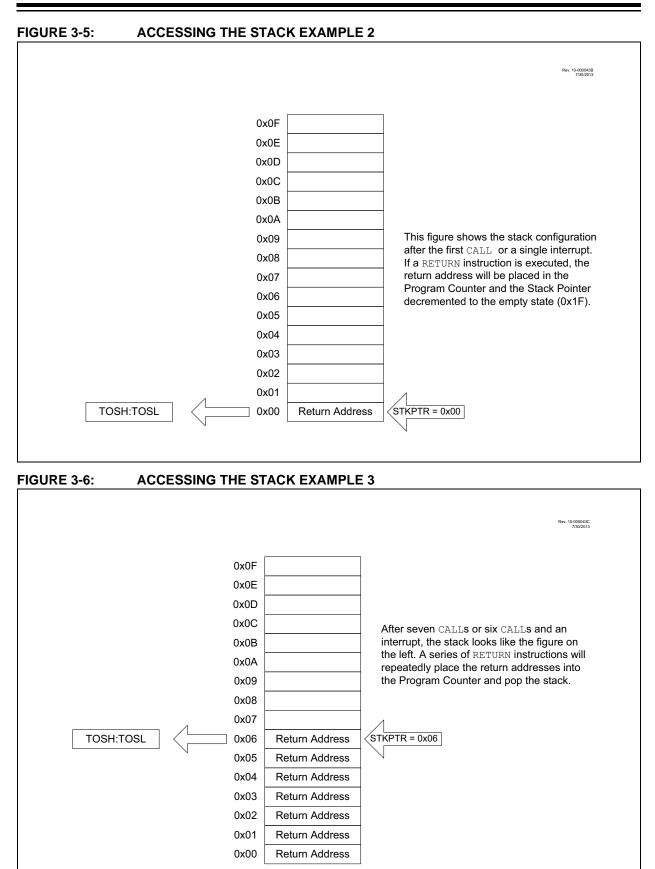
The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust 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.

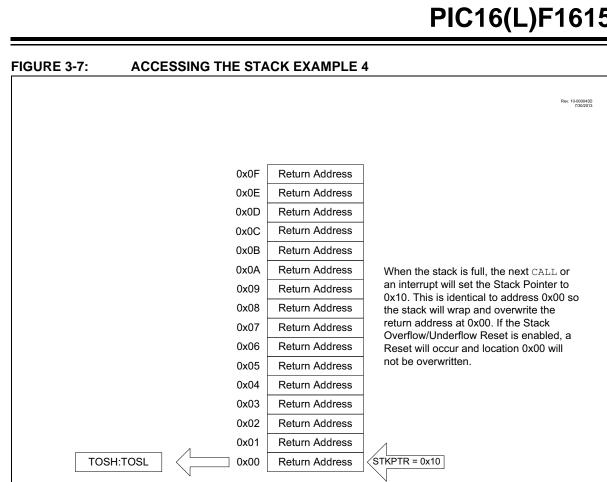
Note:	Care should be taken when modifying the
	STKPTR while interrupts are enabled.

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 3-4 through Figure 3-7 for examples of accessing the stack.







3.5.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words 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.

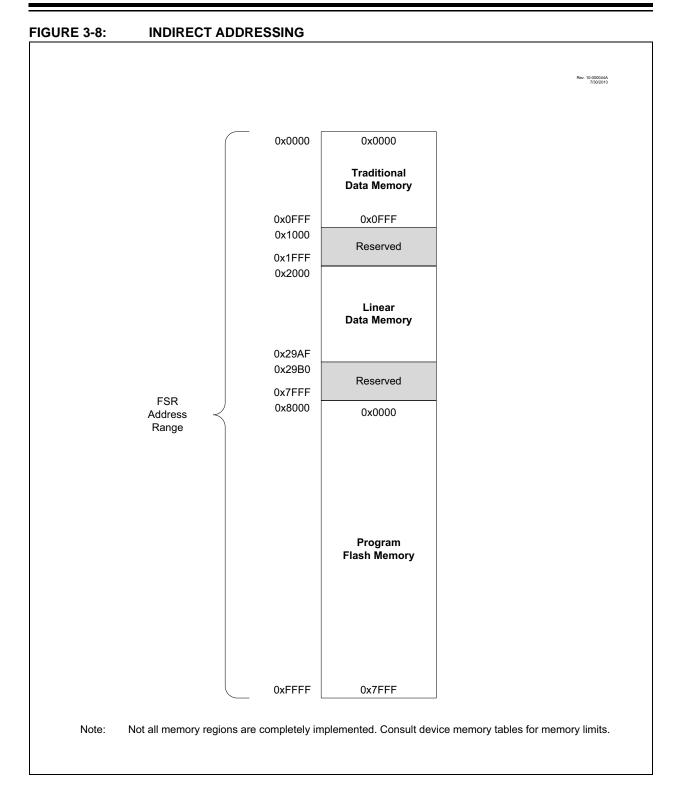
3.6 Indirect Addressing

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

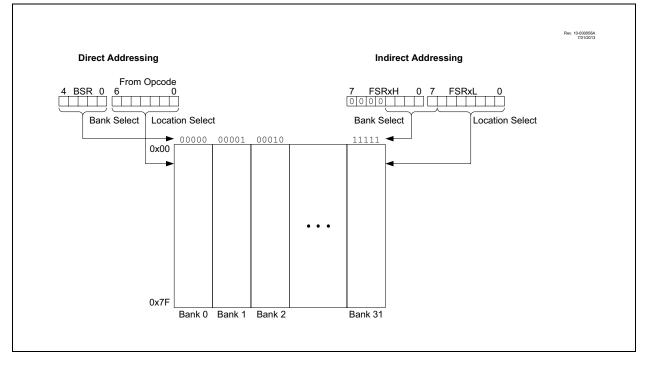
PIC16(L)F1615/9



3.6.1 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 3-9: TRADITIONAL DATA MEMORY MAP



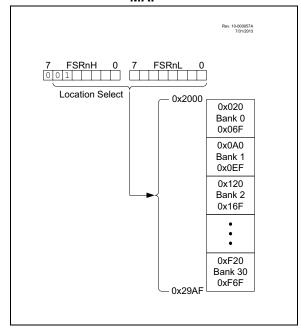
3.6.2 LINEAR DATA MEMORY

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.

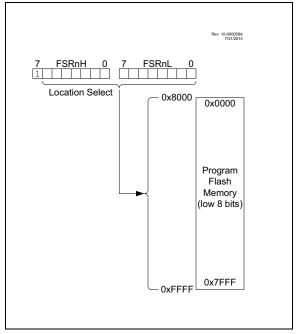
FIGURE 3-10: LINEAR DATA MEMORY MAP



3.6.3 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 3-11: PROGRAM FLASH MEMORY MAP



4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, 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, Configuration Word 2 at 8008h, and Configuration 3 at 8009h.

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'.						

4.2 Register Definitions: Configuration Words

		R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1		
		FCMEN	IESO	CLKOUTEN	BORE	N<1:0>(1)	_		
		bit 13		-			bit 8		
R/P-1	R/P-1	R/P-1	U-1	U-1	R/P-1	R/P-1	R/P-1		
CP(2)	MCLRE	PWRTE		_		FOSC<2:0>			
bit 7							bit 0		
Legend:									
R = Readab	le bit	P = Programm	able bit	U = Unimpleme	ented bit, rea	d as '1'			
'0' = Bit is cl	eared	'1' = Bit is set		-n = Value whe	n blank or af	ter Bulk Erase			
bit 13	1 = Fail-Safe	Clock Monitor E Clock Monitor i Clock Monitor i	s enabled						
bit 12	1 = Internal/E	al/External Switc External Switch (External Switch (Over mode is						
bit 11	1 = CLKOUT	CLKOUTEN: Clock Out Enable bit 1 = CLKOUT function is disabled. I/O function on the CLKOUT pin 0 = CLKOUT function is enabled on the CLKOUT pin							
bit 10-9	11 = BOR en 10 = BOR en	nabled during op ontrolled by SBC	eration and d	_{bits} (1) lisabled in Sleep e BORCON regis	ster				
bit 8	Unimplemer	nted: Read as '1	,						
bit 7	CP: Code Pr	\overline{CP} : Code Protection bit ⁽²⁾							
		memory code p							
bit 6	-	memory code p							
bit 0	If LVP bit = 1								
	This bit is	-							
	<u>If LVP bit = 0</u> 1 =MCLR		is MCLR: We	eak pull-up enable	ed.				
	0 =MCLR			it; MCLR internally		eak pull-up under	control of		
bit 5	PWRTE : Pow 1 = PWRT d 0 = PWRT e		able bit						
bit 4-3		nted: Read as '1	,						
	Jumplemen								

REGISTER 4-1: CONFIG1: CONFIGURATION WORD 1

REGISTER 4-1: CONFIG1: CONFIGURATION WORD 1 (CONTINUED)

- bit 2-0 FOSC<2:0>: Oscillator Selection bits
 - 111 =ECH: External clock, High-Power mode: on CLKIN pin
 - 110 =ECM: External clock, Medium-Power mode: on CLKIN pin
 - 101 =ECL: External clock, Low-Power mode: on CLKIN pin
 - 100 =INTOSC oscillator: I/O function on CLKIN pin
 - 011 =Reserved
 - 010 =HS: HS oscillator, high-speed crystal/resonator connected between OSC1 and OSC2 pins
 - 001 =Reserved
 - 000 =Reserved
- Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
 - 2: Once enabled, code-protect can only be disabled by bulk erasing the device.

		R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	
		LVP ⁽¹⁾	DEBUG ⁽³⁾	LPBOR	BORV ⁽²⁾	STVREN	PLLEN	
		bit 13					bit 8	
R/P-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1	R/P-1	
ZCD	0-1	0-1	0-1	0-1				
				—	PPS1WAY	WRT		
bit 7							bit	
Legend:								
R = Read	able bit	P = Program	nable bit	U = Unimplen	nented bit, read	l as '1'		
'0' = Bit is	cleared	'1' = Bit is set		-n = Value wh	en blank or afte	er Bulk Erase		
bit 13	1 = Low-volta	ge programmir	ming Enable bit ng enabled nust be used fo		2 C			
bit 12	DEBUG: In-C 1 = In-Circuit	ircuit Debugge Debugger disa	er Mode bit ⁽³⁾ ibled, ICSPCLK	and ICSPDA	Γ are general p are dedicated			
bit 11	1 = Low-Powe		nable bit Reset is disable Reset is enabled	-				
bit 10	1 = Brown-ou	BORV: Brown-Out Reset Voltage Selection bit ⁽²⁾ 1 = Brown-out Reset voltage (VBOR), low trip point selected 0 = Brown-out Reset voltage (VBOR), high trip point selected						
bit 9	STVREN: Sta 1 = Stack Ove	 STVREN: Stack Overflow/Underflow Reset Enable bit 1 = Stack Overflow or Underflow will cause a Reset 0 = Stack Overflow or Underflow will not cause a Reset 						
bit 8	PLLEN: PLL 1 = 4xPLL en 0 = 4xPLL dis	abled						
bit 7	ZCD : ZCD Dis 1 = ZCD disal 0 = ZCD alwa	bled. ZCD can	be enabled by	setting the ZC	D1EN bit of ZC	D1CON		
bit 6-3	Unimplemen	ted: Read as '	1'					
bit 2	1 = The PPSL is set, all	OCK bit can o	s to PPS registe	after an unloc ers are preven	king sequence ted provided an unk			
bit 1-0	<u>8 kW Flash m</u> 11 = OF1 10 = BO0 01 = HAD	emory (PIC16 F - Write prot DT - 000h to 0 LF - 000h to 0F	ection off 1FFh write-prot FFFh write-prote	ected, 200h to ected, 1000h to	1FFFh may be o 1FFFh may be resses may be	e modified by Pl	MCON contro	
Note 1: 2: 3:	The LVP bit canno See VBOR parame The DEBUG bit ir debuggers and pr	eter for specific Configuration	c trip point volta Words is mana	iges. aged automatio	cally by device	development to		

REGISTER 4-2: CONFIG2: CONFIGURATION WORD 2

© 2014-2017 Microchip Technology Inc.

REGISTER 4-3: CONFIG3: CONFIGURATION WORD 3

		R/P-0	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1
			WDTCCS<2:0>	>	\	WDTCWS<2:0	>
		bit 13					bit 8
U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
_	WDT	E<1:0>			WDTCPS<4:0>	>	
bit 7							bit 0

Legend:	
Leyenu.	

bit 6-5

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'
'0' = Bit is cleared	'1' = Bit is set	-n = Value when blank or after Bulk Erase

WDTCCS<2:0>: WDT Configuration Clock Select bits bit 13-11

- 111 =Software Control; WDT clock selected by CS<2:0> 110 =Reserved
- 010 =Reserved
- 001 =WDT reference clock is MFINTOSC, 31.25 kHz (default value)
- 000 =WDT reference clock is LFINTOSC, 31.00 kHz output

WDTCWS<2:0>: WDT Configuration Window Select bits. bit 10-8

WDTCWS		WINDOW at P	OR	Software	Keyed	
<2:0>	Value	Window delay Percent of time	Window opening Percent of time	control of WINDOW?	access required?	
111	111	n/a	100	Yes	No	Default fuse = 111
110	111	n/a	100			
101	101	25	75			
100	100	37.5	62.5			
011	011	50	50	No	Yes	
010	010	62.5	37.5			
001	001	75	25			
000	000	87.5	12.5 ⁽¹⁾			

bit 7 Unimplemented: Read as '1'

WDTE<1:0>: Watchdog Timer Enable bits

11 =WDT enabled in all modes, the SEN bit in the WDTCON0 register is ignored

10 =WDT enabled while running and disabled in Sleep

01 =WDT controlled by the SEN bit in the WDTCON0 register

00 = WDT disabled

REGISTER 4-3: CONFIG3: CONFIGURATION WORD 3 (CONTINUED)

bit 4-0 WDTCPS<4:0>: WDT Configuration Period Select bits

		WDTPS at	t POR		Software]
WDTCPS <4:0>	Value	Divider Ra	atio	Typical time out (FIN = 31 kHz)	control of WDTPS	
11111	01011	1:65536	2 ¹⁶	2 s	Yes	Default fuse = 11111
10011 11110	10011 11110	1:32	2 ⁵	1 ms	No	
10010	10010	1:8388608	2 ²³	256 s		
10001	10001	1:4194304	2 ²²	128 s		
10000	10000	1:2097152	2 ²¹	64 s		
01111	01111	1:1048576	2 ²⁰	32 s		
01110	01110	1:524299	2 ¹⁹	16 s		
01101	01101	1:262144	2 ¹⁸	8 s		
01100	01100	1:131072	2 ¹⁷	4 s		
01011	01011	1:65536	2 ¹⁶	2 s		
01010	01010	1:32768	2 ¹⁵	1 s		
01001	01001	1:16384	2 ¹⁴	512 ms	No	
01000	01000	1:8192	2 ¹³	256 ms		
00111	00111	1:4096	2 ¹²	128 ms		
00110	00110	1:2048	2 ¹¹	64 ms		
00101	00101	1:1024	2 ¹⁰	32 ms		
00100	00100	1:512	2 ⁹	16 ms		
00011	00011	1:256	2 ⁸	8 ms		
00010	00010	1:128	2 ⁷	4 ms		
00001	00001	1:64	2 ⁶	2 ms		
00000	00000	1:32	2 ⁵	1 ms		J

Note 1: A window delay of 12.5% is only available in Software Control mode via the WDTCON1 register.

4.3 Code Protection

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

4.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the \overline{CP} bit in Configuration Words. 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 4.4 "Write Protection" for more information.

4.4 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 Words define the size of the program memory block that is protected.

4.5 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 "*PIC12(L)F1612/16(L)F161X Memory Programming Specification*" (DS40001720).

4.6 Device ID and Revision ID

The 14-bit Device ID word is located at 8006h and the 14-bit Revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. 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.

4.7 Register Definitions: Device ID

REGISTER 4-4: DEVID: DEVICE ID REGISTER

		R	R	R	R	R	R
				DEV<	13:8>		
		bit 13					bit 8
R	R	R	R	R	R	R	R
			DEV	<7:0>			
bit 7							bit 0

Legend:

R = Readable bit

'1' = Bit is set

bit 13-0 **DEV<13:0>:** Device ID bits

Device	DEVID<13:0> Values							
PIC16F1615	11 0000 0111 1100 (307Ch)							
PIC16LF1615	11 0000 0111 1110 (307Eh)							
PIC16F1619	11 0000 0111 1101 (307Dh)							
PIC16LF1619	11 0000 0111 1111 (307Fh)							

'0' = Bit is cleared

REGISTER 4-5: REVID: REVISION ID REGISTER

		R	R	R	R	R	R
				REV<	13:8>		
		bit 13					bit 8
R	R	R	R	R	R	R	R
			REV	<7:0>			
bit 7							bit 0

Legend:

R = Readable bit '1' = Bit is set

'0' = Bit is cleared

bit 13-0 **REV<13:0>:** Revision ID bits

5.0 OSCILLATOR MODULE

5.1 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 5-1 illustrates a block diagram of the oscillator module.

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

- Selectable system clock source between external or internal sources via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (HS, ECH, ECM, or ECL modes) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability
 of crystal oscillator sources

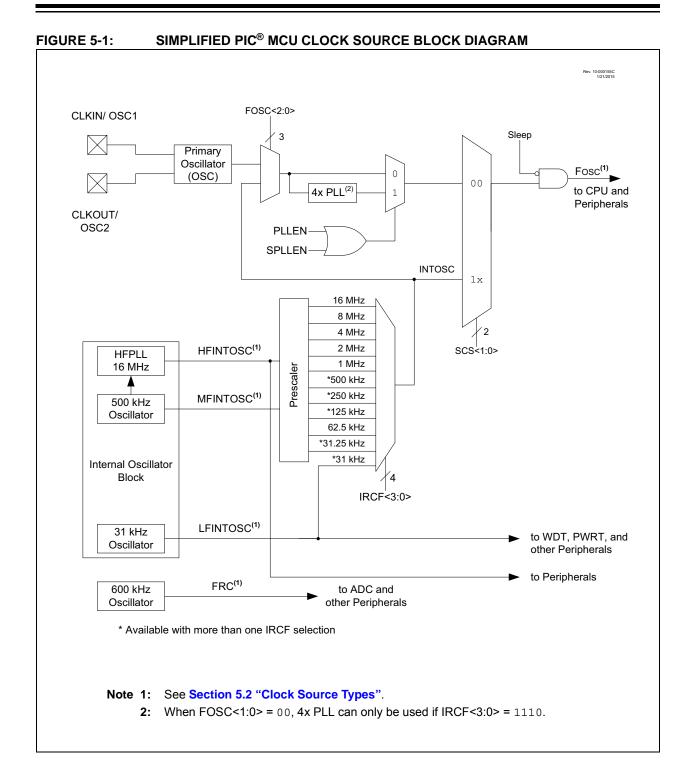
The oscillator module can be configured in one of the following clock modes.

- 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)
- 3. ECH External Clock High-Power mode (4 MHz to 32 MHz)
- 4. INTOSC Internal oscillator (31 kHz to 32 MHz).
- 5. HS High Gain Crystal or Ceramic Resonator mode (4 MHz to 10 MHz)

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

The ECH, ECM, and ECL Clock modes rely on an external logic level signal as the device clock source. The HS Clock mode requires an external crystal or resonator to be connected to the device.

The INTOSC internal oscillator block produces low, medium, and high-frequency clock sources, designated LFINTOSC, MFINTOSC and HFINTOSC. (see Internal Oscillator Block, Figure 5-1). A wide selection of device clock frequencies may be derived from these three clock sources.



5.2 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. Examples are: oscillator modules (ECH, ECM, ECL modes) and quartz crystal resonators or ceramic resonators (HS mode).

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators and a dedicated Phase Lock Loop (HFPLL) that are used to generate three internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC), 500 kHz (MFINTOSC) 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 **Section5.3 "Clock Switching**" for additional information.

5.2.1 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 Words 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:
 - An external clock source determined by the value of the FOSC bits.

See **Section5.3 "Clock Switching**" for more information.

5.2.1.1 EC Mode

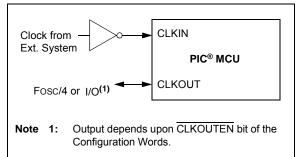
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. CLKOUT is available for general purpose I/O or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through the Fosc bits in the Configuration Words:

- ECH High power, 4-20 MHz
- ECM Medium power, 0.5-4 MHz
- ECL Low power, 0-0.5 MHz

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 limiting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.



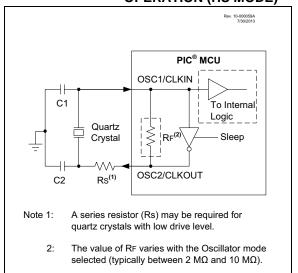


5.2.1.2 HS Mode

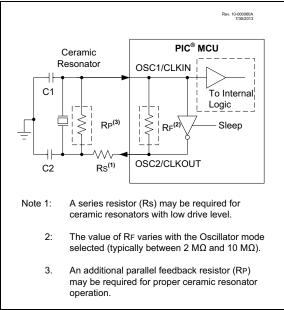
The HS mode supports the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2. Figure 5-3 and Figure 5-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

FIGURE 5-3:

QUARTZ CRYSTAL OPERATION (HS MODE)







5.2.1.3 Oscillator Start-up Timer (OST)

If the oscillator module is configured for HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Poweron Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see Section5.4 "Two-Speed Clock Start-up Mode").

5.2.2 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 Words 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 Section5.3 "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 OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators and a dedicated Phase Lock Loop, HFPLL that can produce one of three internal system clock sources.

- The HFINTOSC (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The HFINTOSC source is generated from the 500 kHz MFINTOSC source and the dedicated Phase Lock Loop, HFPLL. The frequency of the HFINTOSC can be useradjusted via software using the OSCTUNE register (Register 5-3).
- The MFINTOSC (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
- 3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

5.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of multiple frequencies derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See Section5.2.2.8 "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> = 00, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'.

A fast start-up oscillator allows internal circuits to power up and stabilize before switching to HFINTOSC.

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

The High-Frequency Internal Oscillator Status Locked bit (HFIOFL) of the OSCSTAT register indicates when the HFINTOSC is running within 2% of its final value.

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

5.2.2.2 MFINTOSC

The Medium-Frequency Internal Oscillator (MFINTOSC) is a factory calibrated 500 kHz internal clock source. The frequency of the MFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the MFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of nine frequencies derived from the MFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See Section5.2.2.8 "Internal Oscillator Clock Switch Timing" for more information.

The MFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<1:0> = 00, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

The Medium-Frequency Internal Oscillator Ready bit (MFIOFR) of the OSCSTAT register indicates when the MFINTOSC is running.

5.2.2.3 Internal Oscillator Frequency Adjustment

The 500 kHz internal oscillator is factory calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 5-3). Since the HFINTOSC and MFINTOSC clock sources are derived from the 500 kHz internal oscillator a change in the OSCTUNE register value will apply to both.

The default value of the OSCTUNE register is '0'. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), and peripherals, are *not* affected by the change in frequency.

5.2.2.4 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 5-1). Select 31 kHz, via software, using the IRCF<3:0> bits of the OSCCON register. See Section5.2.2.8 "Internal Oscillator Clock Switch Timing" for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

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> = 00, 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)
- Fail-Safe Clock Monitor (FSCM)

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

5.2.2.5 FRC

The FRC clock is an uncalibrated, nominal 600 kHz peripheral clock source.

The FRC is automatically turned on by the peripherals requesting the FRC clock.

The FRC clock will continue to run during Sleep.

5.2.2.6 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 postscaler outputs of the 16 MHz HFINTOSC, 500 kHz MFINTOSC, and 31 kHz LFINTOSC output connect to a multiplexer (see Figure 5-1). 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.

5.2.2.7 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. Either the 8 or 16 MHz internal oscillator settings can be used, with the 16 MHz being divided by two before being input into the PLL. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Words must be set to use the INTOSC source as the device system clock (FOSC<1:0> = 00).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<1:0> in Configuration Words (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to either the 16 MHz (IRCF<3:0> = 1111) or the 8 MHz HFINTOSC (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4x PLL, or the PLLEN bit of the Configuration Words must be programmed to a '1'.

Note: When using the PLLEN bit of the Configuration Words, the 4x PLL cannot be disabled by software and the 8/16 MHz HFINTOSC option will no longer be available.

The 4x PLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to '1x'. The SCS bits must be set to '00' to use the 4x PLL with the internal oscillator.

5.2.2.8 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-3). 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, MFINTOSC 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 5-3 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 5-1.

Start-up delay specifications are located in the oscillator tables of **Section35.0 "Electrical Specifications"**.

FIGURE 5-5:	INTERNAL OSCILLATOR SWITCH TIMING
HFINTOSC/→ MFINTOSC	LFINTOSC (WDT disabled)
HFINTOSC/ MFINTOSC	Start-up Time 2-cycle Sync Running
LFINTOSC	
IRCF <3:0>	$\neq 0$ $\chi = 0$
System Clock	
HFINTOSC/→ MFINTOSC	LFINTOSC (WDT enabled)
HFINTOSC/ MFINTOSC	
LFINTOSC	
IRCF <3:0>	$\neq 0$ $\chi = 0$
System Clock	
LFINTOSC →	HFINTOSC/MFINTOSC
LFINTOSC	LFINTOSC turns off unless WDT is enabled
	Start-up Time 2-cycle Sync Running
HFINTOSC/ MFINTOSC	
IRCF <3:0>	= 0 X ≠ 0
System Clock	

5.3 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 Words
- Internal Oscillator Block (INTOSC)

5.3.1 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 Words.
- 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 5-1.

5.3.2 CLOCK SWITCHING BEFORE SLEEP

When clock switching from an old clock to a new clock is requested just prior to entering Sleep mode, it is necessary to confirm that the switch is complete before the SLEEP instruction is executed. Failure to do so may result in an incomplete switch and consequential loss of the system clock altogether. Clock switching is confirmed by monitoring the clock Status bits in the OSCSTAT register. Switch confirmation can be accomplished by sensing that the ready bit for the new clock is set or the ready bit for the old clock is cleared. For example, when switching between the internal oscillator with the PLL and the internal oscillator without the PLL, monitor the PLLR bit. When PLLR is set, the switch to 32 MHz operation is complete. Conversely, when PLLR is cleared, the switch from 32 MHz operation to the selected internal clock is complete.

Switch From	Switch To	Frequency	Oscillator Delay
Sleep	LFINTOSC ⁽¹⁾ MFINTOSC ⁽¹⁾ HFINTOSC ⁽¹⁾	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	Oscillator Warm-up Delay (Twarm) ⁽²⁾
Sleep/POR	EC ⁽¹⁾	DC – 32 MHz	2 cycles
LFINTOSC	EC ⁽¹⁾	DC – 32 MHz	1 cycle of each
Any clock source	MFINTOSC ⁽¹⁾ HFINTOSC ⁽¹⁾	31.25 kHz-500 kHz 31.25 kHz-16 MHz	2 μs (approx.)
Any clock source	LFINTOSC ⁽¹⁾	31 kHz	1 cycle of each
PLL inactive	PLL active	16-32 MHz	2 ms (approx.)

TABLE 5-1: OSCILLATOR SWITCHING DELAYS

Note 1: PLL inactive.

2: See Section 35.0, Electrical Specifications.

5.4 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device. This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC internal oscillator block as the clock source and go back to Sleep without waiting for the external oscillator to become stable.

Two-Speed Start-up provides benefits when the oscillator module is configured for HS modes. The Oscillator Startup Timer (OST) is enabled for these modes and must count 1024 oscillations before the oscillator can be used as the system clock source.

If the oscillator module is configured for any mode other than HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

If the OST count reaches 1024 before the device enters Sleep mode, the OSTS bit of the OSCSTAT register is set and program execution switches to the external oscillator. However, the system may never operate from the external oscillator if the time spent awake is very short.

Note: Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCSTAT register to remain clear.

5.4.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO (of the Configuration Words) = 1; Internal/ External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 00.
- FOSC<2:0> bits in the Configuration Words configured for HS mode.

Two-Speed Start-up mode is entered after:

- Power-On Reset (POR) and, if enabled, after Power-up Timer (PWRT) has expired, or
- Wake-up from Sleep.

Note: When FSCM is enabled, Two-Speed Startup will automatically be enabled.

5.4.2 TWO-SPEED START-UP SEQUENCE

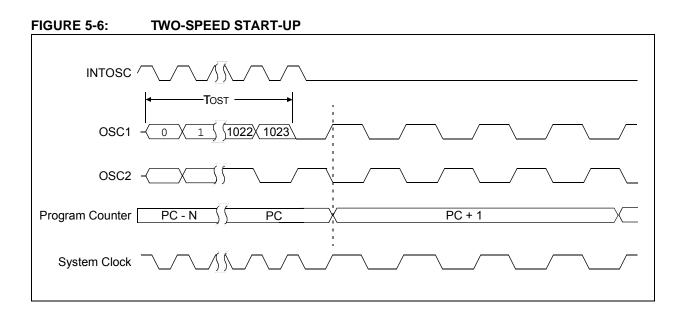
- 1. Wake-up from Power-on Reset or Sleep.
- Instructions begin execution by the internal oscillator at the frequency set in the IRCF<3:0> bits of the OSCCON register.
- 3. OST enabled to count 1024 clock cycles.
- 4. OST timed out, wait for falling edge of the internal oscillator.
- 5. OSTS is set.
- 6. System clock held low until the next falling edge of new clock (HS mode).
- 7. System clock is switched to external clock source.

5.4.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCSTAT register will confirm if the CPU is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Words, or the internal oscillator. See Table 5-2.

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

TABLE 5-2: OSCILLATOR SWITCHING DELAYS



5.5 Fail-Safe Clock Monitor

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

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

Clock Monitor Latch External sŧ Q Clock г LFINTOSC ÷ 64 Q Oscillator 31 kHz 488 Hz (~32 µs) (~2 ms) Sample Clock Clock Failure Detected

FIGURE 5-7: FSCM BLOCK DIAGRAM

5.5.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by monitoring falling clock edges and using LFINTOSC as a time base. See Figure 5-7. Detection of a failed oscillator will take 32 to 96 cycles of the LFINTOSC. Figure 5-10 shows a timing diagram of the FSCM module.

5.5.2 FAIL-SAFE OPERATION

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

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

The system clock will continue to be sourced from the internal clock source until the Fail-Safe condition has been cleared, see Section5.5.3 "Fail-Safe Condition Clearing".

5.5.3 FAIL-SAFE CONDITION CLEARING

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

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

5.5.3.1 External Oscillator with SCS<1:0> = 00

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

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

SCS<1:0> = 1x:

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

OSFIF = 0:

Clear the OSFIF bit in the PIR2 register.

SCS<1:0> = 00:

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

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

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

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

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

SCS<1:0> = 1x:

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

OSFIF = 0:

Clear the OSFIF bit in the PIR2 register.

SCS<1:0> = 00:

Change the SCS bits in the OSCCON register to select the FOSC Configuration Word clock selection. Since the OST is not applicable with external clocks, the clock module will immediately switch to the external clock, and the Fail-Safe condition will be cleared.

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

5.5.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect external oscillator or external clock failures.

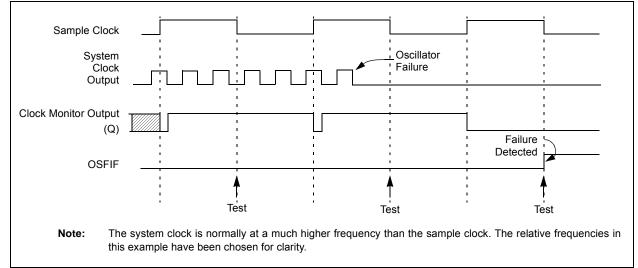
When FSCM is used with an external oscillator, the Oscillator Start-up Timer (OST) count must expire before the FSCM becomes active. The OST is used after waking up from Sleep and after any type of Reset.

FIGURE 5-8: FSCM TIMING DIAGRAM

When the FSCM is used with external clocks, the OST is not used and the FSCM will be active as soon as the Reset or wake-up has completed.

When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

Note: Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep).



5.6 Register Definitions: Oscillator Control

R/W-0/0	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
SPLLEN		IRCF	<3:0>		_	SCS	<1:0>
bit 7	•						bit (
1							
Legend: R = Readable	a hit	W = Writable	hit	U = Unimpler	onted hit rea	ad as '0'	
u = Bit is unc		x = Bit is unkr		-		OR/Value at all	other Decete
(1) = Bit is unc(1) = Bit is set	•	'0' = Bit is cle					
1 - Dit 13 361	L		areu				
bit 7	If PLLEN in SPLLEN bit	Configuration W	′ <u>ords = 1:</u> LL is always e	enabled (subject	to oscillator r	equirements)	
bit 6-3	1111 =16 M 1110 =8 MH 1101 =4 MH 1000 =2 MH 1011 =1 MH 1010 =500 H 1001 =250 H 1000 =125 H	Iz HF Iz HF Iz HF Iz HF (Hz HF ⁽¹⁾ (Hz HF ⁽¹⁾ (Hz HF ⁽¹⁾) (Hz MF (default (Hz MF (Hz MF kHz MF kHz MF 5 kHz MF		Select bits			
bit 2 bit 1-0	SCS<1:0>: \$	nted: Read as ' System Clock S I oscillator block	elect bits	tor block)			

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0/q	U-0	R-0/q	R-0/q	R-q/q	R-0/q	R-0/q
—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, read	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 o	other Resets
'1' = Bit is set		'0' = Bit is clea	ared	q = Condition	al		
bit 7	•	ted: Read as '	0'				
bit 6	PLLR: 4x PL 1 = 4x PLL i	•					
	1 = 4x PLL i 0 = 4x PLL i						
bit 5		ator Start-Up Ti	mer Status bit				
	1 = Running	from the clock of	defined by the	FOSC<2:0> bi	ts of the Config	uration Words	
	0 = Running	from an interna	l oscillator (FC	SC<2:0> = 10	0)		
bit 4	-	h-Frequency Ir	ternal Oscillate	or Ready bit			
	1 = HFINTO						
		SC is not ready					
bit 3	•	h-Frequency In SC is at least 2		or Locked bit			
		SC is at least 2 SC is not 2% a					
bit 2	MFIOFR: Me	dium-Frequenc	y Internal Osc	illator Ready b	it		
	1 = MFINTO	•		2			
	0 = MFINTO	SC is not ready	/				
bit 1	LFIOFR: Low	v-Frequency Int	ernal Oscillato	r Ready bit			
	1 = LFINTOS	,					
		SC is not ready		.			
bit 0	•	h-Frequency In	ternal Oscillato	or Stable bit			
	1 = HFINTO	SC is stable SC is not stable	`				
			5				

REGISTER 5-2: OSCSTAT: OSCILLATOR STATUS REGISTER

U-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				
	_			TUN	<5:0>						
bit 7							bit 0				
Legend:											
R = Readal	ole bit	W = Writable bitU = Unimplemented bit, read as '0'ed $x = Bit$ is unknown-n/n = Value at POR and BOR/Value at all other									
u = Bit is ur	= Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all o										
'1' = Bit is s	et	'0' = Bit is clea	ared								
bit 7-6	Unimpleme	ented: Read as '	0'								
bit 5-0	TUN<5:0>:	Frequency Tunir	ng bits								
	100000 = N	Ainimum frequen	су								
	•										
	•										
	111111 =	• 111111 =									
		000000 = Oscillator module is running at the factory-calibrated frequency.									
	000001 =										
	•										
	•										
	• 011110 =										
		Maximum frequer									

REGISTER 5-3: OSCTUNE: OSCILLATOR TUNING REGISTER

TABLE 5-3: 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	SPLLEN		IRCF<3:0>				SCS<1:0>		81
OSCSTAT	_	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	82
OSCTUNE	_	_		TUN<5:0>					

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

TABLE 5-4: 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
CONFIG1	13:8	_		FCMEN	IESO	CLKOUTEN	BORE	BOREN<1:0>		69
CONFIGI	7:0	CP	MCLRE	PWRTE	_		F	OSC<2:0>	•	09

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

6.0 RESETS

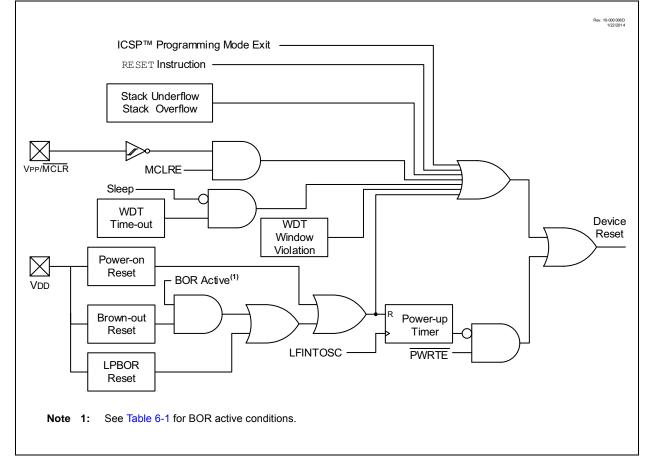
There are multiple ways to reset this device:

- 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.

A simplified block diagram of the On-chip Reset Circuit is shown in Figure 6-1.





6.1 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.

6.1.1 POWER-UP TIMER (PWRT)

The Power-up Timer provides a nominal 64 ms timeout 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 Words.

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).

6.2 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 Words. 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 6-1 for more information.

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

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 6-2 for more information.

BOREN<1:0>	SBOREN	Device Mode	BOR Mode	Instruction Execution upon: Release of POR or Wake-up from Sleep						
11	х	Х	Active	Waits for BOR ready ⁽¹⁾ (BORRDY = 1)						
1.0	37	Awake	Active	Waits for BOR ready						
10	Х	Sleep	Disabled	(BORRDY = 1)						
01	1	х	Active	Waits for BOR ready ⁽¹⁾ (BORRDY = 1)						
	0	Х	Disabled	Begins immediately						
00	Х	х	Disabled	(BORRDY = x)						

TABLE 6-1: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.

6.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Words are programmed 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.

6.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Words are programmed 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.

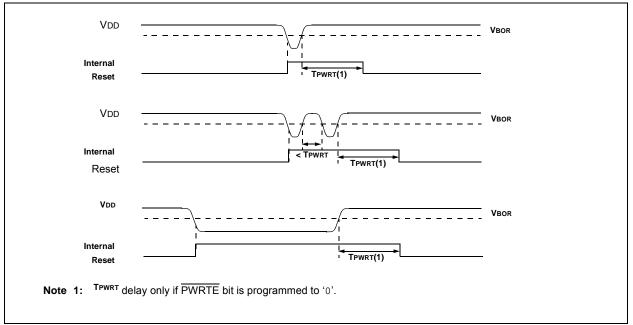
6.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Words are programmed 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.





6.3 Register Definitions: BOR Control

REGISTER 6-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	R/W-0/u U-0		u U-0 U-0 U-0		U-0	U-0 U-0			
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 <u>If BOREN <1:0> in Configuration Words = 01</u> : 1 = BOR Enabled 0 = BOR Disabled <u>If BOREN <1:0> in Configuration Words ≠ 01</u> : SBOREN is read/write, but has no effect on the BOR
bit 6	BORFS: Brown-Out Reset Fast Start bit ⁽¹⁾ If BOREN <1:0> = 10 (Disabled in Sleep) or BOREN<1:0> = 01 (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 If BOREN<1:0> = 11 (Always on) or BOREN<1:0> = 00 (Always off) BORFS is Read/Write, but has no effect.
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
Note 1. PC	DEN/21:05 bits are located in Configuration Words

Note 1: BOREN<1:0> bits are located in Configuration Words.

6.4 Low-Power Brown-Out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) operates like the BOR to detect low voltage conditions on the 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 BOR bit in PCON is used for both BOR and the LPBOR. Refer to Register 6-2.

The LPBOR voltage threshold (VLPBOR) has a wider tolerance than the BOR (VBOR), but requires much less current (LPBOR current) to operate. The LPBOR is intended for use when the BOR is configured as disabled (BOREN = 00) or disabled in Sleep mode (BOREN = 10).

Refer to Figure 6-1 to see how the LPBOR interacts with other modules.

6.4.1 ENABLING LPBOR

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

6.5 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 Words and the LVP bit of Configuration Words (Table 6-2).

TABLE 6-2: MCLR CONFIGURATION

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

6.5.1 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.

6.5.2 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 **Section12.1 "PORTA Registers"** for more information.

6.6 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period and the window is open. The TO and PD bits in the STATUS register are changed to indicate a WDT Reset caused by the timer overflowing, and WDTWV bit in the PCON register is changed to indicate a WDT Reset caused by a window violation. See Section9.0 "Windowed Watchdog Timer (WDT)" for more information.

6.7 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 6-4 for default conditions after a RESET instruction has occurred.

6.8 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 Words. See <u>Section3.5.2 "Overflow/Underflow</u> **Reset**" for more information.

6.9 Programming Mode Exit

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

6.10 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 Words.

6.11 Start-up Sequence

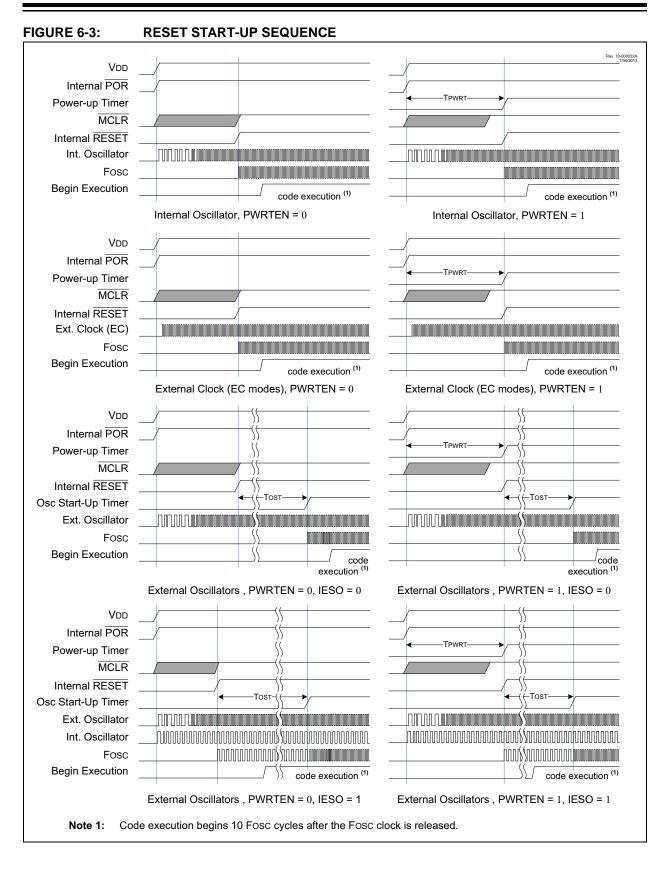
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

- 1. Power-up Timer runs to completion (if enabled).
- 2. MCLR must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section5.0** "Oscillator Module" for more information.

The Power-up Timer runs independently of MCLR Reset. If MCLR is kept low long enough, the Power-up Timer will expire. Upon bringing MCLR high, the device will begin execution after 10 FOSC cycles (see Figure 6-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

PIC16(L)F1615/9



6.12 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 6-3 and Table 6-4 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	х	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
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 6-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

TABLE 6-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	1 1000	0011 110x
MCLR Reset during normal operation	0000h	u uuuu	uuuu Ouuu
MCLR Reset during Sleep	0000h	1 Ouuu	uuuu 0uuu
WDT Reset	0000h	0 uuuu	uuu0 uuuu
WDT Wake-up from Sleep	PC + 1	0 Ouuu	uuuu uuuu
Brown-out Reset	0000h	1 luuu	00uu 11u0
Interrupt Wake-up from Sleep	PC + 1 ⁽¹⁾	1 Ouuu	uuuu uuuu
RESET Instruction Executed	0000h	u uuuu	uuuu u0uu
Stack Overflow Reset (STVREN = 1)	0000h	u uuuu	luuu uuuu
Stack Underflow Reset (STVREN = 1)	0000h	u uuuu	uluu uuuu
WDT Window Violation	0000h	1 uuuu	uu0u uuuu

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

Note 1: When the wake-up is due to an interrupt and the Global Interrupt 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.

6.13 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)

The PCON register bits are shown in Register 6-2.

6.14 Register Definitions: Power Control

REGISTER 6-2: PCON: POWER CONTROL REGISTER

R/W/HS-0/q	R/W/HS-0/q	R/W/HC-1/q	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	WDTWV	RWDT RMCLF		RI	POR	BOR
bit 7	•	•					bit 0

Legend:		
HC = Bit is cleared by hard	dware	HS = Bit is set by hardware
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	STKOVF : Stack Overflow Flag bit 1 = A Stack Overflow occurred 0 = A Stack Overflow has not occurred or cleared by firmware
bit 6	 STKUNF: Stack Underflow Flag bit 1 = A Stack Underflow occurred 0 = A Stack Underflow has not occurred or cleared by firmware
bit 5	 WDTWV: WDT Window Violation Flag bit 1 = A WDT Window Violation Reset has not occurred or set by firmware 0 = A WDT Window Violation Reset has occurred (a CLRWDT instruction was executed either without arming the window or outside the window (cleared by hardware)
bit 4	RWDT : Watchdog Timer Reset Flag bit 1 = A Watchdog Timer Reset has not occurred or set by firmware 0 = A Watchdog Timer Reset has occurred (cleared by hardware)
bit 3	RMCLR : MCLR Reset Flag bit 1 = A MCLR Reset has not occurred or set by firmware 0 = A MCLR Reset has occurred (cleared by hardware)
bit 2	RI: RESET Instruction Flag bit 1 = A RESET instruction has not been executed or set by firmware 0 = A RESET instruction has been executed (cleared by hardware)
bit 1	POR: Power-On Reset Status bit 1 = No Power-on Reset occurred 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
bit 0	BOR: Brown-Out Reset Status bit 1 = No Brown-out Reset occurred 0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

TABLE 6-5: S	JMMARY OF REGISTERS ASSOCIATED WITH RESETS
--------------	--

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	BORFS	_		—	_	_	BORRDY	86
PCON	STKOVF	STKUNF	WDTWV	RWDT	RMCLR	RI	POR	BOR	90
STATUS	_	_	_	TO	PD	Z	DC	С	25
WDTCON0	_	_		1	SEN	116			

Legend: — = unimplemented bit, reads as '0'. <u>Shaded</u> cells are not used by Resets.

Note 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

TABLE 6-6: SUMMARY OF CONFIGURATION WORD WITH RESETS

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			FCMEN	IESO	CLKOUTEN	BOREI	N<1:0>		60
CONFIGT	7:0	CP	MCLRE	PWRTE	_	_	FOSC<2:0>			69
	13:8	_	_	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	<u></u>
CONFIG2	7:0	ZCD	_		_	_	PPS1WAY	WRT	<1:0>	68
	13:8	_	_		WDTCCS<	<2:0>	W	DTCWS<2:0)>	<u> </u>
CONFIG3	7:0	_	WDT	E<1:0>		WD	TCPS<4:0>			69

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

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.

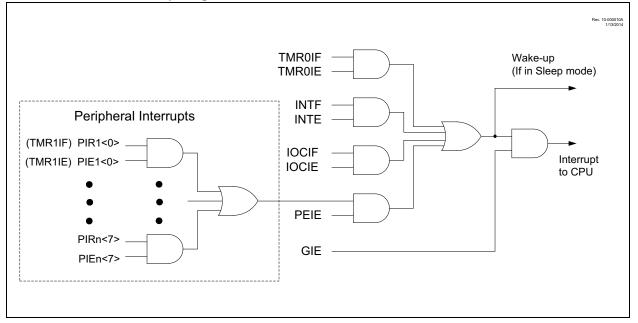
This chapter contains the following information for Interrupts:

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

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

A block diagram of the interrupt logic is shown in Figure 7-1.

FIGURE 7-1: Interrupt Logic



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, PIE2 and PIE3 registers)

The INTCON, PIR1, PIR2 and PIR3 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 "Section7.5 "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.

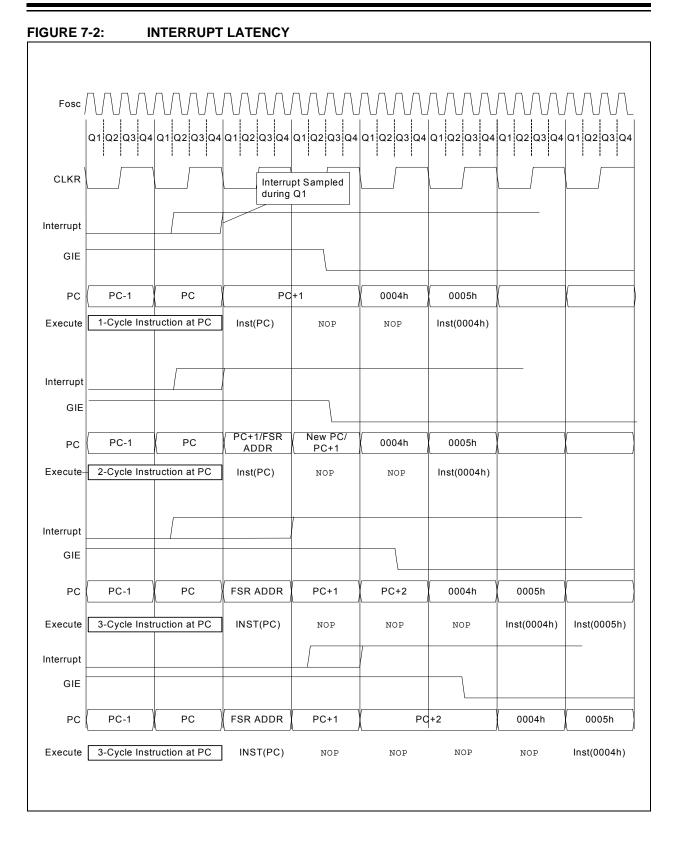
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

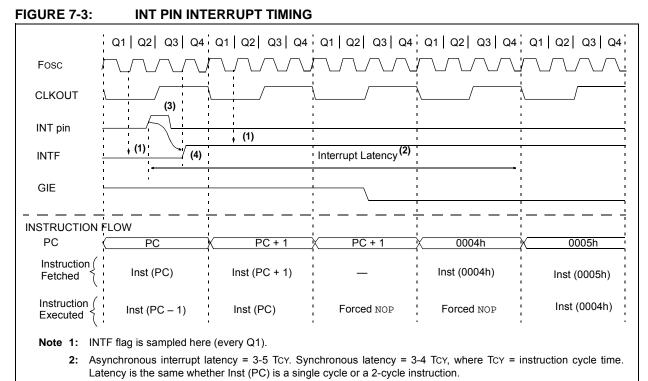
Note 1:	Individual interre		rrupt	flag	bits	s are	set,
	regardless	of	the	state	of	any	other
	enable bits						

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.





3: For minimum width of INT pulse, refer to AC specifications in Section35.0 "Electrical Specifications".

4: 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 Section8.0 "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 Register Definitions: Interrupt Control

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-0/0		
GIE ⁽¹⁾	PEIE ⁽²⁾	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF ⁽³⁾		
bit 7							bit (
Legend:									
R = Reada	ble bit	W = Writable	bit	U = Unimplemented bit, read as '0'					
u = Bit is u	•		x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Resets						
'1' = Bit is s	set	'0' = Bit is cle	ared						
bit 7	GIE: Global	Interrupt Enable	e bit ⁽¹⁾						
		all active interru all interrupts	ıpts						
bit 6	1 = Enables	PEIE: Peripheral Interrupt Enable bit ⁽²⁾ 1 = Enables all active peripheral interrupts 0 = Disables all peripheral interrupts							
bit 5	1 = Enables	ner0 Overflow Ir the Timer0 inter the Timer0 inter	rupt	e bit					
bit 4	1 = Enables	xternal Interrupt the INT externa the INT externa	l interrupt						
bit 3	1 = Enables	upt-on-Change the interrupt-on the interrupt-or	-change						
bit 2	1 = TMR0 re	ner0 Overflow Ir gister has overf gister did not ov	lowed	it					
bit 1	1 = The INT	kternal Interrupt external interru external interru	pt occurred	ır					
bit 0	1 = When at	upt-on-Change least one of the the interrupt-on	interrupt-on-o	change pins ch					
Note 1:	Interrupt flag bits enable bit or the appropriate inter	Global Interrupt	Enable bit, G	E of the INTCO	ON register. Use				
2:	Bit PEIE of the IN	NTCON register	must be set t	o enable any p	eripheral interru	upt.			
3:	The IOCIF Flag	oit is read-only a	and cleared w	hen all the inte	rrupt-on-change	e flags in the IC	CxF register		

REGISTER 7-1: INTCON: INTERRUPT CONTROL 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					
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE					
bit 7							bit (
Legend:												
R = Readab		W = Writable			nented bit, read							
u = Bit is un	0	x = Bit is unki		-n/n = value a	at POR and BO	R/Value at all o	ther Resets					
'1' = Bit is s	et	'0' = Bit is cle	ared									
bit 7	TMR1GIE: Ti	mer1 Gate Inte	rrunt Enable h	nit								
		TMR1GIE: Timer1 Gate Interrupt Enable bit 1 = Enables the Timer1 gate acquisition interrupt										
		0 = Disables the Timer1 gate acquisition interrupt										
bit 6	ADIE: Analog	ADIE: Analog-to-Digital Converter (ADC) Interrupt Enable bit										
		1 = Enables the ADC interrupt										
		the ADC interre										
bit 5		RCIE: EUSART Receive Interrupt Enable bit										
		 Enables the EUSART receive interrupt Disables the EUSART receive interrupt 										
bit 4		TXIE: EUSART Transmit Interrupt Enable bit										
		1 = Enables the EUSART transmit interrupt										
		0 = Disables the EUSART transmit interrupt										
bit 3	SSP1IE: Syn	SSP1IE: Synchronous Serial Port (MSSP) Interrupt Enable bit										
		1 = Enables the MSSP interrupt										
		0 = Disables the MSSP interrupt										
bit 2		CCP1IE: CCP1 Interrupt Enable bit										
		1 = Enables the CCP1 interrupt										
bit 1		 0 = Disables the CCP1 interrupt TMR2IE: TMR2 to PR2 Match Interrupt Enable bit 										
		1 = Enables the Timer2 to PR2 match interrupt										
		0 = Disables the Timer2 to PR2 match interrupt										
bit 0	TMR1IE: Tim	TMR1IE: Timer1 Overflow Interrupt Enable bit										
		1 = Enables the Timer1 overflow interrupt										
	0 = Disables	the Timer1 ove	erflow interrupt									
	Bit PEIE of the IN											
S	set to enable any	periprieral inter	rupt.									

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

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	R/W-0/0				
OSFIE	C2IE	C1IE	_	BCL1IE	TMR6IE	TMR4IE	CCP2IE				
bit 7							bit C				
Legend:											
R = Readable bit		W = Writable	bit	U = Unimpler	nented bit, read	as '0'					
u = Bit is unchanged		x = Bit is unkn	own	-n/n = Value a	at POR and BO	R/Value at all c	ther Resets				
'1' = Bit is set		'0' = Bit is cleared									
1.11.7											
bit 7		lator Fail Interru	•	t .							
	 Enables the Oscillator Fail interrupt Disables the Oscillator Fail interrupt 										
bit 6	C2IE: Comparator C2 Interrupt Enable bit										
	1 = Enables the Comparator C2 interrupt										
	0 = Disables the Comparator C2 interrupt										
bit 5	C1IE: Comparator C1 Interrupt Enable bit										
	 Enables the Comparator C1 interrupt Disables the Comparator C1 interrupt 										
L:1		•	•								
bit 4	•	ted: Read as '									
bit 3	BCL1IE: MSSP Bus Collision Interrupt Enable bit										
	 Enables the MSSP Bus Collision Interrupt Disables the MSSP Bus Collision Interrupt 										
bit 2		R6 to PR6 Mate									
	1 = Enables the Timer6 to PR6 match interrupt										
	0 = Disables the Timer6 to PR6 match interrupt										
bit 1	TMR4IE: TMR4 to PR4 Match Interrupt Enable bit										
	 = Enables the Timer4 to PR4 match interrupt 0 = Disables the Timer4 to PR4 match interrupt 										
bit 0		P2 Interrupt Ena									
		2 interrupt is e									
	0 = The CCF	2 interrupt is n	ot enabled								
Note 1. Bit	DEIE of the INIT	FCON register r	must ha sat ti	n enable any ne	rinheral interru	ot					

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

Note 1: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

U-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					
—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE					
bit 7							bit					
Legend:												
R = Readable bit		W = Writable			nented bit, read							
u = Bit is un	•		x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Rese									
'1' = Bit is se	et	'0' = Bit is cle	ared									
bit 7-6	Unimplomo	nted: Read as '	o'									
bit 5	-			rator (CMC) In	torrupt Epoble	hit						
DIL 5		CWGIE : Complementary Waveform Generator (CWG) Interrupt Enable bit 1 = Enables the CWG interrupt										
		1 = Disables the CWG interrupt 0 = Disables the CWG interrupt										
bit 4	ZCDIE: Zero	ZCDIE: Zero-Cross Detection (ZCD) Interrupt Enable bit										
		1 = Enables the ZCD interrupt										
	0 = Disables the ZCD interrupt											
bit 3		CLC4IE: Configurable Logic Block 4 Interrupt Enable bit										
		 1 = Enables the CLC 4 interrupt 0 = Disables the CLC 4 interrupt 										
			•									
bit 2		CLC3IE: Configurable Logic Block 3 Interrupt Enable bit 1 = Enables the CLC 3 interrupt										
		s the CLC 3 inte	•									
bit 1		CLC2IE: Configurable Logic Block 2 Interrupt Enable bit										
		1 = Enables the CLC 2 interrupt										
	0 = Disables	s the CLC 2 inte	errupt									
bit 0	CLC1IE: Cor	CLC1IE: Configurable Logic Block 1 Interrupt Enable bit										
		the CLC 1 inte										
	0 = Disables	s the CLC 1 inte	errupt									
			<u>-</u>									
	Bit PEIE of the IN	•										
S	et to enable any	peripheral inter	rupt.									

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

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		
SCANI	E CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE		
bit 7							bit 0		
Legend:									
R = Reada	able bit	W = Writable I	oit	U = Unimplemented bit, read as '0'					
u = Bit is ι	unchanged	x = Bit is unkn	own	-n/n = Value	at POR and BO	R/Value at all of	ther Resets		
'1' = Bit is	set	'0' = Bit is clea	ared						
bit 7	SCANIE: So	canner Interrupt	Enable bit						
		s the scanner in							
		s the scanner ir	•						
bit 6		C Interrupt Enat							
		s the CRC intern s the CRC inter							
bit 5		E: SMT2 Pulse	•	ion Interrunt F	nahle hit				
bit o		s the SMT2 acq	•	•					
		s the SMT2 acc							
bit 4	SMT2PRAI	E: SMT2 Period	Acquisition Int	errupt Enable	bit				
		s the SMT2 acq							
	0 = Disable	s the SMT2 acc	uisition interru	pt					
bit 3		IT2 Match Inter	•						
		s the SMT2 peri							
hit O		s the SMT2 per		•	nabla bit				
bit 2		E: SMT1 Pulse s the SMT1 acq	•	•					
		s the SMT1 acq							
bit 1		E: SMT1 Period		-	bit				
		s the SMT1 acq	•	•					
	0 = Disable	s the SMT1 acc	uisition interru	pt					
bit 0	SMT1IE: SM	IT1 Match Inter	rupt Enable bit						
		s the SMT1 peri							
	0 = Disable	s the SMT1 per	lod match inter	rupt					
			<u>_</u>						
Note:	Bit PEIE of the II								
	set to enable any	peripheral inte	rrupt.						

REGISTER 7-5: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/HS-0/0	R/W/HS-0/0	R/W/HS-0/0			
TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	_	AT1IE	PID1EIE	PID1DIE			
bit 7							bit (
Legend:										
R = Readable	bit	W = Writable	bit	•	mented bit, read					
u = Bit is unch	anged	x = Bit is unkr	iown	-n/n = Value	at POR and BC	R/Value at all o	other Resets			
'1' = Bit is set		'0' = Bit is clea	ared							
bit 7	TMP3CIE	- imer3 Cate Int	arrunt Enable h	sit						
	TMR3GIE : Timer3 Gate Interrupt Enable bit 1 = Enables the Timer3 Gate interrupt									
	0 = Disables the Timer3 Gate interrupt									
bit 6	TMR3IE: Tin	ner3 Overflow I	nterrupt Enable	e bit						
	1 = Enables the Timer3 overflow interrupt									
	0 = Disable	s the Timer3 ov	erflow interrup	t						
bit 5		imer5 Gate Inte	•	bit						
		Enables the Timer5 Gate interrupt								
1.11.4	0 = Disables the Timer5 Gate interrupt									
bit 4	TMR5IE: Timer5 Overflow Interrupt Enable bit									
	 1 = Enables the Timer5 overflow interrupt 0 = Disables the Timer5 overflow interrupt 									
bit 3	Unimpleme	nted: Read as	·0'							
bit 2	AT1IE: Angu	ular Timer 1 Inte	errupt Enable b	it						
		s the Angular Ti								
		s the Angular T	•	t						
bit 1		D Error Interrup								
		the PID error i								
h it 0		s the PID error	•							
bit 0		D Interrupt Ena								
		s the PID interr	•							
			~p.							
Note: Bit		NTCON registe	r must he							
		peripheral inte								

REGISTER 7-6: PIE5: PERIPHERAL INTERRUPT ENABLE REGISTER 5

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		
TMR10	SIE ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF		
bit 7							bit		
Legend:									
R = Read	able bit	W = Writable	bit	U = Unimpler	mented bit, read	l as '0'			
u = Bit is unchanged		x = Bit is unki	nown	-n/n = Value at POR and BOR/Value at all other Res					
'1' = Bit is	set	ʻ0' = Bit is cle	ared						
bit 7	TMR1GI	F: Timer1 Gate Inte	rrupt Flag bit						
		upt is pending upt is not pending							
bit 6	ADIF: AD	DC Interrupt Flag bi	t						
		upt is pending upt is not pending							
bit 5	RCIF: EL	JSART Receive Int	errupt Flag bit						
		upt is pending upt is not pending							
bit 4	TXIF: EU	ISART Transmit Int	errupt Flag bit						
		upt is pending							
bit 3		upt is not pending		Interrupt Flog	hit				
bit 5	1 = Interr	Synchronous Seria upt is pending upt is not pending		intenupt riag	bit				
bit 2	CCP1IF: 1 = Interr	CCP1 Interrupt Fla	g bit						
		upt is not pending							
bit 1		Timer2 to PR2 Inte	errupt Flag bit						
		upt is pending upt is not pending							
bit 0		Timer1 Overflow Ir	nterrupt Flag b	it					
	1 = Interr	upt is pending upt is not pending	1 0						
Note:	condition occu its correspond Interrupt Enab register. User	oits are set when an irs, regardless of the ling enable bit or the ble bit, GIE of the software should er errupt flag bits are c	e state of le Global INTCON isure the						

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

to enabling an interrupt.

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	R/W-0/0			
OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF			
bit 7							bit			
Legend:										
R = Readal	ble bit	W = Writable I	bit	U = Unimpler	mented bit, read	d as '0'				
u = Bit is ur	nchanged	x = Bit is unkn	own	-n/n = Value at POR and BOR/Value at all other Res						
'1' = Bit is s	set	'0' = Bit is clea	ared							
bit 7	OSFIF : Osci	llator Fail Interru	pt Flag bit							
	1 = Interrupt									
bit 6	C2IF: Compa	arator C2 Interru	pt Flag bit							
	1 = Interrupt	is pending is not pending								
bit 5	-	arator C1 Interru	ot Flag bit							
	1 = Interrupt		pri log all							
	0 = Interrupt	is not pending								
bit 4	Unimpleme	nted: Read as 'o)'							
bit 3	BCL1IF: MSSP Bus Collision Interrupt Flag bit									
		1 = Interrupt is pending								
		0 = Interrupt is not pending								
bit 2		her6 to PR6 Inte	rrupt Flag bi	t						
	1 = Interrupt	is pending is not pending								
bit 1	•	her4 to PR4 Inter	rrunt Elaa hit	ŀ						
	1 = Interrupt		inupt i lag bi	L						
		is not pending								
bit 0	CCP2IF: CC	P2 Interrupt Flag	a bit							
	1 = Interrupt		-							
	0 = Interrupt	is not pending								
	Interrupt flag bits									
	condition occurs, its corresponding									
	Enable bit, GIE									
	User software	should ensu	re the							
	appropriate interru		ear prior							
	to enabling an inte	errupt.								

REGISTER 7-8: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

U-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				
—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF				
bit 7							bit (
Legend:											
R = Read	able bit	W = Writable	bit	U = Unimpler	nented bit, read	1 as '0'					
u = Bit is	unchanged	x = Bit is unk	nown	-n/n = Value a	at POR and BO	R/Value at all c	other Resets				
'1' = Bit is	s set	'0' = Bit is cle	ared								
bit 7-6	Unimplem	Unimplemented: Read as '0'									
bit 5	CWGIF: CV	VG Interrupt Flag	g bit								
		ot is pending ot is not pending									
bit 4	ZCDIF: ZC	D Interrupt Flag	oit								
		ot is pending									
	0 = Interrup	ot is not pending									
bit 3		CLC4IF: Configurable Logic Block 4 Interrupt Flag bit									
		1 = Interrupt is pending 0 = Interrupt is not pending									
h:4 0		0 = Interrupt is not pending									
bit 2		CLC3IF: Configurable Logic Block 3 Interrupt Flag bit									
		1 = Interrupt is pending 0 = Interrupt is not pending									
bit 1		onfigurable Logic	Block 2 Interr	rupt Flag bit							
		ot is pending		1 3							
	0 = Interrup	ot is not pending									
bit 0	CLC1IF: Co	onfigurable Logic	Block 1 Interr	upt Flag bit							
		ot is pending									
	0 = Interrup	ot is not pending									
Note:		are set when ar									
		, regardless of th									
		g enable bit or th of the INTCON									
	User software		•								
		rupt flag bits are o	clear prior								
	to onobling on in	to mark the									

REGISTER 7-9: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3

to enabling an interrupt.

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
SCANI	F CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
bit 7				•			bit
Legend:							
R = Reada	able bit	W = Writable I	bit	U = Unimple	mented bit, read	l as '0'	
u = Bit is u	•	x = Bit is unkn	own	-n/n = Value	at POR and BO	R/Value at all ot	her Resets
'1' = Bit is	set	'0' = Bit is clea	ared				
bit 7	SCANIF: So	canner Interrupt	Flag bit				
		t is pending t is not pending					
bit 6	CRCIF: CR	C Interrupt Flag	bit				
	1 = Interrup	t is pending					
hit 5	•	t is not pending	Midth Acquisiti	on Intorrunt E	log bit		
bit 5		IF: SMT2 Pulse If is pending	width Acquisiti	оп іпцентирі г	nag bit		
		t is not pending					
bit 4		F: SMT2 Period	Acquisition Int	errupt Flag bi	t		
		t is pending					
	0 = Interrup	t is not pending					
bit 3		MT2 Match Inter	upt Flag bit				
		t is pending t is not pending					
bit 2	-	IF: SMT1 Pulse	Width Acquisiti	on Interrunt F	ilaa hit		
		t is pending		on interrupt i	lag bit		
		t is not pending					
bit 1	SMT1PRAI	F: SMT1 Period	Acquisition Int	errupt Flag bi	t		
		t is pending					
	-	t is not pending					
bit 0		MT1 Match Inter	upt Flag bit				
		t is pending t is not pending					
	0 – interrup	t is not pending					
Note:	Interrupt flag bits						
	condition occurs,						
	its corresponding Enable bit, GIE						
	User software						
	appropriate interi	rupt flag bits are o					
	to enabling an in	terrupt.					

REGISTER 7-10: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/HS-0/0	R/W/HS-0/0	R/W/HS-0/0				
TMR3GI	F TMR3IF	TMR5GIF	TMR5IF	-	AT1IF	PID1EIF	PID1DIF				
bit 7					·	·	bit (
Legend:											
R = Reada	ıble bit	W = Writable	bit	•	mented bit, read						
u = Bit is u	•	x = Bit is unkr	iown	-n/n = Value	at POR and BC	R/Value at all o	ther Resets				
'1' = Bit is :	set	'0' = Bit is clea	ared								
bit 7	TMR3GIF:	Timer3 Gate Inte	errupt Flag bit								
	1 = Interrup	t is pending									
	0 = Interrup) = Interrupt is not pending									
bit 6	TMR3IF: Tir	mer3 Overflow I	nterrupt Flag b	it							
	1 = Interrup										
		t is not pending									
bit 5		Timer5 Gate Inte	errupt Flag bit								
	1 = Interrup	t is penaing t is not pending									
bit 4		mer5 Overflow I	nterrunt Flag h	it							
	1 = Interrup		nien aprillag b	it.							
		t is not pending									
bit 3	Unimpleme	nted: Read as ')'								
bit 2	AT1IF: Ang	ular Timer 1 Inte	errupt Flag bit								
	1 = Interrup										
		t is not pending									
bit 1		D Error Interrup	ot Flag bit								
	1 = Interrup										
	-	t is not pending									
bit 0		ID Interrupt Flag	j bit								
	1 = Interrup	t is penaing t is not pending									
	0 – Interrup	t is not pending									
Note:	Interrupt flag bits										
	condition occurs,										
	its corresponding Interrupt Enable										
	register. User so										
	appropriate interr	upt flag bits are									
	to enabling an int	terrupt.									

REGISTER 7-11: PIR5: PERIPHERAL INTERRUPT REQUEST REGISTER 5

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	97
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		223
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PIE3	—	_	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	108
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IF	101
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	_	AT1IE	PID1EIE	PID1DIE	102
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
PIR2	OSFIF	C2IF	C1IF	_	BCL1IF	TMR6IF	TMR4IF	CCP2IF	112
PIR3	—	_	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	113
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	106
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	_	AT1IF	PID1EIF	PID1DIF	107

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. TO bit of the STATUS register is set.
- 4. CPU clock is disabled.
- 5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
- 6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
 - LFINTOSC
 - T1CKI
 - Timer1 oscillator
- 7. ADC is unaffected, if the dedicated FRC oscillator is selected.
- 8. I/O ports maintain the status they had before SLEEP was executed (driving high, low or highimpedance).
- 9. Resets other than WDT are not affected by Sleep mode.

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

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

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

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

Examples of internal circuitry that might be sourcing current include the FVR module. See **Section 15.0 "Fixed Voltage Reference (FVR)"** for more information on this module.

8.1 Wake-up from Sleep

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

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

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

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

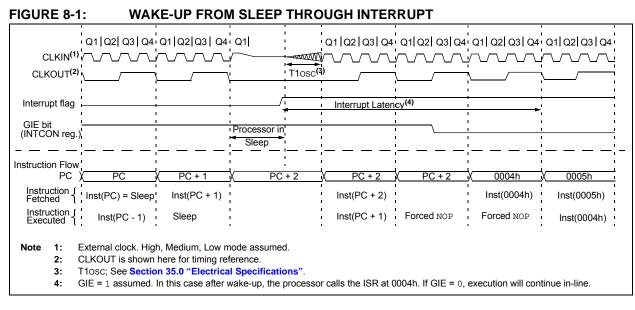
The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

8.1.1 WAKE-UP USING INTERRUPTS

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

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

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



8.2 Low-Power Sleep Mode

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

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

8.2.1 SLEEP CURRENT VS. WAKE-UP TIME

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

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

8.2.2 PERIPHERAL USAGE IN SLEEP

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

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

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

Please refer to sections **Section 28.11 "Operation During Sleep**" for more information.

The PIC16LF1615/9 does not have a con-Note: figurable Low-Power Sleep mode. PIC16LF1615/9 is an unregulated device and is always in the lowest power state when in Sleep, with no wake-up time penalty. This device has a lower maximum voltage VDD and I/O than the PIC16F1615/9. See **Section** 35.0 "Electrical Specifications" for more information.

8.3 Register Definitions: Voltage Regulator Control

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

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

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

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

Note 1: PIC16F1615/9 only.

2: See Section 35.0 "Electrical Specifications".

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
IOCAF	—	_	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	180
IOCAN	—	_	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	180
IOCAP	—	_	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	180
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	180
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	180
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	180
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIE2	OSFIE	C2IE	C1IE	_	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PIE3	—	_	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	108
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IF	101
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	_	AT1IE	PID1EIE	PID1DIE	102
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
PIR2	OSFIF	C2IF	C1IF	_	BCL1IF	TMR6IF	TMR4IF	CCP2IF	112
PIR3	—	-	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	113
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	106
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	_	AT1IF	PID1EIF	PID1DIF	107
STATUS	—	—	—	TO	PD	Z	DC	С	25
WDTCON0	—	—			WDTPS<4:0>			SEN	116

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

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

Note 1: PIC16(L)F1619 only.

9.0 WINDOWED WATCHDOG TIMER (WDT)

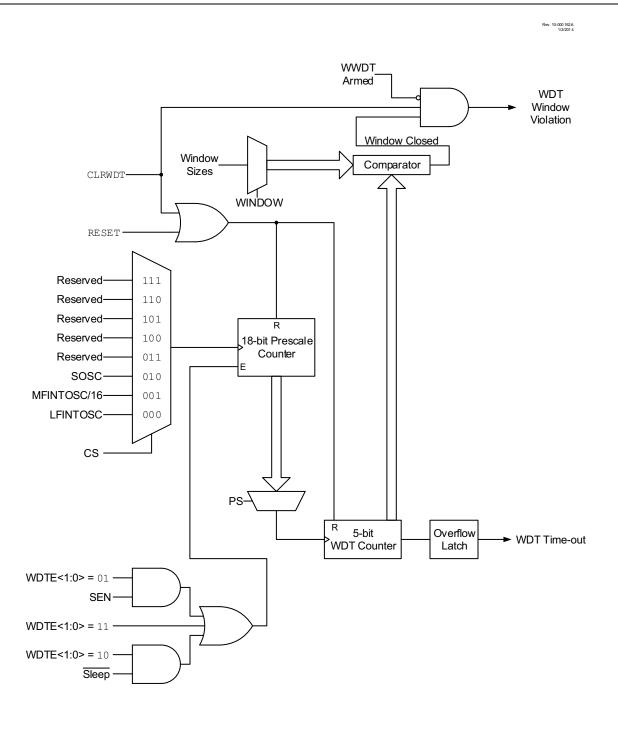
The Watchdog Timer (WDT) is a system timer that generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events. The Windowed Watchdog Timer (WDT) differs in that CLRWDT instructions are only accepted when they are performed within a specific window during the time-out period.

The WDT has the following features:

- Selectable clock source
- · Multiple operating modes
 - WDT is always on
 - WDT is off when in Sleep
 - WDT is controlled by software
 - WDT is always off
- Configurable time-out period is from 1 ms to 256 seconds (nominal)
- Configurable window size from 12.5 to 100
 percent of the time-out period
- Multiple Reset conditions
- Operation during Sleep

PIC16(L)F1615/9





9.1 Independent Clock Source

The WDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of either the WDTCCS<2:0> configuration bits or the WDTCS<2:0> bits of WDTCON1. Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section35.0 "Electrical Specifications**" for LFINTOSC and MFINTOSC tolerances.

9.2 WDT Operating Modes

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

9.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Words 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 Words 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 Words are set to '01', the WDT is controlled by the SEN bit of the WDTCON0 register.

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

WDTE<1:0>	SEN	Device Mode	WDT Mode
11	Х	Х	Active
10		Awake	Active
10	х	Sleep	Disabled
0.1	1	х	Active
01	0	х	Disabled
00	Х	х	Disabled

TABLE 9-1: WDT OPERATING MODES

9.3 Time-Out Period

The WDTPS bits of the WDTCON0 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 Watchdog Window

The Watchdog Timer has an optional Windowed mode that is controlled by the WDTCWS<2:0> Configuration bits and WINDOW<2:0> bits of the WDTCON1 register. In the Windowed mode, the CLRWDT instruction must occur within the allowed window of the WDT period. Any CLRWDT instruction that occurs outside of this window will trigger a window violation and will cause a WDT Reset, similar to a WDT time out. See Figure 9-2 for an example.

The window size is controlled by the WDTCWS<2:0> Configuration bits, or the WINDOW<2:0> bits of WDTCON1, if WDTCWS<2:0> = 111.

In the event of a window violation, a Reset will be generated and the WDTWV bit of the PCON register will be cleared. This bit is set by a POR or can be set in firmware.

9.5 Clearing the WDT

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

- Any Reset
- Valid CLRWDT instruction is executed
- · Device enters Sleep
- · Device wakes up from Sleep
- · WDT is disabled
- · Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or WDTCON1 registers

9.5.1 CLRWDT CONSIDERATIONS (WINDOWED MODE)

When in Windowed mode, the WDT must be armed before a CLRWDT instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a CLRWDT instruction without performing such an arming action will trigger a window violation.

See Table 9-2 for more information.

9.6 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.

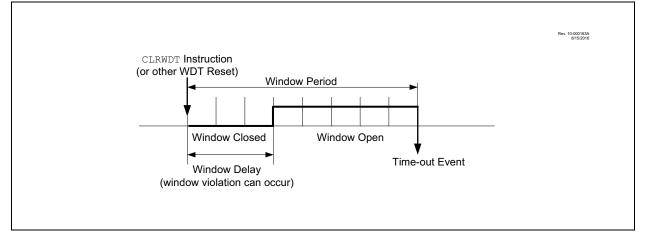
The WDT remains clear until the OST, if enabled, completes. See **Section5.0** "**Oscillator Module**" for more information on the OST.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The TO and PD bits in the STATUS register are changed to indicate the event. The RWDT bit in the PCON register can also be used. See Section3.0 "Memory Organization" for more information.

TABLE 9-2: WDT CLEARING CONDITIONS

Conditions	WDT					
WDTE<1:0> = 00						
WDTE<1:0> = 01 and SEN = 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					

FIGURE 9-2: WINDOW PERIOD AND DELAY



9.7 Register Definitions: Windowed Watchdog Timer Control

U-0	U-0	R/W ⁽³⁾ -q/q ⁽²⁾	R/W ⁽³⁾ -q/q	²⁾ R/W ⁽³⁾ -q/q ⁽²⁾	R/W ⁽³⁾ -q/q ⁽²⁾	R/W ⁽³⁾ -q/q ⁽²⁾	R/W-0/0
—	—			WDTPS<4:0>(1)			SEN
oit 7							bit
Legend:							
R = Reada	able bit	W = Writable	bit	U = Unimplem	ented bit, read	l as '0'	
u = Bit is u	inchanged	x = Bit is unkr	nown	-n/n = Value a	t POR and BO	R/Value at all oth	ner Resets
'1' = Bit is set '0' = Bit is cleared			q = Value dep	ends on condit	ion		
bit 7-6	Unimpleme	ented: Read as '	0'				
bit 5-1	WDTPS<4:	0>: Watchdog Ti	mer Prescale	e Select bits ⁽¹⁾			
		Prescale Rate					
	11111 = R	Reserved. Results	s in minimum	interval (1:32)			
	•						
	•						
	10011 = F	Reserved. Results	s in minimum	interval (1:32)			
	10010 = 1	:8388608 (2 ²³) (Interval 256s	nominal)			
	10001 = 1	:4194304 (2 ²²) (Interval 128s	nominal)			
		:2097152 (2 ²¹) (1					
	01111 = 1	:1048576 (2 ²⁰) (1	Interval 32s i	nominal)			
	01110 = 1 01101 = 1	:524288 (2 ¹⁹) (In :262144 (2 ¹⁸) (In :131072 (2 ¹⁷) (In	iterval 8s no	minal)			
	01100 = 1	:131072 (2 ¹⁷) (In	nterval 4s no	minal)			
	01011 = 1	:65536 (Interval	2s nominal)	(Reset value)			
	01010 = 1	:32768 (Interval	1s nominal)				
		:16384 (Interval					
		:8192 (Interval 2					
		:4096 (Interval 1) :2048 (Interval 6)					
		:1024 (Interval 3		,			
		:512 (Interval 16					
	00011 = 1	:256 (Interval 8 r	ns nominal)				
		:128 (Interval 4 r					
		:64 (Interval 2 m	,				
L:1 0		:32 (Interval 1 m	,	de a Tirre e a bit			
bit 0		are Enable/Disat	ole for watch	laog Timer bit			
	<u>If WDTE<1:</u> This bit is ig						
	If WDTE<1:						
	1 = WDT is						
	0 = WDT is						
	If WDTE<1:						
	This bit is ig	nored.					
lote 1:	Times are appr	oximate. WDT tir	ne is based (on 31 kHz LFINTC	DSC.		

REGISTER 9-1: WDTCON0: WATCHDOG TIMER CONTROL REGISTER 0

- Note 1: Times are approximate. WDT time is based on 31 kHz LFINTOSC.
 - 2: When WDTCPS <4:0> in CONFIG3 = 11111, the Reset value of WDTPS<4:0> is 01011. Otherwise, the Reset value of WDTPS<4:0> is equal to WDTCPS<4:0> in CONFIG3.
 - 3: When WDTCPS <4:0> in CONFIG3 \neq 11111, these bits are read-only.

U-0	$R/W^{(3)}_{-q/q}^{(1)} R/W^{(3)}_{-q/q}^{(1)} R/W^{(3)}_{-q/q}^{(1)}$	U-0	R/W ⁽⁴⁾ -q/q ⁽²⁾	R/W ⁽⁴⁾ -q/q ⁽²⁾	R/W ⁽⁴⁾ -q/q ⁽²⁾
-	WDTCS<2:0>	-		WINDOW<2:0>	
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	q = Value depends on condition

bit 7 Unimplemented: Read as '0'

bit 6-4 WDTCS<2:0>: Watchdog Timer Clock Select bits

- 111 = Reserved
 - •
 - •
 - 010 = Reserved
 - 001 = MFINTOSC 31.25 kHz
 - 000 = LFINTOSC 31 kHz
- bit 3 Unimplemented: Read as '0'
- bit 2-0 WINDOW<2:0>: Watchdog Timer Window Select bits

WINDOW<2:0>	Window delay Percent of time	Window opening Percent of time
111	N/A	100
110	12.5	87.5
101	25	75
100	37.5	62.5
011	50	50
010	62.5	37.5
001	75	25
000	87.5	12.5

- Note 1: If WDTCCS <2:0> in CONFIG3 = 111, the Reset value of WDTCS<2:0> is 000.
 - 2: The Reset value of WINDOW<2:0> is determined by the value of WDTCWS<2:0> in the CONFIG3 register.
 - **3:** If WDTCCS<2:0> in CONFIG3 \neq 111, these bits are read-only.
 - 4: If WDTCWS<2:0> in CONFIG3 \neq 111, these bits are read-only.

REGISTER 9-3: WDTPSL: WDT PRESCALE SELECT LOW BYTE REGISTER (READ ONLY)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
			PSCN	۲<7:0> ⁽¹⁾			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimpleme	ented bit, read as	'0'	
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and						alue at all other l	Resets

bit 7-0 **PSCNT<7:0>:** Prescale Select Low Byte bits⁽¹⁾

'0' = Bit is cleared

'1' = Bit is set

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should be read during normal operation.

REGISTER 9-4: WDTPSH: WDT PRESCALE SELECT HIGH BYTE REGISTER (READ ONLY)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
			PSCNT	<15:8> ⁽¹⁾			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimpleme	ented bit, read as	ʻ0'	
u = Bit is unchan	ged	x = Bit is unknown	n -n/n = Value at POR and BOR/Value at all other Resets				sets
'1' = Bit is set		'0' = Bit is cleared					

bit 7-0 **PSCNT<15:8>**: Prescale Select High Byte bits⁽¹⁾

Note 1: The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should be read during normal operation.

REGISTER 9-5: WDTTMR: WDT TIMER REGISTER (READ ONLY)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
	WDTTMR<3:0>	STATE	PSCNT<	17:16> ⁽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	

bit 7-3 WDTTMR<4:0>: Watchdog Timer Value

- bit 2 STATE: WDT Armed Status bit 1 = WDT is armed 0 = WDT is not armed
- bit 1-0 **PSCNT<17:16>**: Prescale Select Upper Byte bits⁽¹⁾
- **Note 1:** The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should be read during normal operation.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN		BIT5 BIT4 BIT3 BIT7 BIT1 BIT0			81			
PCON	STKOVF	STKUNF	WDTWV	RWDT	RMCLR	RI	POR	BOR	90
STATUS	—	_	_	TO	PD	Z	DC	С	25
WDTCON0	_	—		WDTPS<4:0> SEN					116
WDTCON1	-	V	WDTCS<2:0					116	
WDTPSL		PSCNT<7:0>							116
WDTPSH		PSCNT<15:8>							116
WDTTMR	_		WDTTM	1R<4:0>		STATE	PSCNT	<17:16>	116

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

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
	13:8	_		FCMEN	IESO	CLKOUTEN	BORE	BOREN<1:0> —		69
CONFIG1	7:0	CP	MCLRE	PWRTE	_	_	FOSC<2:0>		09	
	13:8	_	_	V	VDTCCS<2:	0>	WDTCWS<2:0>		<u> </u>	
CONFIG3	7:0	_	WDTE	E<1:0>		WDTCPS<4:0>				69

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 Words) and write protection (WRT<1:0> bits in Configuration Words).

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 Words.

10.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address up to a maximum of 16K 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.

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. How- ever, 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.

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

TABLE 10-1:	FLASH MEMORY
	ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)	
PIC16(L)F1615	32	32	
PIC16(L)F1619	52	52	

PIC16(L)F1615/9

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.

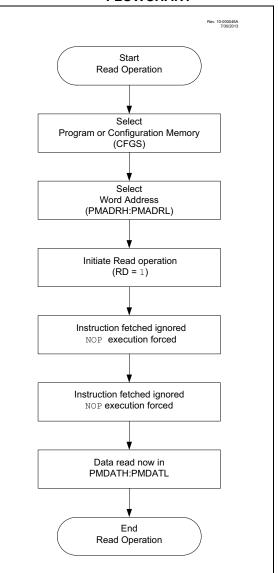
Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF PMCON1, RD" instruction to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register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

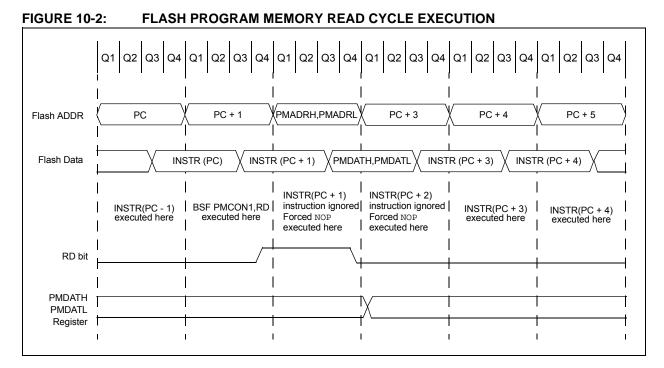
PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

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 FLOWCHART





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
            PROG_ADDR_LO
   MOVLW
                             ;
   MOVWF
            PMADRL
                             ; Store LSB of address
            PROG_ADDR_HI
   MOVLW
                              ;
   MOVWF
            PMADRH
                              ; Store MSB of address
   BCF
            PMCON1,CFGS
                             ; Do not select Configuration Space
   BSF
            PMCON1,RD
                              ; Initiate read
   NOP
                              ; Ignored (Figure 10-2)
   NOP
                              ; Ignored (Figure 10-2)
   MOVF
            PMDATL,W
                              ; Get LSB of word
   MOVWF
            PROG_DATA_LO
                             ; Store in user location
                             ; Get MSB of word
            PMDATH,W
   MOVF
   MOVWF
            PROG_DATA_HI
                             ; Store in user location
```

PIC16(L)F1615/9

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:

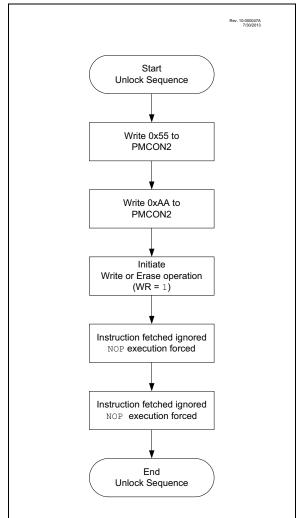
- 1. Write 55h to PMCON2
- 2. Write AAh to PMCON2
- 3. Set the WR bit in PMCON1
- 4. NOP instruction
- 5. NOP instruction

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 SEQUENCE FLOWCHART



PIC16(L)F1615/9

FLASH PROGRAM

MEMORY ERASE

FIGURE 10-4:

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.
- 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 immediately following the WR bit set instruction. 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.

FLOWCHART Rev. 10-000048A 7/30/2013 Start Erase Operation **Disable Interrupts** (GIE = 0) Select Program or Configuration Memory (CFGS) Select Row Address (PMADRH:PMADRL) Select Erase Operation (FREE = 1) Enable Write/Erase Operation (WREN = 1) Unlock Sequence (See Note 1) CPU stalls while Erase operation completes (2 ms typical) **Disable Write/Erase Operation** (WREN = 0) **Re-enable Interrupts** (GIE = 1) End Erase Operation Note 1: See Figure 10-3.

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 0x70 - 0x7F (common RAM)

	BCF BANKSEL MOVF MOVWF MOVF BCF	INTCON,GIE PMADRL ADDRL,W PMADRL ADDRH,W PMADRH PMCON1,CFGS	 ; Disable ints so required sequences will execute properly ; Load lower 8 bits of erase address boundary ; Load upper 6 bits of erase address boundary ; Not configuration space
	BSF	PMCON1, FREE	
	BSF	PMCON1,WREN	; Enable writes
Required Sequence	MOVLW MOVWF MOVLW MOVWF BSF NOP NOP	55h PMCON2 AAh 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	PMCON1,WREN	
	BSF	INTCON,GIE	; 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.

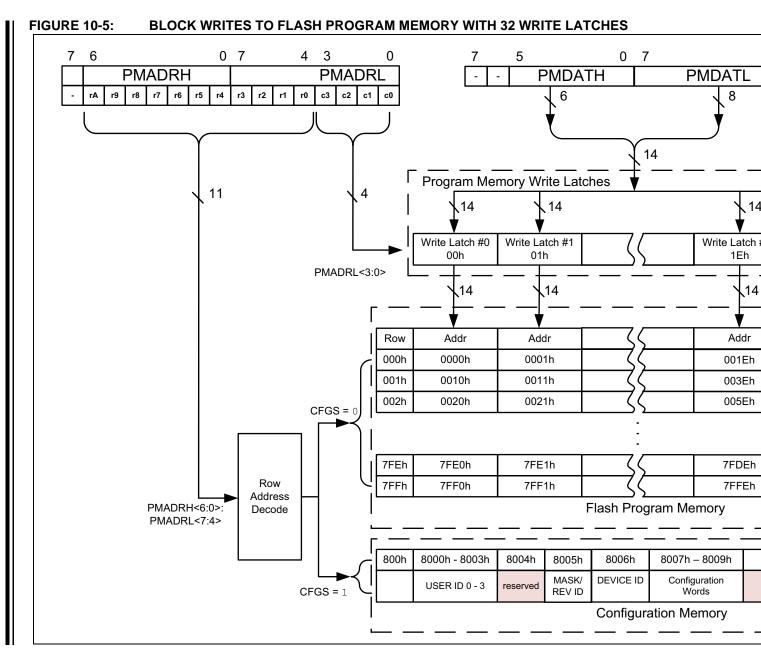
Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 10-5 (row writes to program memory with 32 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper 11 bits of PMADRH:PMADRL, (PMADRH<6:0>:PMADRL<7:4>) with the lower four bits of PMADRL, (PMADRL<7:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF. The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the PMDATH:PMDATL using the unlock sequence with LWLO = 1. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

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

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

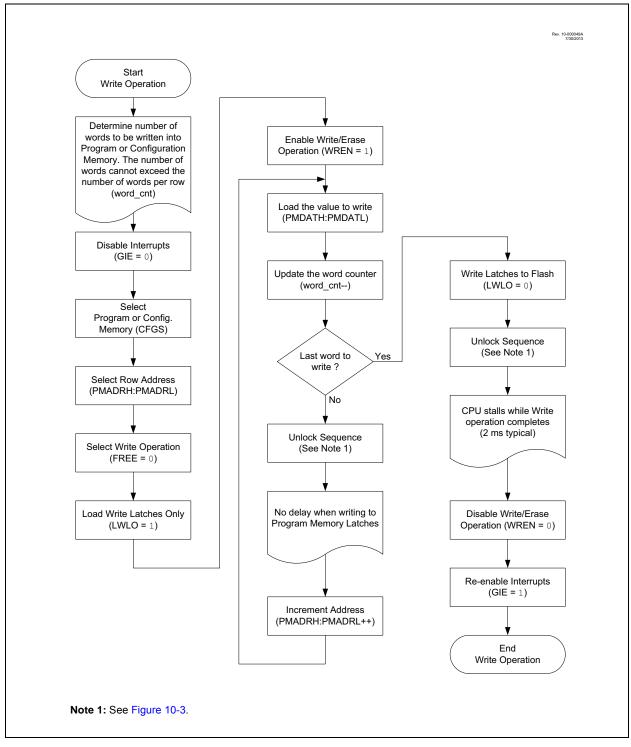


© 2014-2017 Microchip Technology Inc.

DS40001770D-page 127

PIC16(L)F1615/9





EXAMPLE 10-3: WRITING TO FLASH PROGRAM MEMORY (32 WRITE LATCHES)

	LE 10-3:	WRITING TO FLA	ASH PROGRAM MEMORY (32 WRITE LATCHES)
; This	write rout	ine assumes the f	following:
			starting at the address in DATA_ADDR
	-		en is made up of two adjacent bytes in DATA_ADDR,
		ttle endian forma.	
; 3. A	valid star	ting address (the	e Least Significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. AI	DDRH and AD	DRL are located i	n shared data memory 0x70 - 0x7F (common RAM)
;			
	BCF	INTCON,GIE	; Disable ints so required sequences will execute properly
	BANKSEL	PMADRH	; Bank 3
	MOVF	ADDRH,W	; Load initial address
	MOVWF	PMADRH	;
	MOVF	ADDRL,W	;
	MOVWF	PMADRL	;
	MOVLW	LOW DATA_ADDR	; Load initial data address
	MOVWF	FSROL	i
	MOVLW	HIGH DATA_ADDR	; Load initial data address
	MOVWF		;
	BCF		; Not configuration space
	BSF		; Enable writes
	BSF	PMCON1,LWLO	; Only Load Write Latches
LOOP			
	MOVIW	FSR0++	; Load first data byte into lower
	MOVWF	PMDATL	;
	MOVIW	FSR0++	; Load second data byte into upper
	MOVWF	PMDATH	;
	MOVF	PMADRL,W	; Check if lower bits of address are '00000'
	XORLW	0x1F	; Check if we're on the last of 32 addresses .
	ANDLW		; . This if less of 20 monda
	BTFSC GOTO	STATUS,Z	; Exit if last of 32 words, ;
	GOIO	START_WRITE	1
	MOVLW	55h	; Start of required write sequence:
	MOVWF		; Write 55h
- O	MOVLW		;
anc of	MOVWF		; Write AAh
Required Sequence	BSF		; Set WR bit to begin write
Sec	NOP		; NOP instructions are forced as processor
			; loads program memory write latches
	NOP		;
	INCF	PMADRL, F	; Still loading latches Increment address
	GOTO		; Write next latches
START_W	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
e e	MOVLW		i
uire	MOVWF	PMCON2	; Write AAh
Required Sequence	BSF	PMCON1,WR	; Set WR bit to begin write
шs	NOP		; NOP instructions are forced as processor writes
			; all the program memory write latches simultaneously
	NOP		; to program memory.
L			; After NOPs, the processor
			; stalls until the self-write process in complete
	DOE	DMOON1 LIE	; after write processor continues with 3rd instruction
	BCF	PMCON1, WREN	; Disable writes
	BSF	INTCON,GIE	; Enable interrupts

FLASH PROGRAM

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.

Read Operation (See Note 1) An image of the entire row read must be stored in RAM Modify Image The words to be modified are changed in the RAM image

Write Operation Use RAM image (See Note 3)

End Modify Operation

Note 1: See Figure 10-2. 2: See Figure 10-4. 3: See Figure 10-5.

FIGURE 10-7:

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.

Address	Function	Read Access	Write Access	
8000h-8003h	User IDs	Yes	Yes	
8006h/8005h	Device ID/Revision ID	Yes	No	
8007h-8009h	Configuration Words 1, 2, and 3	Yes	No	

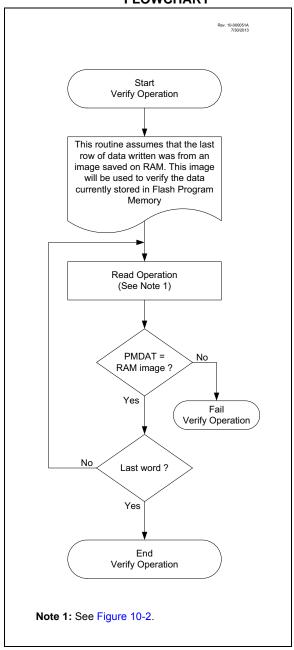
EXAMPLE 10-4: CONFIGURATION WORD AND DEVICE ID ACCESS

* PROG_ADD	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	PMADRL	; Select correct Bank				
MOVLW	PROG_ADDR_LO	;				
MOVWF	PMADRL	; Store LSB of address				
CLRF	PMADRH	; Clear MSB of address				
BSF	PMCON1,CFGS	; Select Configuration Space				
BCF	INTCON,GIE	; Disable interrupts				
BSF	PMCON1,RD	; Initiate read				
NOP		; Executed (See Figure 10-2)				
NOP		; Ignored (See Figure 10-2)				
BSF	INTCON,GIE	; Restore interrupts				
MOVF	PMDATL,W	; Get LSB of word				
MOVWF	PROG_DATA_LO	; Store in user location				
MOVF	PMDATH,W	; Get MSB of word				
MOVWF	PROG_DATA_HI	; 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 Register Definitions: Flash Program Memory Control

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 = Unimpleme	nted bit, read as '0	,	
u = Bit is unchanged		x = Bit is unknown		-n/n = Value at F	POR and BOR/Valu	ue at all other Res	ets
'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			
—	—		PMDAT<13:8>							
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 7-6 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	0 R/W-0/0 R/W-0/0 I				R/W-0/0	R/W-0/0	R/W-0/0			
PMADR<7:0>										
bit 7	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 **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
(1)				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

Note 1: Unimplemented, read as '1'.

U-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
_(1)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit (
Legend:							
R = Reada	ble bit	W = Writable b	it	U = Unimplemer	nted bit, read as	s 'O'	
S = Bit car	only be set	x = Bit is unkno	own	-n/n = Value at F	OR and BOR/	/alue at all other I	Resets
'1' = Bit is :	set	'0' = Bit is clea	red	HC = Bit is clear	ed by hardware)	
bit 7	Unimplemen	ted: Read as '1'					
bit 6	-	guration Select bit					
		Configuration, Use Flash program me		ID Registers			
bit 5	1 = Only the 0 = The add	Write Latches On addressed progra ressed program m itiated on the nex	am memory write emory write latcl				
bit 4	FREE: Progra	am Flash Erase E	nable bit				
	1 = Perform	s an erase operati s a write operatior	ion on the next V		rdware cleared	upon completion)	I
bit 3	1 = Conditio on any s	gram/Erase Error n indicates an imp et attempt (write ' gram or erase ope	proper program 1') of the WR bit	.)	e attempt or te	rmination (bit is s	et automatical
bit 2		am/Erase Enable		a normany			
511 2	1 = Allows p	rogram/erase cyc programming/eras	les	lash			
bit 1	WR: Write Co	ontrol bit					
	The ope The WR	a program Flash p ration is self-timed bit can only be se n/erase operation t	d and the bit is cleated) in	leared by hardwai n software.		on is complete.	
bit 0	RD: Read Co						
	1 = Initiates (not clea	a program Flash r ired) in software. t initiate a prograr		s one cycle. RD is	cleared in hard	lware. The RD bit	can only be s
Note 1:	Unimplemented bit						
2: 3:	The WRERR bit is The LWLO bit is igr	automatically set I			-	ase operation is st	tarted (WR = 1

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

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	bit	U = Unimpler	nented bit, read	as '0'				
S = Bit can onl	y be set	x = Bit is unkn	iown	-n/n = Value at POR and BOR/Value at all other Reset						
'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
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PMCON1	(1)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	134
PMCON2	Program Memory Control Register 2								
PMADRL	PMADRL<7:0>								
PMADRH	(1)	(1) PMADRH<6:0>							
PMDATL	PMDATL<7:0>								133
PMDATH	_	— — PMDATH<5:0>							

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

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
	13:8	_	_	FCMEN	IESO	CLKOUTEN	BOREI	N<1:0>	_	<u> </u>
CONFIG1	7:0	CP	MCLRE	PWRTE	—	—	FOSC<2:0>		69	
	13:8	_	_	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	<u> </u>
CONFIG2	7:0	ZCD	_	_	—	—	PPS1WAY	WRT<	<1:0>	68
	13:8	_	_	V	WDTCCS<2:0>			WDTCWS<2:0>		
CONFIG3	7:0	_	WDTE	E<1:0> WE			DTCPS<4:0>	69		

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

11.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE

The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- · Configurable Polynomial
- Any seed value up to 16 bits can be used
- · Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program memory user data
- Software loadable data registers for calculating CRC values not from the memory scanner

11.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using the scanner.

11.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the Flash memory using the built-in memory scanner or through user input RAM. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 11-1:

Rev. 10-000206A 1/8/2014

```
CRC-16-ANSI
x^{16} + x^{15} + x^2 + 1 (17 bits)
```

Standard 16-bit representation = 0x8005

CRCXORH = 0b1000000 CRCXORL = 0b0000010- (1)

Data Sequence: 0x55, 0x66, 0x77, 0x88 DLEN = 0b0111

PLEN = Ob1111

Data entered into the CRC: SHIFTM = 0: 01010101 01100110 01110111 10001000

SHIFTM = 1: 10101010 01100110 11101110 00010001

Check Value (ACCM = 1):

SHIFTM = 0: 0x32D6 CRCACCH = 0b00110010 CRCACCL = 0b11010110

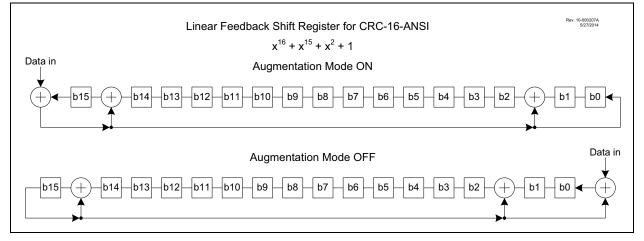
SHIFTM = 1: 0x6BA2 CRCACCH = 0b01101011 CRCACCL = 0b10100010

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always '1' and will always be treated as a '1' by the CRC for calculating the CRC check value. This bit will be read in software as a '0'.

11.3 CRC Polynomial Implementation

Any polynomial can be used. The polynomial and accumulator sizes are determined by the PLEN<3:0> bits. For an n-bit accumulator, PLEN = n-1 and the corresponding polynomial is n+1 bits. Therefore, the accumulator can be any size up to 16 bits with a corresponding polynomial up to 17 bits. The MSb and LSb of the polynomial are always '1', which is forced by hardware. All polynomial bits between the MSb and LSb are specified by the CRCXOR registers. For example, when using CRC-16-ANSI, the polynomial is defined as $X^{16}+X^{15}+X^2+1$. The X^{16} and $X^0 = 1$ terms are the MSb and LSb controlled by hardware. The X¹⁵ and X² terms are specified by setting the corresponding CRCXOR<15:0> bits with the value of 0x8004. The actual value is 0x8005 because the hardware sets the LSb to 1. However, the LSb of the CRCXORL register is unimplemented and always reads as '0'. Please refer to Example 11-1.

EXAMPLE 11-2: CRC LFSR EXAMPLE



11.4 CRC Data Sources

Data can be input to the CRC module in two ways:

- User data using the CRCDAT registers
- Flash using the Program Memory Scanner

To set the number of bits of data, up to 16 bits, the DLEN bits of CRCCON1 must be set accordingly. Only data bits in CRCDATA registers up to DLEN will be used, other data bits in CRCDATA registers will be ignored.

Data is moved into the CRCSHIFT as an intermediate to calculate the check value located in the CRCACC registers.

The SHIFTM bit is used to determine the bit order of the data being shifted into the accumulator. If SHIFTM is not set, the data will be shifted in MSb first. The value of DLEN will determine the MSb. If SHIFTM bit is set, the data will be shifted into the accumulator in reversed order, LSb first.

The CRC module can be seeded with an initial value by setting the CRCACC<15:0> registers to the appropriate value before beginning the CRC.

11.4.1 CRC FROM USER DATA

To use the CRC module on data input from the user, the user must write the data to the CRCDAT registers. The data from the CRCDAT registers will be latched into the shift registers on any write to the CRCDATL register.

11.4.2 CRC FROM FLASH

To use the CRC module on data located in Flash memory, the user can initialize the Program Memory Scanner as defined in Section 11.8, Program Memory Scan Configuration.

11.5 CRC Check Value

The CRC check value will be located in the CRCACC registers after the CRC calculation has finished. The check value will depend on two mode settings of the CRCCON register: ACCM and SHIFTM. When the ACCM bit is set, the CRC module augments the data with a number of zeros equal to the length of the polynomial to align the final check value. When the ACCM bit is not set, the CRC will stop at the end of the data. A number of zeros equal to the length of the polynomial can then be entered into CRCDAT to find the same check value as Augmented mode. Alternatively, the expected check value can be entered at this point to make the final result equal 0.

When the CRC check value is computed with the SHIFTM bit set (selecting LSb first), and the ACCM bit is set, then the final value in the CRCACC registers will be reversed such that the LSb will be in the MSb position and vice versa. This is the expected check value in bit reversed form. If you are creating a check value to be appended to a data stream, a bit reversal must be performed on the final value to achieve the correct checksum. The CRC can be used to do this reversal by the following method:

- Save the CRCACC value in user RAM space
- Clear the CRCACC registers
- Clear the CRCXOR registers
- Write the saved CRCACC value to the CRCDAT input

The properly oriented check value will be in the CRCACC registers as the result.

11.6 CRC Interrupt

The CRC will generate an interrupt when the BUSY bit transitions from 1 to 0. The CRCIF interrupt flag bit of the PIR4 register is set every time the BUSY bit transitions, regardless of whether or not the CRC interrupt is enabled. The CRCIF bit can only be cleared in software. The CRC interrupt enable is the CRCIE bit of the PIE4 register.

11.7 Configuring the CRC

The following steps illustrate how to properly configure the CRC.

- Determine if the automatic Program Memory scan will be used with the Scanner or manual calculation through the SFR interface and perform the actions specified in Section11.4 "CRC Data Sources", depending on which decision was made.
- 2. If desired, seed a starting CRC value into the CRCACCH/L registers.
- 3. Program the CRCXORH/L registers with the desired generator polynomial.
- Program the DLEN<3:0> bits of the CRCCON1 register with the length of the data word - 1 (refer to Example 11-1). This determines how many times the shifter will shift into the accumulator for each data word.
- Program the PLEN<3:0> bits of the CRCCON1 register with the length of the polynomial - 2 (refer to Example 11-1).
- 6. Determine whether shifting in trailing zeros is desired and set the ACCM bit of CRCCON0 register appropriately.
- 7. Likewise, determine whether the MSb or LSb should be shifted first and write the SHIFTM bit of CRCCON0 register appropriately.
- 8. Write the CRCGO bit of the CRCCON0 register to begin the shifting process.
- 9a. If manual SFR entry is used, monitor the FULL bit of CRCCON0 register. When FULL = 0, another word of data can be written to the CRCDATH/L registers, keeping in mind that CRCDATH should be written first if the data has >8 bits, as the shifter will begin upon the CRCDATL register being written.
- 9b. If the scanner is used, the scanner will automatically stuff words into the CRCDATH/L registers as needed, as long as the SCANGO bit is set.
- 10a. If using the Flash memory scanner, monitor the SCANIF (or the SCANGO bit) for the scanner to finish pushing information into the CRCDATA registers. After the scanner is completed, monitor the CRCIF (or the BUSY bit) to determine that the CRC has been completed and the check value can be read from the CRCACC registers. If both the interrupt flags are set (or both BUSY and SCANGO bits are cleared), the completed CRC calculation can be read from the CRCACCH/L registers.
- 10b.If manual entry is used, monitor the CRCIF (or BUSY bit) to determine when the CRCACC registers will hold the check value.

11.8 Program Memory Scan Configuration

If desired, the Program Memory Scan module may be used in conjunction with the CRC module to perform a CRC calculation over a range of program memory addresses. In order to set up the Scanner to work with the CRC, perform the following steps:

- Set the EN bit to enable the module. This can be performed at any point preceding the setting of the SCANGO bit, but if it gets disabled, all internal states of the Scanner are reset (registers are unaffected).
- Choose which memory access mode is to be used (see Section11.10 "Scanning Modes") and set the MODE bits of the SCANCON0 register appropriately.
- 3. Based on the memory access mode, set the INTM bits of the SCANCON0 register to the appropriate interrupt mode (see Section11.10.5 "Interrupt Interaction")
- 4. Set the SCANLADRL/H and SCANHADRL/H registers with the beginning and ending locations in memory that are to be scanned.
- 5. Begin the scan by setting the SCANGO bit in the SCANCON0 register. The scanner will wait (CRCGO must be set) for the signal from the CRC that it is ready for the first Flash memory location, then begin loading data into the CRC. It will continue to do so until it either hits the configured end address or an address that is unimplemented on the device, at which point the SCANGO bit will clear, Scanner functions will cease, and the SCANIF interrupt will be triggered. Alternately, the SCANGO bit can be cleared in software if desired.

11.9 Scanner Interrupt

The scanner will trigger an interrupt when the SCANGO bit transitions from 1 to 0. The SCANIF interrupt flag of PIR4 is set when the last memory location is reached and the data is entered into the CRCDATA registers. The SCANIF bit can only be cleared in software. The SCAN interrupt enable is the SCANIE bit of the PIE4 register.

11.10 Scanning Modes

The memory scanner can scan in four modes: Burst, Peek, Concurrent, and Triggered. These modes are controlled by the MODE bits of the SCANCON0 register. The four modes are summarized in Table 11-1.

11.10.1 BURST MODE

When MODE = 01, the scanner is in Burst mode. In Burst mode, CPU operation is stalled beginning with the operation after the one that sets the SCANGO bit, and the scan begins, using the instruction clock to execute. The CPU is held until the scan stops. Note that because the CPU is not executing instructions, the SCANGO bit cannot be cleared in software, so the CPU will remain stalled until one of the hardware end-conditions occurs. Burst mode has the highest throughput for the scanner, but has the cost of stalling other execution while it occurs.

11.10.2 CONCURRENT MODE

When MODE = 00, the scanner is in Concurrent mode. Concurrent mode, like Burst mode, stalls the CPU while performing accesses of memory. However, while Burst mode stalls until all accesses are complete, Concurrent mode allows the CPU to execute in between access cycles.

11.10.3 TRIGGERED MODE

When MODE = 11, the scanner is in Triggered mode. Triggered mode behaves identically to Concurrent mode, except instead of beginning the scan immediately upon the SCANGO bit being set, it waits for a rising edge from a separate trigger clock, the source of which is determined by the SCANTRIG register.

11.10.4 PEEK MODE

When MODE = 10, the scanner is in Peek mode. Peek mode waits for an instruction cycle in which the CPU does not need to access the NVM (such as a branch instruction) and uses that cycle to do its own NVM access. This results in the lowest throughput for the NVM access (and can take a much longer time to complete a scan than the other modes), but does so without any impact on execution times, unlike the other modes.

MC)DE<1:0>		Description						
WIC		First Scan Access	CPU Operation						
11	Triggered	As soon as possible following a trigger	Stalled during NVM access	CPU resumes execution following each access					
10	Peek	At the first dead cycle	Timing is unaffected	CPU continues execution following each access					
01	Burst	As soon as possible	Stalled during NV/M appage	CPU suspended until scan completes					
00	Concurrent	As soon as possible	Stalled during NVM access	CPU resumes execution following each access					

TABLE 11-1: SUMMARY OF SCANNER MODES

11.10.5 INTERRUPT INTERACTION

The INTM bit of the SCANCON0 register controls the scanner's response to interrupts depending on which mode the NVM scanner is in, as described in Table 11-2.

TABLE 11-2: SCAN INTERRUPT MODES

INTM	MODE<1:0>						
	MODE == Burst	MODE != Burst					
1	Interrupt overrides SCANGO to pause the burst and the interrupt handler executes at full speed; Scanner Burst resumes when interrupt completes.	Scanner suspended during interrupt response; interrupt executes at full speed and scan resumes when the interrupt is complete.					
0	Interrupts do not override SCANGO, and the scan (burst) operation will continue; interrupt response will be delayed until scan completes (latency will be increased).	Scanner accesses NVM during interrupt response. If MODE != Peak the interrupt handler execution speed will be affected.					

In general, if INTM = 0, the scanner will take precedence over the interrupt, resulting in decreased interrupt processing speed and/or increased interrupt response latency. If INTM = 1, the interrupt will take precedence and have a better speed, delaying the memory scan.

11.10.6 WDT INTERACTION

Operation of the WDT is not affected by scanner activity. Hence, it is possible that long scans, particularly in Burst mode, may exceed the WDT time-out period and result in an undesired device Reset. This should be considered when performing memory scans with an application that also utilizes WDT.

11.10.7 IN-CIRCUIT DEBUG (ICD) INTERACTION

The scanner freezes when an ICD halt occurs, and remains frozen until user-mode operation resumes. The debugger may inspect the SCANCON0 and SCANLADR registers to determine the state of the scan.

The ICD interaction with each operating mode is summarized in Table 11-3.

		Scanner Operating Mode	
ICD Halt	Peek	Concurrent Triggered	Burst
External Halt		If external halt is asserted during a scan cycle, the instruction (delayed by scan) may or may not execute before ICD entry, depending on external halt timing.	If external halt is asserted during the BSF(SCANCON.GO), ICD entry occurs, and the burst is delayed until ICD exit. Otherwise, the current NVM-access cycle will complete, and then the scanner will be interrupted for ICD entry.
		If external halt is asserted during the burst, the burst is suspended and wil resume with ICD exit.	
PC Breakpoint	If Scanner would peek an instruction that is not executed (because of ICD entry), the peek will occur after ICD	Scan cycle occurs before ICD entry and instruction execution happens after the ICD exits.	If PCPB (or single step) is on
Data Breakpoint	exit, when the instruction executes.	The instruction with the dataBP executes and ICD entry occurs immediately after. If scan is requested during that cycle, the scan cycle is postponed until the ICD exits.	BSF (SCANCON.GO), the ICD is entered before execution; execution of the burst will occur at ICD exit, and the burst will run to completion.
Single Step		If a scan cycle is ready after the debug instruction is executed, the scan will read PFM and then the ICD is re-entered.	Note that the burst can be interrupted by an external halt.
SWBP and ICDINST		If scan would stall a SWBP, the scan cycle occurs and the ICD is entered.	If SWBP replaces BSF(SCANCON.GO), the ICD will be entered; instruction execution will occur at ICD exit (from ICDINSTR register), and the burst will run to completion.

TABLE 11-3: ICD AND SCANNER INTERACTIONS

11.11 Register Definitions: CRC and Scanner Control

R/W-0/0	R/W-0/0	R-0	R/W-0/0	U-0	U-0	R/W-0/0	R-0			
EN	CRCGO	BUSY	ACCM	_	_	SHIFTM	FULL			
bit 7							bit C			
Legend:										
R = Readable	e bit	W = Writable	bit	U = Unimpler	nented bit, rea	d as '0'				
u = Bit is unc	hanged	x = Bit is unk	nown	-n/n = Value a	at POR and BC	OR/Value at all o	other Resets			
'1' = Bit is set '0' = Bit is cleared										
bit 7	EN: CRC En	abla bit								
oit 7		able bit Jule is released from Reset								
		CRC is disabled and consumes no operating current								
bit 6	CRCGO: CR			g						
	1 = Start CR	tart CRC serial shifter								
	0 = CRC seri	ial shifter turne	d off							
bit 5	BUSY: CRC	Busy bit								
		n progress or p pits in shifter ha		ed into accumula	ator and EMP1	T Y = 1				
bit 4	ACCM: Accu	imulator Mode	bit							
		ugmented with								
		ot augmented								
bit 3-2	Unimplemer	nted: Read as	'0'							
bit 1	SHIFTM: Shi									
	1 = Shift righ 0 = Shift left									
bit 0	FULL: Data I	Path Full Indica	ator bit							
	1 = CRCDAT	H/L registers a	are full							
	0 = CRCDAT	H/L registers h	ave shifted the	air data into the	chiftor					

REGISTER 11-1: CRCCON0: CRC CONTROL REGISTER 0

REGISTER 11-2: CRCCON1: CRC CONTROL REGISTER 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	R/W-0/0	
	DLEN	<3:0>		PLEN<3:0>				
bit 7							bit 0	

Legend:							
		W = Writable bit	U = Unimplemented bit, read as '0'				
		x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets				
'1' = Bit is :	1' = Bit is set '0' = Bit is cleared						
bit 7-4	DLEN<3:0	D>: Data Length bits					
	Denotes t	Denotes the length of the data word -1 (See Example 11-1)					
bit 3-0	PLEN<3:0	>: Polynomial Length bits					
	Denotes t	he length of the polynomial -	1 (See Example 11-1)				

REGISTER 11-3: CRCDATH: CRC DATA HIGH BYTE REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
			DAT<	:15:8>			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimpleme	nted bit, read as '0	,	
u = Bit is unchanged		x = Bit is unknown		-n/n = Value at F	OR and BOR/Valu	ue at all other Rese	ts
'1' = Bit is set		'0' = Bit is cleared					

bit 7-0 DAT<15:8>: CRC Input/Output Data bits

REGISTER 11-4: CRCDATL: CRC DATA LOW BYTE REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
			DAT	<7:0>			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimpleme	nted bit, read as '0	3	
u = Bit is unchanged	b	x = Bit is unknowr	n	-n/n = Value at F	POR and BOR/Valu	ue at all other Rese	ets

bit 7-0

'1' = Bit is set

DAT<7:0>: CRC Input/Output Data bits Writing to this register fills the shifter.

'0' = Bit is cleared

REGISTER 11-5: CRCACCH: CRC ACCUMULATOR HIGH 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
			ACC	<15:8>			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimpleme	nted bit, read as '0	,	
u = Bit is unchang	ed	x = Bit is unknown		-n/n = Value at F	POR and BOR/Val	ue at all other Res	ets
'1' = Bit is set		'0' = Bit is cleared					

bit 7-0 ACC<15:8>: CRC Accumulator Register bits

Writing to this register writes to the CRC accumulator register. Reading from this register reads the CRC accumulator.

REGISTER 11-6: CRCACCL: CRC ACCUMULATOR 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
			ACC	<7:0>			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimplemer	nted bit, read as '0	,	
u = Bit is unchang	ed	x = Bit is unknown		-n/n = Value at F	OR and BOR/Val	ue at all other Res	ets
'1' = Bit is set		'0' = Bit is cleared					

bit 7-0

ACC<7:0>: CRC Accumulator Register bits Writing to this register writes to the CRC accumulator register through the CRC write bus. Reading from this register reads the CRC accumulator.

REGISTER 11-7: CRCSHIFTH: CRC SHIFT HIGH BYTE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
			SHIF	T<15:8>			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimpleme	nted bit, read as '0	,	
u = Bit is unchange	ed	x = Bit is unknown		-n/n = Value at F	POR and BOR/Val	ue at all other Resets	3
'1' = Bit is set		'0' = Bit is cleared					

bit 7-0 SHIFT<15:8>: CRC Shifter Register bits Reading from this register reads the CRC Shifter.

REGISTER 11-8: CRCSHIFTL: CRC SHIFT LOW BYTE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
			SHIF	T<7:0>			
bit 7							bit 0
Legend:							
R = Readable b	it	W = Writable bit		U = Unimplemer	nted bit, read as '0	,	
u = Bit is uncha	nged	x = Bit is unknown		-n/n = Value at F	OR and BOR/Valu	ue at all other Res	ets
'1' = Bit is set '0' = Bit is cleared							

Reading from this register reads the CRC Shifter.

REGISTER 11-9: CRCXORH: CRC XOR HIGH BYTE REGISTER

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			XOR	<15:8>			
bit 7							bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimplemer	nted bit, read as '0		

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 XOR<15:8>: XOR of Polynomial Term X_N Enable bits

REGISTER 11-10: CRCXORL: CRC XOR LOW BYTE REGISTER

x = Bit is unknown

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-0
			XOR<7:1>				_
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	

bit 7-1 XOR<7:1>: XOR of Polynomial Term XN Enable bits

bit 0 Unimplemented: Read as '0'

u = Bit is unchanged

R/W-0/0		R-0	R-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
EN ⁽¹⁾	SCANGO ^(2, 3)	BUSY ⁽⁴⁾	INVALID	INTM		MODE	<1:0> (5)
bit 7							bit 0
Legend:							
R = Readal	ole hit	W = Writable	hit	II = I Inimpler	mented bit, read	l as 'O'	
u = Bit is ur		x = Bit is unk		•	at POR and BO		other Resets
'1' = Bit is s	0	'0' = Bit is cle			eared by hardw		
bit 7	EN: Scanner	Enable bit ⁽¹⁾					
	1 = Scanner is						
			ernal states are	e reset			
bit 6	SCANGO: Sc						
		ent peripheral.	i ready signal,	NVM WIII be ad	ccessed accord	ing to MDx an	d data passed
	0 = Scanner o		not occur				
bit 5	BUSY: Scann	•					
	1 = Scanner o	•					
			ete (or never st	arted)			
bit 4	INVALID: Sca	-					
			remented or co o a valid addre	ontains an inva ess	lid address ⁽⁶⁾		
bit 3	INTM: NVM S	canner Interru	ipt Manageme	nt Mode Select	t bit		
	If MODE = 10:						
	This bit is igno						
			<u>d until all data is</u>				
		is overridden	(to zero) during	g interrupt oper	ration; scanner	resumes after	returning from
	interrupt 0 = SCANGO	is not affected	d bv interrupts.	the interrupt re	esponse will be	affected	
	If MODE = 00		,				
	1 = SCANGO from inter		(to zero) during	interrupt opera	ation; scan oper	ations resume	after returning
			t NVM access				
bit 2	Unimplemen	ted: Read as	0'				
bit 1-0	MODE<1:0>:	Memory Acce	ess Mode bits ⁽⁵)			
	11 = Triggere	d mode					
	10 = Peek mo						
	01 = Burst mo 00 = Concurre						
Note 1:	Setting EN = 0 (SC	CANCON0 red	ister) does not	affect any othe	er register conte	ent.	
	This bit is cleared	-		-	-		
	f INTM = 1, this bi			-		response.	
	BUSY = 1 when th				-	-	
	See Table 11-1 for		-			-	
I	An invalid address memory is 0x4000						

REGISTER 11-11: SCANCONO: SCANNER ACCESS CONTROL REGISTER 0

address is invalid.

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
			LADR<	15:8> (1, 2)			
bit 7							bit 0
Legend:							
R = Readable I	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 Re			ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				

bit 7-0 LADR<15:8>: Scan Start/Current Address bits^(1, 2) Most Significant bits of the current address to be fetched from, value increments on each fetch of memory.

- **Note 1:** Registers SCANLADRH/L form a 16-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
 - 2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 11-13: SCANLADRL: SCAN LOW 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
	LADR<7:0> ^(1, 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 7-0 LADR<7:0>: Scan Start/Current Address bits^(1, 2)

Least Significant bits of the current address to be fetched from, value increments on each fetch of memory

- **Note 1:** Registers SCANLADRH/L form a 16-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
 - 2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

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
	HADR<15:8> ^(1, 2)						
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable bit		U = Unimpler	nented bit, read	l as '0'	
u = Bit is unch	anged	x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other Re			ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				

REGISTER 11-14: SCANHADRH: SCAN HIGH ADDRESS HIGH BYTE REGISTER

bit 7-0 HADR<15:8>: Scan End Address bits^(1, 2)

Most Significant bits of the address at the end of the designated scan

Note 1: Registers SCANHADRH/L form a 16-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).

2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 11-15: SCANHADRL: SCAN HIGH 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
			HADR<	7:0> (1, 2)			
bit 7							bit 0
Legend:							
P - Poodablo k	sit	M = M/ritable bi	+	II – I Inimpion	nontod hit road	ac 'O'	

•		
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 HADR<7:0>: Scan End Address bits^(1, 2)

Least Significant bits of the address at the end of the designated scan

- **Note 1:** Registers SCANHADRH/L form a 16-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SCANGO = 0 (SCANCON0 register).
 - 2: While SCANGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 11-16: SCANTRIG: SCAN TRIGGER SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	
—	—	—	—		TSEI	_<3:0>		
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 a	at POR and BO	DR/Value at all o	ther Resets	
'1' = Bit is set		'0' = Bit is clea	ared					
bit 7-4	Unimplemer	ted: Read as '	0'					
hit 3-0	nit 3-0 TSEL <3:0- Scanner Data Trigger Input Selection hits							

bit 3-0	TSEL<3:0>: Scanner Data Trigger Input Selection bits
	1111-1010 = Reserved
	1001 = SMT2_Match
	1000 = SMT1_Match
	0111 = TMR0_Overflow
	0110 = TMR5_Overflow
	0101 = TMR3_Overflow
	0100 = TMR1_Overflow
	0011 = TMR6_postscaled
	0010 = TMR4_postscaled
	0001 = TMR2_postscaled

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH CRC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page	
CRCACCH		•	•	ACC<	15:8>		·		142	
CRCACCL		ACC<7:0>								
CRCCON0	EN	CRCGO	BUSY	ACCM		—	SHIFTM	FULL	141	
CRCCON1		DLEN<3:0> PLEN<3:0>								
CRCDATH				DAT<	15:8>				142	
CRCDATL		DAT<7:0>								
CRCSHIFTH		SHIFT<15:8>							143	
CRCSHIFTL	SHIFT<7:0>								143	
CRCXORH		XOR<15:8>								
CRCXORL				XOR<7:1>				—	143	
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97	
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	106	
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	101	
SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<	1:0>	144	
SCANHADRH				HADR<	:15:8>				146	
SCANHADRL				HADR	<7:0>				146	
SCANLADRH				LADR<	:15:8>				145	
SCANLADRL				LADR	<7:0>				145	
SCANTRIG						TSEL	<3:0>		147	

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

* Page provides register information.

PIC16(L)F1615/9

12.0 I/O PORTS

Each port has six 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)
- INLVLx (input level control)
- ODCONx registers (open-drain)
- SLRCONx registers (slew rate)

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

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

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.

TABLE 12-1: PORT AVAILABILITY PER DEVICE

Device	PORTA	PORTB	PORTC
PIC16(L)F1619	•	•	•
PIC16(L)F1615	•		•

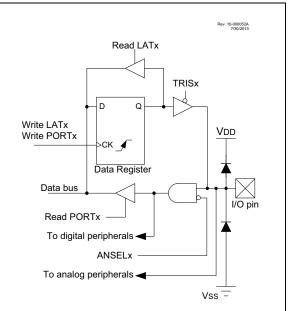
The Data Latch (LATx registers) is useful for readmodify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx 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 12-1.

FIGURE 12-1: GEN

GENERIC I/O PORT OPERATION



12.1 PORTA Registers

12.1.1 DATA REGISTER

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 12-2). 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 12-1 shows how to initialize an I/O port.

Reading the PORTA register (Register 12-1) 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).

12.1.2 DIRECTION CONTROL

The TRISA register (Register 12-2) 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'.

12.1.3 OPEN-DRAIN CONTROL

The ODCONA register (Register 12-6) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONA bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONA bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

12.1.4 SLEW RATE CONTROL

The SLRCONA register (Register 12-7) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONA bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONA bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

12.1.5 INPUT THRESHOLD CONTROL

The INLVLA register (Register 12-8) controls the input voltage threshold for each of the available PORTA input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTA register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See **35.3** "DC Characteristics" for more information on threshold levels.

Note:	Changing the input threshold selection
	should be performed while all peripheral
	modules are disabled. Changing the
	threshold level during the time a module is
	active may inadvertently generate a tran-
	sition associated with an input pin, regard-
	less of the actual voltage level on that pin.

12.1.6 ANALOG CONTROL

The ANSELA register (Register 12-4) 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.

EXAMPLE 12-1: INITIALIZING PORTA

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

12.1.7 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other functions are selected with the peripheral pin select logic. See **Section13.0** "**Peripheral Pin Select** (**PPS**) **Module**" for more information. Analog input functions, such as ADC inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELA register. Digital output functions may continue to control the pin when in Analog mode.

12.2 Register Definitions: PORTA

U-0	U-0	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x	
—	_	RA5	RA4	RA3	RA2	RA1	RA0	
bit 7					•		bit 0	
Legend:								
R = Readable b	oit	W = Writable	bit	U = Unimpler	mented bit, read	l as '0'		
u = Bit is unchanged x = Bit is unknown			nown	-n/n = Value at POR and BOR/Value at all other Resets				

REGISTER 12-1:	PORTA: PORTA REGISTER
----------------	-----------------------

bit 7-6	Unimplemented: Read as '0'
bit 5-0	RA<5:0> : PORTA I/O Value bits ⁽¹⁾ 1 = Port pin is <u>></u> VIH 0 = Port pin is <u><</u> VIL

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

REGISTER 12-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	(1)	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 7-6	Unimplemented: Read as '0'
bit 5-4	TRISA<5:4>: PORTA Tri-State Control bit 1 = PORTA pin configured as an input (tri-stated) 0 = PORTA pin configured as an output
bit 3	Unimplemented: Read as '1'
bit 2-0	TRISA<2:0>: PORTA Tri-State Control bit 1 = PORTA pin configured as an input (tri-stated) 0 = PORTA pin configured as an output

Note 1: Unimplemented, read as '1'.

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
—	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
						bit 0
bit	W = Writable	bit	U = Unimpler	mented 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			
	'0' = Bit is clea	ared				
		LATA5 Dit W = Writable anged x = Bit is unkr	LATA5 LATA4 Dit W = Writable bit	LATA5 LATA4 LATA3 Dit W = Writable bit U = Unimpler anged x = Bit is unknown -n/n = Value a	—LATA5LATA4LATA3LATA2DitW = Writable bitU = Unimplemented bit, read anged $x = Bit is unknown$ $-n/n = Value at POR and BOI$	—LATA5LATA4LATA3LATA2LATA1DitW = Writable bitU = Unimplemented bit, read as '0' anged $x = Bit$ is unknown $-n/n = Value at POR and BOR/Value at all complemented bit of the second secon$

REGISTER 12-3: LATA: PORTA DATA LATCH REGISTER

bit 7-6 Unimplemented: Read as '0'

bit 5-0 LATA<5:0>: RA<5:0> Output Latch Value bits⁽¹⁾

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

REGISTER 12-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	U-0	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	_	ANSA4	_	ANSA2	ANSA1	ANSA0
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 7-5	Unimplemented: Read as '0'
bit 4	 ANSA4: Analog Select between Analog or Digital Function on Pins RA4, respectively 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled. 0 = Digital I/O. Pin is assigned to port or digital special function.
bit 3	Unimplemented: Read as '0'
bit 2-0	 ANSA<2:0>: Analog Select between Analog or Digital Function on Pins RA<2:0>, respectively 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled. 0 = Digital I/O. Pin is assigned to port or digital special function.
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.

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
—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
bit 7							bit 0
Legend:							

REGISTER 12-5: WPUA: WEAK PULL-UP PORTA REGISTER

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	WPUA<5:0>: Weak Pull-up Register bits ⁽³⁾
	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 configured as an output.
- **3:** For the WPUA3 bit, when MCLRE = 1, weak pull-up is internally enabled, but not reported here.

REGISTER 12-6: ODCONA: PORTA OPEN-DRAIN CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0
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 5-4	ODA<5:4>: PORTA Open-Drain Enable bits For RA<5:4> pins, respectively 1 = Port pin operates as open-drain drive (sink current only) 0 = Port pin operates as standard push-pull drive (source and sink current)
bit 3	Unimplemented: Read as '0'
bit 2-0	ODA<2:0>: PORTA Open-Drain Enable bits For RA<2:0> pins, respectively 1 = Port pin operates as open-drain drive (sink current only) 0 = Port pin operates as standard push-pull drive (source and sink current)

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1	
_	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	
bit 7							bit 0	
Legend:								
R = Readab	ole bit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'		
u = Bit is un	ichanged	x = Bit is unkr	nown	-n/n = Value a	at POR and BOI	R/Value at all c	other Resets	
'1' = Bit is s	et	'0' = Bit is clea	ared					
bit 7-6	Unimplemen	ted: Read as '	0'					
bit 5-4	SLRA<5:4>:	PORTA Slew F	Rate Enable bi	its				
		pins, respectiv	•					
		lew rate is limit						
	0 = Port pin slews at maximum rate							
bit 3	Unimplemented: Read as '0'							
bit 2-0	SLRA<2:0>:	PORTA Slew F	Rate Enable bi	its				
For RA<2:0> pins, respectively								
		t pin slew rate is limited						
	0 = Port pin s	lews at maxim	um rate					

REGISTER 12-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

REGISTER 12-8: INLVLA: PORTA INPUT LEVEL CONTROL 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
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
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 INLVLA<5:0>: PORTA Input Level Select bits

For RA<5:0> pins, respectively

1 = ST input used for PORT reads and interrupt-on-change

0 = TTL input used for PORT reads and interrupt-on-change

TABLE 12-2:	SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	_	_	—	ANSA4		ANSA2	ANSA1	ANSA0	152
INLVLA	—	_	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	154
LATA	_	_	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	152
ODCONA	_	_	ODA5	ODA4	_	ODA2	ODA1	ODA0	153
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		223
PORTA	_	_	RA5	RA4	RA3	RA2	RA1	RA0	151
SLRCONA	_	_	SLRA5	SLRA4	_	SLRA2	SLRA1	SLRA0	154
TRISA	_	_	TRISA5	TRISA4	_(1)	TRISA2	TRISA1	TRISA0	151
WPUA	_	_	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	153

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTA.**Note 1:**Unimplemented, read as '1'.

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		—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	<u> </u>
CONFIG1	7:0	CP	MCLRE	PWRTE	_	_	FOSC<2:0>		69	

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

12.3 PORTB Registers (PIC16(L)F1619 Only)

12.3.1 DATA REGISTER

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 12-10). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., disable the output driver). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 12-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 12-9) 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).

12.3.2 DIRECTION CONTROL

The TRISB register (Register 12-10) 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'.

12.3.3 OPEN-DRAIN CONTROL

The ODCONB register (Register 12-14) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONB bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONB bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

12.3.4 SLEW RATE CONTROL

The SLRCONB register (Register 12-15) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONB bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONB bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

12.3.5 INPUT THRESHOLD CONTROL

The INLVLB register (Register 12-16) controls the input voltage threshold for each of the available PORTB input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTB register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See **35.3 "DC Characteristics"** for more information on threshold levels.

Note: Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

12.3.6 ANALOG CONTROL

The ANSELB register (Register 12-12) 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 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 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.

12.3.7 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other functions are selected with the peripheral pin select logic. See **Section13.0** "**Peripheral Pin Select** (**PPS**) **Module**" for more information. Analog input functions, such as ADC inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELB register. Digital output functions continue to may continue to control the pin when it is in Analog mode.

12.4 Register Definitions: PORTB

REGISTER 12-9: PORTB: PORTB REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	_	—
bit 7							bit 0
Legend:							
R = Readable	W = Writable	bit	U = Unimpler	mented bit, read	d as '0'		
u = Bit is uncha	x = Bit is unkr	nown	-n/n = Value at POR and BOR/Value at all other Resets				
'1' = Bit is set		'0' = Bit is clea	ared				

RB<7:4>: PORTB I/O Value bits ⁽¹⁾
1 = Port pin is <u>></u> Vін
0 = Port pin is <u><</u> VIL

bit 3-0 Unimplemented: Read as '0'

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

REGISTER 12-10: TRISB: PORTB TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
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	TRISB<7:4>: PORTB Tri-State Control bits
	1 = PORTB pin configured as an input (tri-stated)
	0 = PORTB pin configured as an output
bit 3-0	Unimplemented: Read as '0'

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	_
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	LATB<7:4>: RB<7:4> Output Latch Value bits ⁽¹⁾
---------	---

bit 3-0 Unimplemented: Read as '0'

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

REGISTER 12-12: ANSELB: PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
—	—	ANSB5	ANSB4	—	—	—	
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 5-4 **ANSB<5:4>**: Analog Select between Analog or Digital Function on Pins RB<5:4>, respectively 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.

analog input: Pin is assigned as analog input⁽¹⁾. Digital input buffer disact
 0 = Digital I/O. Pin is assigned to port or digital special function.

bit 3-0 Unimplemented: Read as '0'

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.

-n/n = Value at POR and BOR/Value at all other Resets

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit				U = Unimpler	mented bit, read	as '0'	

REGISTER 12-13: WPUB: WEAK PULL-UP PORTB REGISTER^{(1),(2)}

x = Bit is unknown

'0' = Bit is cleared

bit 7-4	WPUB<7:4>: Weak Pull-up Register bits
	1 = Pull-up enabled
	0 = Pull-up disabled

bit 3-0 Unimplemented: Read as '0'

u = Bit is unchanged

'1' = Bit is set

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 configured as an output.

REGISTER 12-14: ODCONB: PORTB OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODB7	ODB6	ODB5	ODB4	—	—	—	—
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	ODB<7:4>: PORTB Open-Drain Enable bits
	For RB<7:4> pins, respectively
	1 = Port pin operates as open-drain drive (sink current only)
	0 = Port pin operates as standard push-pull drive (source and sink current)
bit 3-0	Unimplemented: Read as '0'

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0	
SLRB7	SLRB6	SLRB5	SLRB4	_	—	—	—	
bit 7						bit 0		
Legend:								
R = Readable bit W = Write		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 7-4	For RB<7:4> 1 = Port pin s 0 = Port pin s	PORTA Slew F pins, respective lew rate is limit lews at maximu	ely ed um rate	its				
bit 3-0	Unimplemented: Read as '0'							

REGISTER 12-15: SLRCONB: PORTB SLEW RATE CONTROL REGISTER

REGISTER 12-16: INLVLB: PORTB INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—
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 INLVLB<7:4>: PORTB Input Level Select bits For RB<7:4> pins, respectively 1 = ST input used for PORT reads and interrupt-on-change 0 = TTL input used for PORT reads and interrupt-on-change bit 3-0 Unimplemented: Read as '0'

TABLE 12-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	_	—	ANSB5	ANSB4	_	—	_	—	159
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	_	_	_	_	161
LATB	LATB7	LATB6	LATB5	LATB4	_	_	_	_	159
ODCONB	ODB7	ODB6	ODB5	ODB4	_	_	_	_	160
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		223
PORTB	RB7	RB6	RB5	RB4	_	_	_	_	158
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	_	_	_	_	161
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	_	_	_	-	158
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	_	—	—	160

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

Note 1: Unimplemented, read as '1'.

PIC16(L)F1615/9

	υ.									
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	_	_	FCMEN	IESO	CLKOUTEN	BORE	N<1:0>	—	69
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>			FOSC<2:0>		

TABLE 12-5: SUMMARY OF CONFIGURATION WORD WITH PORTB

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

12.5 PORTC Registers

12.5.1 DATA REGISTER

PORTC is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 12-18). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). 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 12-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 12-17) 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).

12.5.2 DIRECTION CONTROL

The TRISC register (Register 12-18) 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'.

12.5.3 OPEN-DRAIN CONTROL

The ODCONC register (Register 12-22) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONC bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONC bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

12.5.4 SLEW RATE CONTROL

The SLRCONC register (Register 12-23) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONC bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONC bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

12.5.5 INPUT THRESHOLD CONTROL

The INLVLC register (Register 12-24) controls the input voltage threshold for each of the available PORTC input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTC register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See **35.3** "DC Characteristics" for more information on threshold levels.

Note:	Changing the input threshold selection
	should be performed while all peripheral
	modules are disabled. Changing the
	threshold level during the time a module is
	active may inadvertently generate a tran-
	sition associated with an input pin, regard-
	less of the actual voltage level on that pin.

12.5.6 ANALOG CONTROL

The ANSELC register (Register 12-20) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC 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 ANSELC bits has no effect on digital output functions. A pin with TRIS clear and ANSELC 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 ANSELC 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.

12.5.7 HIGH DRIVE STRENGTH PINS

The HIDRVC register (Register 12-25) controls the high drive options on the RC4 and RC5. When a HIDRVC bit is cleared, the pin has normal drive strengths. When a HIDRVC bit is set, its respective pin can sink or source currents up to 100mA.

12.5.8 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other functions are selected with the peripheral pin select logic. See **Section13.0** "**Peripheral Pin Select** (**PPS**) **Module**" for more information. Analog input functions, such as ADC inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELC register. Digital output functions continue to may continue to control the pin when it is in Analog mode.

12.6 Register Definitions: PORTC

REGISTER 12-17: PORTC: PORTC 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
RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0
bit 7		·		•			bit 0
Legend:							
R = Readable bit W = Writable bit		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		ared					

bit 7-0	RC<7:0>: PORTC I/O Value bits ^(1, 2)
	1 = Port pin is <u>></u> Vін
	0 = Port pin is <u><</u> Vı∟

Note 1: RC<7:6> on PIC16(L)F1619 only.

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

REGISTER 12-18: TRISC: PORTC TRI-STATE REGISTER

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
TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	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⁾
	1 = PORTC pin configured as an input (tri-stated)
	0 = PORTC pin configured as an output

Note 1: TRISC<7:6> on PIC16(L)F1619 only.

REGISTER 12-19: LATC: PORTC DATA LATCH 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
LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	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>: RC<7:0> Output Latch Value bits ⁽¹⁾
	1 = PORTC pin configured as an input (tri-stated)
	0 = PORTC pin configured as an output

Note 1: LATC<7:6> on PIC16(L)F1619 only.

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

REGISTER 12-20: ANSELC: PORTC ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0
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 **ANSC<7:6>**: Analog Select between Analog or Digital Function on Pins RC<7:6>, respectively⁽¹⁾ 1 = Analog input. Pin is assigned as analog input⁽²⁾. Digital input buffer disabled.

- 0 = Digital I/O. Pin is assigned to port or digital special function.
- bit 5-4 Unimplemented: Read as '0'

bit 3-0 **ANSC<3:0>**: Analog Select between Analog or Digital Function on Pins RC<3:0>, respectively 1 = Analog input. Pin is assigned as analog input⁽²⁾. Digital input buffer disabled. 0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: ANSC<7:6> on PIC16(L)F1619 only.

2: 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.

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	
WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	
bit 7							bit 0	
Legend:								
R = Readable	= Readable bit W = Writable bit U = Un		U = Unimplemented bit, read as '0'					
u = Bit is uncha	anged	x = Bit is unkr	s unknown -n/n = Value at POR and BOR/Value at all other Re			other Resets		

REGISTER 12-21: WPUC: WEAK PULL-UP PORTC REGISTER^{(2),(3)}

'0' = Bit is cleared

bit 7-0 WPUC<7:0>: Weak Pull-up Register bits⁽¹⁾

1 = Pull-up enabled

0 = Pull-up disabled

Note 1: WPUC<7:6> on PIC16(L)F1619 only.

'1' = Bit is set

- 2: Global WPUEN bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
- 3: The weak pull-up device is automatically disabled if the pin is configured as an output.

REGISTER 12-22: ODCONC: PORTC OPEN-DRAIN CONTROL 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
ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0
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 **ODC<7:0>:** PORTC Open-Drain Enable bits⁽¹⁾

For RC<7:0> pins, respectively

- 1 = Port pin operates as open-drain drive (sink current only)
- 0 = Port pin operates as standard push-pull drive (source and sink current)

Note 1: ODC<7:6> on PIC16(L)F1619 only.

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
SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
bit 7						•	bit 0
Legend:							
R = Readable I	oit	W = Writable	bit	U = Unimpler	nented bit, read	as '0'	
u = Bit is uncha	anged	x = Bit is unkr	iown	-n/n = Value at POR and BOR/Value at all other Reset			
'1' = Bit is set		'0' = Bit is clea	ared				

REGISTER 12-23: SLRCONC: PORTC SLEW RATE CONTROL REGISTER

bit 7-0 SLRC<7:0>: PORTC Slew Rate Enable bits⁽¹⁾ For RC<7:0> pins, respectively 1 = Port pin slew rate is limited 0 = Port pin slews at maximum rate

Note 1: SLRC<7:6> on PIC16(L)F1619 only.

REGISTER 12-24: INLVLC: PORTC INPUT LEVEL CONTROL REGISTER

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
INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
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 INLVLC<7:0>: PORTC Input Level Select bits⁽¹⁾

For RC<7:0> pins, respectively

- 1 = ST input used for PORT reads and interrupt-on-change
- 0 = TTL input used for PORT reads and interrupt-on-change

Note 1: INLVLC<7:6> on PIC16(L)F1619 only.

REGISTER 12-25: HIDRVC: PORTC HIGH DRIVE STRENGTH REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	HIDC5	HIDC4	—	—	—	—
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	as '0'	
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all o	ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7-6	Unimplemen	ted: Read as '	0'				
bit 5	HIDC5: High	Current Drive E	Enable on Port	t C5			
	1 = High Drive enabled						
	0 = High Drive	0 = High Drive disabled					
bit 4	HIDC4: High Current Drive Enable on Port C4						
	1 = High Drive						
	0 = High Drive	e disabled					

bit 3-0 Unimplemented: Read as '0'

TABLE 12-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	166
HIDRVC		_	HIDC5	HIDC4	_	_	_	_	169
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	168
LATC	LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	166
ODCONC	ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	167
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	165
SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	168
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	167

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTC. **Note 1:** PIC16(L)F1619 only

13.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 13-1.

13.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Multiple peripherals can operate from the same source simultaneously. Port reads always return the pin level regardless of peripheral PPS selection. If a pin also has associated analog functions, the ANSEL bit for that pin must be cleared to enable the digital input buffer.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 13-1.

Note: The notation "xxx" in the register name is a place holder for the peripheral identifier. For example, CLC1PPS.

13.2 PPS Outputs

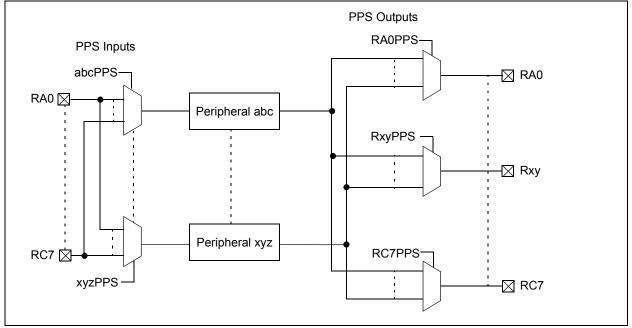
Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- · EUSART (synchronous operation)
- MSSP (I²C)
- · CWG (auto-shutdown)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 13-2.

Note: The notation "Rxy" is a place holder for the pin identifier. For example, RA0PPS.

FIGURE 13-1: SIMPLIFIED PPS BLOCK DIAGRAM



13.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- EUSART (synchronous operation)
- MSSP (I²C)

Note: The I²C default input pins are I²C and SMBus compatible and are the only pins on the device with this compatibility.

13.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 13-1.

EXAMPLE 13-1: PPS LOCK/UNLOCK SEQUENCE

; suspend interrupts
bcf INTCON,GIE
; BANKSEL PPSLOCK ; set bank
; required sequence, next 5 instructions
movlw 0x55
movwf PPSLOCK
movlw 0xAA
movwf PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
bsf PPSLOCK, PPSLOCKED
; restore interrupts
bsf INTCON,GIE

13.5 PPS Permanent Lock

The PPS can be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

13.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

13.7 Effects of a Reset

A device Power-On-Reset (POR) clears all PPS input and output selections to their default values. All other Resets leave the selections unchanged. Default input selections are shown in Table 13-1.

13.8 Register Definitions: PPS Input Selection

REGISTER 13-1:	XXXPPS: PERIPHERAL XXX INPUT SELECTION
----------------	---

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u		
	_	—	xxxPPS<4:0>						
bit 7							bit 0		
Legend:									
R = Readable bit W = Writable bit			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 cle	ared	q = value depends on peripheral						
bit 7-5	Unimplement	ted: Read as '	0'						
bit 4-3	<pre>xxxPPS<4:3>: Peripheral xxx Input PORT Selection bits 11 = Reserved. Do not use. 10 = Peripheral input is PORTC 01 = Peripheral input is PORTB (PIC16(L)F1619 only) 00 = Peripheral input is PORTA</pre>								
bit 2-0	<pre>xxxPPS<2:0>: Peripheral xxx Input Bit Selection bits ⁽¹⁾ 111 = Peripheral input is from PORTx Bit 7 (Rx7) 110 = Peripheral input is from PORTx Bit 6 (Rx6) 101 = Peripheral input is from PORTx Bit 5 (Rx5) 100 = Peripheral input is from PORTx Bit 4 (Rx4) 011 = Peripheral input is from PORTx Bit 3 (Rx3) 010 = Peripheral input is from PORTx Bit 2 (Rx2) 001 = Peripheral input is from PORTx Bit 1 (Rx1) 000 = Peripheral input is from PORTx Bit 0 (Rx0)</pre>								

Note 1: See Table 13-1 for Reset values.

REGISTER 13-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u		
—	—	—	RxyPPS<4:0>						
bit 7	•						bit 0		
Legend:									
R = Readable 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 clea	ared						

bit 7-5 Unimplemented: Read as '0'

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits Selection code determines the output signal on the port pin. See Table 13-2 for the selection codes

REGISTER 13-3: PPSLOCK: PPS LOCK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
_		—	_	—	_	_	PPSLOCKED
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		bit	U = Unimplemented bit, read as '0'				
u = Bit is unch	it 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 0	PPSLOCKED: PPS Locked bit
-------	---------------------------

1 = PPS is locked. PPS selections can not be changed.

0 = PPS is not locked. PPS selections can be changed.

Devinhevel	xxxPPS	Default Pi	n Selection	Reset Value (Reset Value (xxxPPS<4:0>)		
Peripheral	Register	PIC16(L)F1619	PIC16(L)F1615	PIC16(L)F1619	PIC16(L)F1615		
Interrupt on change	INTPPS	RA2	RA2	00010	00010		
Timer 0 clock	T0CKIPPS	RA2	RA2	00010	00010		
Timer 1 clock	T1CKIPPS	RA5	RA5	00101	00101		
Timer 1 gate	T1GPPS	RA4	RA4	00100	00100		
Timer 2 clock	T2CKIPPS	RA5	RA5	0101	0101		
Timer 3 clock	T3CKIPPS	RC5	RC5	10101	10101		
Timer 3 gate	T3GPPS	RC4	RC4	10100	10100		
Timer 4 clock	T4CKIPPS	RC1	RC1	10001	10001		
Timer 5 clock	T5CKIPPS	RC0	RC0	10000	10000		
Timer 5 gate	T5GPPS	RC3	RC3	10011	10011		
Timer 6 clock	T6CKIPPS	RA3	RA3	00011	00011		
CCP1	CCP1PPS	RC5	RC5	10101	10101		
CCP2	CCP2PPS	RC3	RC3	10011	10011		
CWG1	CWG1INPPS	RA2	RA2	00010	00010		
SPI and I ² C clock	SSPCLKPPS	RB6	RC0	01110	10000		
SPI and I ² C data	SSPDATPPS	RB4	RC1	01100	10001		
SPI slave select	SSPSSPPS	RC6	RC3	10110	10011		
EUSART RX	RXPPS	RB5	RC5	01101	10101		
EUSART CK	CKPPS	RB7	RC4	01111	10100		
All CLCs	CLCIN0PPS	RC3	RC3	10011	10011		
All CLCs	CLCIN1PPS	RC4	RC4	10100	10100		
All CLCs	CLCIN2PPS	RC1	RC1	10001	10001		
All CLCs	CLCIN3PPS	RA5	RA5	00101	00101		
SMT1 Window Input	SMTWIN1PPS	RA5	RA5	00101	00101		
SMT1 Signal Input	SMTSIG1PPS	RA4	RA4	00100	00100		
SMT2 Window Input	SMTWIN2PPS	RA3	RA3	00101	00101		
SMT2 Signal Input	SMTSIG2PPS	RC1	RC1	10001	10001		
Angular Timer 1 Clock Input	AT1INPPS	RC5	RC5	10101	10101		
Angular Timer 1 CC1 Input	AT1CC1PPS	RC3	RC3	10011	10011		
Angular Timer 1 CC2 Input	AT1CC2PPS	RC4	RC4	10100	10100		
Angular Timer 1 CC3 Input	AT1CC3PPS	RC5	RC5	10101	10101		

 TABLE 13-1:
 PPS INPUT REGISTER RESET VALUES

Example: CCP1PPS = 0x13 selects RC3 as the CCP1 input.

D DDO - 4-0	Outrast Olamal		PIC16(L)F161	9	PIC16(L)F1615		
RxyPPS<4:0>	Output Signal	PORTA	PORTB	PORTC	PORTA	PORTO	
llxxx	Reserved	•	•	٠	•	•	
10111	Reserved	•	•	•	•	•	
10110	Reserved	•	•	•	•	•	
10101	Reserved	•	•	•	•	•	
10100	Reserved	•	•	•	•	•	
10011	DT	•	•	•	•	•	
10010	TX/CK	•	•	•	•	•	
10001	SDO/SDA ⁽¹⁾	•	•	•	•	•	
10000	SCK/SCL ⁽¹⁾	•	•	•	•	•	
01111	PWM4_out	•	•	•	•	•	
01110	PWM3_out	•	•	•	•	•	
01101	CCP2_out	•	•	•	•	•	
01100	CCP1_out	•	•	•	•	•	
01011	CWG1OUTD ⁽¹⁾	•	•	•	•	•	
01010	CWG1OUTC ⁽¹⁾	•	•	•	•	•	
01001	CWG1OUTB ⁽¹⁾	•	•	•	•	•	
01000	CWG1OUTA ⁽¹⁾	•	•	•	•	•	
00111	LC4_out	•	•	•	•	•	
00110	LC3_out	•	•	•	•	•	
00101	LC2_out	•	•	•	•	•	
00100	LC1_out	•	•	•	•	•	
00011	ZCD1_out	•	•	•	•	•	
00010	sync_C2OUT	•	•	•	•	•	
00001	sync_C1OUT	•	•	•	•	•	
00000	LATxy	•	•	•	•	•	

TABLE 13-2: AV	AILABLE PORTS FOR OUTPUT BY PERIPHERAL ⁽²⁾
----------------	---

Note 1: TRIS control is overridden by the peripheral as required.

2: Unsupported peripherals will output a '0'.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page	
PPSLOCK	—		—	—	—	—	—	PPSLOCKED	173	
INTPPS		_	_		INTPPS<4:0>					
TOCKIPPS		_	_		T0CKIPPS<4:0>					
T1CKIPPS			_			T1CKIPPS<	4:0>		172	
T1GPPS		_	_			T1GPPS<4	l:0>		172	
T2CKIPPS	_	_	—			T2CKIPPS<	4:0>		172	
T3CKIPPS		_	_			T3CKIPPS<	4:0>		172	
T3GPPS	_	_	—			T3GPPS<4	l:0>		172	
T4CKIPPS	_	_	—			T4CKIPPS<	4:0>		172	
T5CKIPPS			_			T5CKIPPS<	4:0>		172	
T5GPPS	_	_	—			T5GPPS<4	l:0>		172	
T6CKIPPS		_	_			T6CKIPPS<	4:0>		172	
CCP1PPS		_	_			CCP1PPS<	4:0>		172	
CCP2PPS	_	_	—			CCP2PPS<	4:0>		172	
CWG1INPPS	_	_	—		C	WG1INPPS	<4:0>		172	
SSPCLKPPS	_	_	—		S	SPCLKPPS	<4:0>		172	
SSPDATPPS	_	_	_		S	SPDATPPS	<4:0>		172	
SSPSSPPS	_	_	_		:	SSPSSPPS	<4:0>		172	
RXPPS	_	_	_			RXPPS<4	:0>		172	
CKPPS			_	CKPPS<4:0>					172	
CLCIN0PPS	_	_	_	CLCIN0PPS<4:0>						
CLCIN1PPS	_	_	_		(CLCIN1PPS	<4:0>		172	
CLCIN2PPS	_	_	_		(CLCIN2PPS	<4:0>		172	
CLCIN3PPS	_	_	_		CLCIN3PPS<4:0>					
AT1INPPS	_	_	—			AT1INPPS<	4:0>		172	
ATCC1PPS	_	_	—	ATCC1PPS<4:0>					172	
ATCC2PPS		_	_			ATCC2PPS<	<4:0>		172	
ATCC3PPS			_			ATCC3PPS<	<4:0>		172	
SMT1SIGPPS		_	_		S	MT1SIGPPS	8<4:0>		172	
SMT1WINPPS			_		SI	MT1WINPPS	S<4:0>		172	
SMT2SIGPPS		_	_		S	MT2SIGPPS	8<4:0>		172	
SMT2WINPPS	_	_	—		SI	MT2WINPPS	S<4:0>		172	
RA0PPS		_	_	RA0PPS<4:0>					172	
RA1PPS		_	_	RA1PPS<4:0>						
RA2PPS	_	_	—	RA2PPS<4:0>						
RA4PPS	_	—	—			RA4PPS<4	k:0>		172	
RA5PPS	_	—	—			RA5PPS<4	k:0>		172	
RB4PPS ⁽¹⁾	_	_	—			RB4PPS<4	k:0>		172	
RB5PPS ⁽¹⁾	_	_	_			RB5PPS<4	l:0>		172	
RB6PPS ⁽¹⁾	_	_	_			RB6PPS<4	l:0>		172	
RB7PPS ⁽¹⁾		_	_			RB7PPS<4	l:0>		172	
RC0PPS	_	_	_			RC0PPS<4	l:0>		172	

Note 1: PIC16(L)F1619 only.

TABLE 13-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
RC1PPS	—	-	_		RC1PPS<4:0>				172
RC2PPS	—	_	_		RC2PPS<4:0>				172
RC3PPS	—	_	_		RC3PPS<4:0>				172
RC4PPS	—	_	_	RC4PPS<4:0>				172	
RC5PPS	_	_	_	RC5PPS<4:0>				172	
RC6PPS ⁽¹⁾	—	_	_	RC6PPS<4:0>				172	
RC7PPS ⁽¹⁾	—			RC7PPS<4:0>				172	

Note 1: PIC16(L)F1619 only.

14.0 INTERRUPT-ON-CHANGE

The PORTA, PORTB⁽¹⁾ and PORTC 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 port pin, or combination of port 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 14-1 is a block diagram of the IOC module.

Note 1: PORTB available on PIC16(L)F1619 only.

14.1 Enabling the Module

To allow individual port 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.

14.2 Individual Pin Configuration

For each port pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both associated bits of the IOCxP and IOCxN registers, respectively.

14.3 Interrupt Flags

The IOCAFx, IOCBFx and IOCCFx bits located in the IOCAF, IOCBF and IOCCF registers, respectively, are status flags that correspond to the interrupt-on-change pins of the associated port. 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 IOCAFx, IOCBFx and IOCCFx bits.

14.4 Clearing Interrupt Flags

The individual status flags, (IOCAFx, IOCBFx and IOCCFx 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 14-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

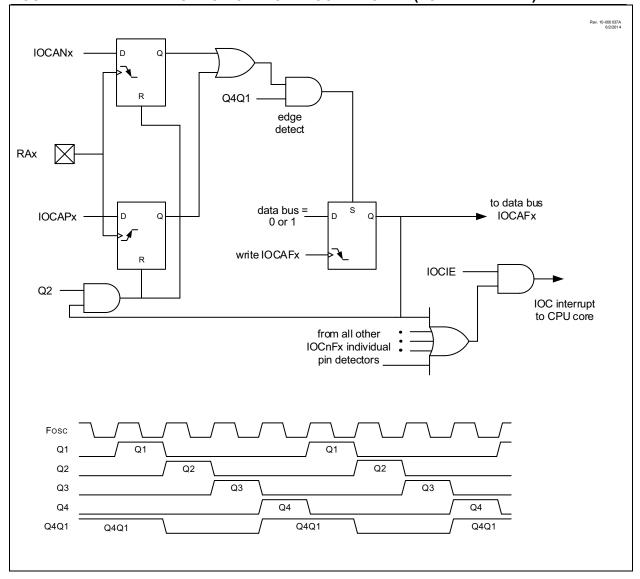
MOVLW 0xff XORWF IOCAF, W ANDWF IOCAF, F

14.5 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 IOCxF register will be updated prior to the first instruction executed out of Sleep.

FIGURE 14-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)



14.6 Register Definitions: Interrupt-on-Change Control

REGISTER 14-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

U-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
_	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
bit 7				-			bit 0
Legend:							
R = Readable bit W = Writable bit			U = Unimplemented bit, read as '0'				
u = Bit is unchan	s 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

bit 5-0

bit 5-0

IOCAP<5:0>: Interrupt-on-Change PORTA Positive Edge Enable bits

1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.

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

REGISTER 14-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

U-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
—		IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
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'

IOCAN<5:0>: Interrupt-on-Change PORTA Negative Edge Enable bits

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

REGISTER 14-3: IOCAF: INTERRUPT-ON-CHANGE PORTA FLAG REGISTER

U-0	U-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
—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
bit 7							bit 0

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

bit 7-6 Unimplemented: Read as '0'

IOCAF<5:0>: Interrupt-on-Change PORTA Flag bits

1 = An enabled change was detected on the associated pin.

Set when IOCAPx = 1 and a rising edge was detected on RAx, or when IOCANx = 1 and a falling edge was detected on RAx.

0 = No change was detected, or the user cleared the detected change.

REGISTER 14-4: IOCBP: INTERRUPT-ON-CHANGE PORTB POSITIVE EDGE REGISTER⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0		
IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—		
bit 7							bit (
Legend:									
R = Readable bit		W = Writable bit	t	U = Unimplemented bit, read as '0'					
u = Bit is unchan	ged	x = Bit is unknown		-n/n = Value at	POR and BOR/Va	ue at all other Re	esets		
'1' = Bit is set '0' = Bit is cleared									

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

bit 3-0 Unimplemented: Read as '0'

Note 1: PIC16(L)F1619 only.

REGISTER 14-5: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE REGISTER⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
bit 7							bit 0

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

bit 7-4 IOCBN<7:4>: Interrupt-on-Change PORTB Negative Edge Enable bits

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

bit 3-0 Unimplemented: Read as '0'

Note 1: PIC16(L)F1619 only.

REGISTER 14-6: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER⁽¹⁾

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	_	—	—	—
bit 7							bit 0

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

bit 7-4 **IOCBF<7:4>**: Interrupt-on-Change PORTB Flag bits 1 = An enabled change was detected on the associated pin.

- Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx.
- 0 = No change was detected, or the user cleared the detected change.

bit 3-0 Unimplemented: Read as '0'

Note 1: PIC16(L)F1619 only.

REGISTER 14-7: IOCCP: INTERRUPT-ON-CHANGE PORTC 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
IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
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	

IOCCP<7:0>: Interrupt-on-Change PORTC Positive Edge Enable bits(1)

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

Note 1: IOCCP<7:6> available on PIC16(L)F1619 only.

REGISTER 14-8: IOCCN: INTERRUPT-ON-CHANGE PORTC NEGATIVE EDGE REGISTER⁽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	R/W-0/0
IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
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

bit 7-0

IOCCN<7:0>: Interrupt-on-Change PORTC Negative Edge Enable bits(1)

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

Note 1: IOCCN<7:6> available on PIC16(L)F1619 only.

REGISTER 14-9: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER⁽¹⁾

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
IOCCF ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
bit 7							bit 0

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

bit 7-0

IOCCF<7:0>: Interrupt-on-Change PORTC Flag bits(1)

1 = An enabled change was detected on the associated pin.

- Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
- 0 = No change was detected, or the user cleared the detected change.

Note 1: IOCCF<7:6> available on PIC16(L)F1619 only.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	_	_		ANSA4	-	ANSA2	ANSA1	ANSA0	152
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	180
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	180
IOCAP	_	_	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	180
IOCBF ⁽²⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	_	_	_	_	181
IOCBN ⁽²⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	_	_	_	_	181
IOCBP ⁽²⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	_	_	_	_	181
IOCCF	IOCCF7 ⁽²⁾	IOCCF6 ⁽²⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	182
IOCCN	IOCCN7 ⁽²⁾	IOCCN6 ⁽²⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	182
IOCCP	IOCCP7 ⁽²⁾	IOCCP6 ⁽²⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	182
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	151
TRISC	TRISC7 ⁽²⁾	TRISC7(2)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165

TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE

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

Note 1: Unimplemented, read as '1'.

2: PIC16(L)F1619 only.

15.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference (FVR) is a stable voltage reference, independent of VDD, with a nominal output level (VFVR) of 1.024V. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- Comparator positive input
- · Comparator negative input

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

15.1 Independent Gain Amplifier

The output of the FVR supplied to the peripherals, (listed above), is routed through a programmable gain amplifier. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three 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 Section17.0 "Analog-to-Digital Converter (ADC) Module" for additional information.

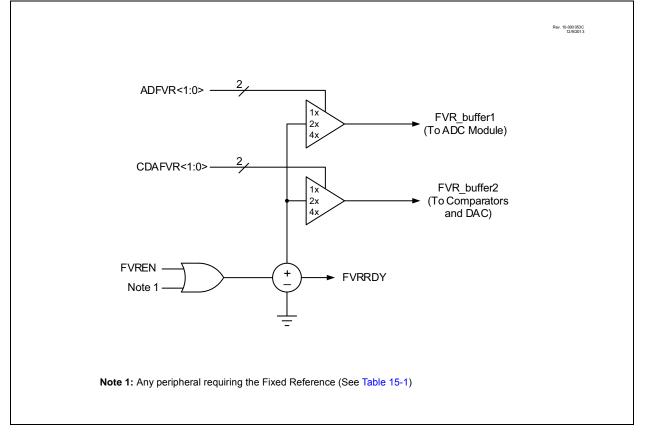
The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the comparator modules. Reference **Section19.0** "Comparator Module" for additional information.

To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the Buffer Gain Selection bits.

15.2 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 Figure 36-62: FVR Stabilization Period, PIC16LF1614/8 Only.

FIGURE 15-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Peripheral	Conditions	Description
HFINTOSC	FOSC<2:0> = 010 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.
LDO	All PIC16F1615/9 devices, when VREGPM = 1 and not in Sleep	The device runs off of the Low-Power Regulator when in Sleep mode.

 TABLE 15-1:
 PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

15.3 Register Definitions: FVR Control

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN ⁽¹⁾	FVRRDY ⁽²⁾	TSEN ⁽³⁾	TSRNG ⁽³⁾	CDAFV	R<1:0> ⁽¹⁾	ADFVR	<1:0> ⁽¹⁾
bit 7				•			bit
Legend: R = Readable	a hit	W = Writable	bit	II – I Inimpler	nented bit, read	as '0'	
u = Bit is unc		x = Bit is unk			at POR and BO		thar Resots
1' = Bit is unc	•	(0) = Bit is cle			pends on condit		
	•			q - value dej			
bit 7	1 = Fixed Vo	d Voltage Refe Itage Referenc Itage Referenc	e is enabled	bit ⁽¹⁾			
bit 6	1 = Fixed Vo	ed Voltage Re Itage Referenc Itage Referenc	e output is rea		enabled		
bit 5	1 = Tempera	erature Indicato ture Indicator i ture Indicator i	s enabled)			
bit 4	1 = VOUT = V	perature Indica /DD - 4VT (High /DD - 2VT (Low	n Range)	lection bit ⁽³⁾			
bit 3-2	11 = Compara 10 = Compara 01 = Compara	ator FVR Buffe ator FVR Buffe	er Gain is 4x, w er Gain is 2x, w er Gain is 1x, w	/ith output VCD	bits ⁽¹⁾ DAFVR = 4x VFVR DAFVR = 2x VFVR DAFVR = 1x VFVR	₂ (4)	
bit 1-0	11 = ADC FV 10 = ADC FV 01 = ADC FV	R Buffer Gain	is 4x, with out _l is 2x, with out _l	ection bit ⁽¹⁾ put VADFVR = 4 put VADFVR = 2 put VADFVR = 1	2x V _{FVR} (4)		
ing	minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clear the Buffer Gain Selection bits.						

REGISTER 15-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

- 2: FVRRDY is always '1' for the PIC16LF1615/9 devices.
- 3: See Section16.0 "Temperature Indicator Module" for additional information.
- 4: Fixed Voltage Reference output cannot exceed VDD.

TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE 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	CDAFV	′R<1:0>	ADFVF	R<1:0>	186

Legend: Shaded cells are unused by the Fixed Voltage Reference module.

16.0 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 -40°C and +85°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.

16.1 Circuit Operation

Figure 16-1 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 16-1 describes the output characteristics of the temperature indicator.

EQUATION 16-1: VOUT RANGES

High Range: VOUT = VDD - 4VT

Low Range: VOUT = VDD - 2VT

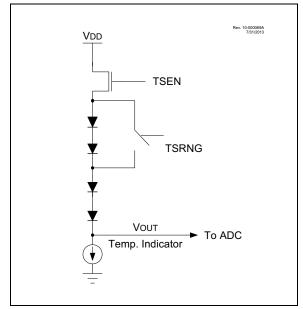
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section15.0 "Fixed Voltage Reference (FVR)"** for more information.

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 16-1: TEMPERATURE CIRCUIT DIAGRAM



16.2 Minimum Operating VDD

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 16-1 shows the recommended minimum VDD vs.range setting.

TABLE 16-1: RECOMMENDED VDD VS. RANGE

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

16.3 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 Section17.0 "Analog-to-Digital Converter (ADC) Module" for detailed information.

16.4 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.

TABLE 16-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFV	R<1:0>	ADFVF	R<1:0>	118

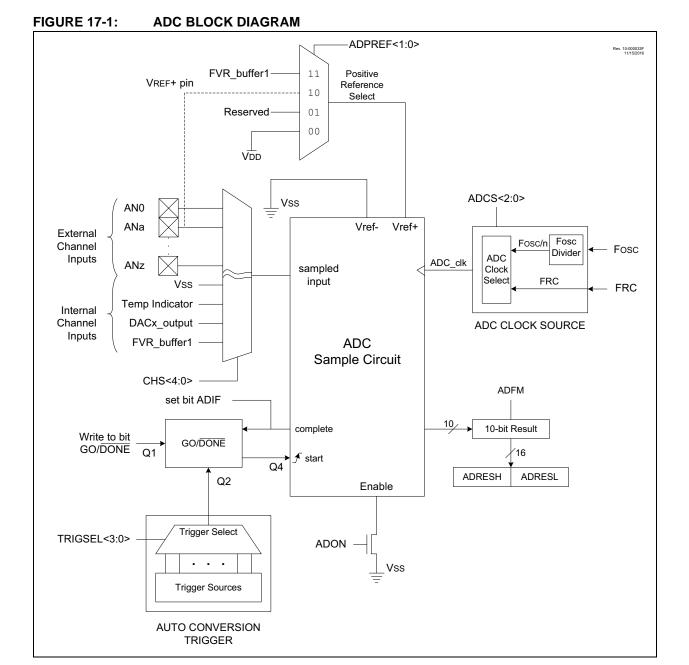
Legend: Shaded cells are unused by the temperature indicator module.

17.0 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 17-1 shows the block diagram of the ADC.

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

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



17.1 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

17.1.1 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 **Section12.0 "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.

17.1.2 CHANNEL SELECTION

There are up to 15 channel selections available:

- AN<11:0> pins (PIC16(L)F1619 only)
- AN<7:0> pins (PIC16(L)F1615 only)
- Temperature Indicator
- DAC1_output
- FVR_buffer1

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

When changing channels, a delay (TACQ) is required before starting the next conversion. Refer to **Section17.2.6 "ADC Conversion Procedure"** for more information.

Note: It is recommended that when switching from an ADC channel of a higher voltage to a channel of a lower voltage, the software selects the Vss channel before switching to the channel of the lower voltage. If the ADC does not have a dedicated Vss input channel, the Vss selection (DAC1R<4:0> = b'00000') through the DAC output channel can be used. If the DAC is in use, a free input channel can be connected to Vss, and can be used in place of the DAC.

17.1.3 ADC VOLTAGE REFERENCE

The ADC module uses a positive and a negative voltage reference. The positive reference is labeled ref+ and the negative reference is labeled ref-.

The positive voltage reference (ref+) is selected by the ADPREF bits in the ADCON1 register. The positive voltage reference source can be:

- VREF+ pin
- Vdd
- FVR_buffer1

The negative voltage reference (ref-) source is:

• Vss

17.1.4 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 (internal RC 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 17-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the ADC conversion requirements in **Section35.0** "**Electrical Specifications**" for more information. Table 17-1 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.

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

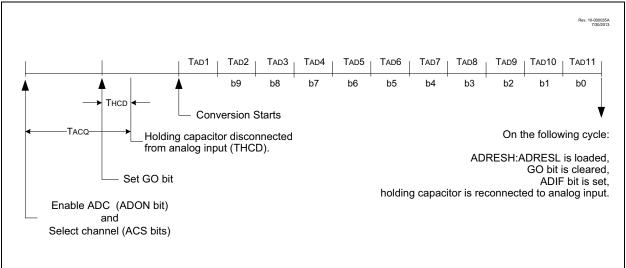
TABLE 17-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES

Legend: Shaded cells are outside of recommended range.

Note 1: The FRC source has a typical TAD time of 1.7 us.

- **2:** When the device frequency is greater than 1 MHz, the FRC clock source is only recommended if the conversion will be performed during Sleep.
- 3: The TAD period when using the FRC clock source can fall within a specified range, (see TAD parameter). The TAD period when using the FOSC-based clock source can be configured for a more precise TAD period. However, the FRC clock source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 17-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES



17.1.5 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.

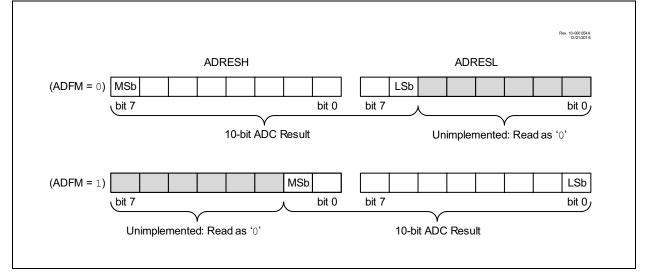
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume 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.

17.1.6 RESULT FORMATTING

The 10-bit ADC 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 17-3 shows the two output formats.





17.2 ADC Operation

17.2.1 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 Section17.2.6 "ADC Conver-
	sion Procedure".

17.2.2 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

17.2.3 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.

17.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. Performing the ADC conversion during Sleep can reduce system noise. 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.

17.2.5 AUTO-CONVERSION TRIGGER

The auto-conversion trigger allows periodic ADC measurements without software intervention. When a rising edge of the selected source occurs, the GO/DONE bit is set by hardware.

The auto-conversion trigger source is selected with the TRIGSEL<4:0> bits of the ADCON2 register.

Using the auto-conversion trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See Table 17-2 for auto-conversion sources.

TABLE 17-2: AUTO-CONVERSION SOURCES

Source Peripheral	Signal Name
Timer0	T0_overflow
Timer1	T1_overflow
Timer2	TMR2_postscaled
Timer4	TMR4_postscaled
Timer6	TMR6_postscaled
Comparator C1	C1_OUT_sync
Comparator C2	C2_OUT_sync
SMT1	SMT1_CPW
SMT1	SMT1_CPR
SMT1	SMT1_PR
SMT2	SMT2_CPW
SMT2	SMT2_CPR
SMT2	SMT2_PR
CCP1	CCP1_out
CCP2	CCP2_out

17.2.6 ADC 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)
 - Disable weak pull-ups either globally (Refer to the OPTION_REG register) or individually (Refer to the appropriate WPUx 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 Section17.4 "ADC Acquisition Requirements".

EXAMPLE 17-1: ADC CONVERSION

;This coo	de block confi	gures the ADC							
;for polling, Vdd and Vss references, FRC									
;oscillator and AN0 input.									
;									
;Convers	ion start & po	lling for completion							
; are ind	cluded.								
;									
BANKSEL	ADCON1	;							
MOVLW	B'11110000'	;Right justify, FRC							
		;oscillator							
MOVWF	ADCON1	;Vdd and Vss Vref+							
BANKSEL	TRISA	;							
BSF	TRISA,0	;Set RA0 to input							
BANKSEL	WPUA	;							
BCF	wpua,0	;Disable weak							
		;pull-up on RA0							
BANKSEL	ANSEL	;							
BSF	ANSEL,0	;Set RA0 to analog							
BANKSEL	ADCON0	;							
MOVLW	B'0000001'	;Select channel ANO							
MOVWF	ADCON0	;Turn ADC On							
CALL	SampleTime	;Acquisiton delay							
BSF	ADCON0, ADGO	;Start conversion							
BTFSC	ADCON0, ADGO	;Is conversion done?							
GOTO	\$-1	;No, test again							
BANKSEL	ADRESH	;							
MOVF	ADRESH,W	;Read upper 2 bits							
MOVWF	RESULTHI	;store in GPR space							
BANKSEL	-	;							
MOVF	ADRESL,W	;Read lower 8 bits							
MOVWF	RESULTLO	;Store in GPR space							

17.3 Register Definitions: ADC Control

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				
oit 7	·						bit				
Legend:											
R = Readable bit		W = Writable	bit	-	mented bit, rea						
u = Bit is	unchanged	x = Bit is unk	nown	-n/n = Value a	at POR and B	OR/Value at all o	other Resets				
'1' = Bit is	set	'0' = Bit is cle	ared								
bit 7	Unimpleme	nted: Read as	' ∩'								
bit 6-2	-										
JIL 0-2		CHS<4:0>: Analog Channel Select bits 11111 = FVR (Fixed Voltage Reference) Buffer 1 Output ⁽³⁾									
	11110 = DA	C (Digital-to-Ar	alog Converte	r) ⁽²⁾							
	11101 = Te r	nperature Indic	ator ⁽¹⁾	,							
	11100 = Re	served. No cha	nnel connecte	d.							
	•										
	•										
	01100 = Re	served. No cha	innel connecte	d.							
	01011 = AN										
	01010 = AN										
	01001 = AN										
	01000 = AN			d							
		eserved. No channel connected. Reserved. No channel connected.									
		eserved. No channel connected.									
		00 = Reserved. No channel connected.									
		served. No cha	innel connecte	d.							
	00111 = AN										
	00110 = AN 00101 = AN										
	00100 = AN										
	00011 = AN	13									
	00010 = AN										
	00001 = AN										
hit 1	00000 = AN	IU ADC Conversio	n Statua hit								
bit 1				ttina this hit sta	irts an ADC co	nversion cycle.					
						sion has comple	eted.				
		version comple				•					
bit 0	ADON: ADC	Enable bit									
	1 = ADC is ϵ										
	0 = ADC is c	lisabled and co	nsumes no ope	erating current							
Note 1:	See Section16.0			odule".							
2:	See Section18.0	-	-		•	more informatio	n.				
3:	See Section15.0	-		FVR)" for more	information.						
4:	AN<11:8> availab	ble on PIC16(L)	F1619 only								

REGISTER 17-1: ADCON0: ADC CONTROL REGISTER 0

4: AN<11:8> available on PIC16(L)F1619 only.

PIC16(L)F1615/9

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		ADCS<2:0>		_	_	ADPRE	EF<1:0>
bit 7							bit (
Legend:							
R = Readable	e bit	W = Writable	bit		mented bit, read		
u = Bit is unc	hanged	x = Bit is unk	nown	-n/n = Value a	at POR and BC	R/Value at all	other Resets
'1' = Bit is set	t	'0' = Bit is cle	ared				
bit 6-4	loaded. ADCS<2:0> 111 = FRC 110 = Fosc 101 = Fosc 011 = FRC 010 = Fosc 010 = Fosc 001 = Fosc	:/16 :/4 (clock supplied :/32 :/8	ion Clock Sele from an intern	ect bits al RC oscillator	r)	hen the conve	ersion result is
	000 = Fosc						
bit 3-2	-	nted: Read as '					
bit 1-0	11 = VRPOS 10 = VRPOS 01 = Reserv	:0>: ADC Positi is connected to is connected to ved is connected to	o internal Fixed o external VREP	Voltage Refer			
Note 1: W	hen selecting t	he VREF+ pin as	the source of	the positive re	ference, be awa	are that a minii	num voltage

e 1: When selecting the VREFF pin as the source of the positive reference, be aware that a m specification exists. See SectionTABLE 35-13: "Analog-to-Digital Converter (ADC) Characteristics(1,2,3)" for details.

R/W-0/		R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0		
	Т	RIGSEL<4:0>(1)			—	—		
bit 7							bit		
Legend:									
-	able bit	W = Writable	bit	U = Unimplem	ented bit. read	d as '0'			
R = Readable bit u = Bit is unchanged		x = Bit is unkr		-		R/Value at all o	other Resets		
'1' = Bit is set		'0' = Bit is cle							
bit 7-3	TRIGSEL<4	:0>: Auto-Conv	ersion Trigger	Selection bits ⁽¹⁾					
	11111 = Re	eserved							
	•								
	10101 = Re	served							
	10100 = AT	10100 = AT1_cmp3							
		10011 = AT1_cmp2							
	10010 = AT								
	10001 = CL 10000 = CL								
	01111 = CL								
	01110 = CL								
		IR5_overflow							
		IR3_overflow							
	01011 = SN								
	01010 =SM	1R6_postscaled							
		1R4_postscaled							
	00111 = C2								
		_OUT_sync							
		1R2_postscaled							
	00100 = T1 00011 = T0								
	00011 = 10 00010 = CC								
	00001 = CC								
		auto-conversio	n trigger seled	ted					
bit 2-0	Unimpleme	nted: Read as '	0'						
Note 1:	This is a rising ed	dge sensitive inp	out for all sour	ces.					
2:	Signal also sets i	to corresponding	n interrunt floo						

REGISTER 17-3: ADCON2: ADC CONTROL REGISTER 2

REGISTER 17-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) 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
			ADRE	S<9:2>			
bit 7							bit 0
Legend:							
R = Readable	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 at POR and BOR/Value at all other F			
'1' = Bit is set		'0' = Bit is clea	red				

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

REGISTER 17-5: 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 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 7-6 ADRES<1:0>: ADC Result Register bits Lower two bits of 10-bit conversion result bit 5-0 Reserved: Do not use.

-n/n = Value at POR and BOR/Value at all other Resets

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									
Legend:									
R = Readable bit W = Writable bit				U = Unimplemented bit, read as '0'					

REGISTER 17-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

'1' = Bit is set	'0' = Bit is cleared
bit 7-2	Reserved: Do not use.

x = Bit is unknown

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

u = Bit is unchanged

REGISTER 17-7: ADRESL: ADC RESULT REGISTER LOW (ADRESL) 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		
ADRES<7:0>									
bit 7	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

17.4 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 17-4. 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 17-4. The maximum recommended impedance for analog sources is 10 k Ω . As the

source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an ADC acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 17-1 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 17-1: ACQUISITION TIME EXAMPLE

Assumptions: Temperature = 50°C and external impedance of 10k
$$\Omega$$
 5.0V VDD

$$TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient
= TAMP + TC + TCOFF
= 2µs + TC + [(Temperature - 25°C)(0.05µs/°C)]
The value for TC can be approximated with the following equations:
$$VAPPLIED\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = VCHOLD \qquad ;[1] VCHOLD charged to within 1/2 lsb$$

$$VAPPLIED\left(1 - e^{\frac{-TC}{RC}}\right) = VCHOLD \qquad ;[2] VCHOLD charge response to VAPPLIED$$

$$VAPPLIED\left(1 - e^{\frac{-TC}{RC}}\right) = VAPPLIED\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \qquad ;combining [1] and [2]$$
Note: Where n = number of bits of the ADC.
Solving for TC:

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

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

= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885)
= 1.12\mus

Therefore:

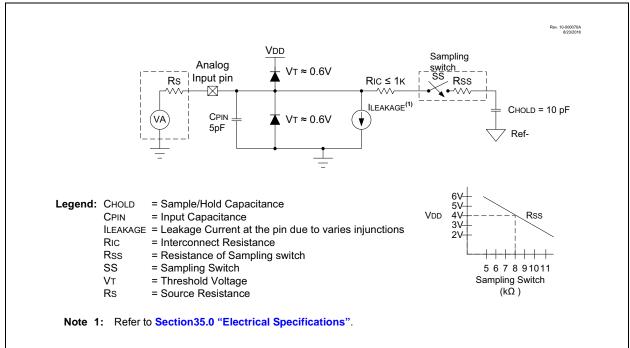
$$TACQ = 2\mu s + 1.12\mu s + [(50^{\circ}C - 25^{\circ}C)(0.05\mu s/^{\circ}C)]$$

= 4.37\mu s

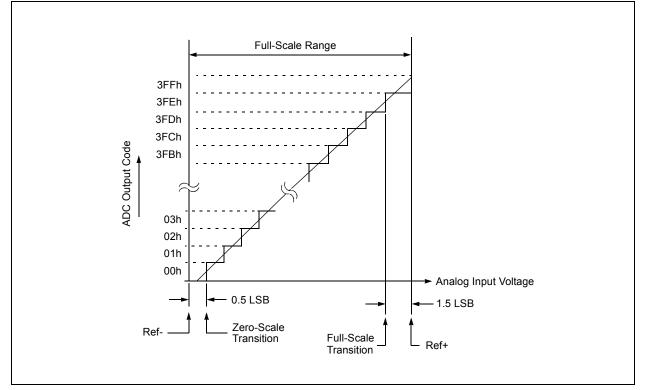
Note 1: The reference voltage (VRPOS) has no effect on the equation, since it cancels itself out.

- 2: The charge holding capacitor (CHOLD) is not discharged after each conversion.
- **3:** The maximum recommended impedance for analog sources is 10 k Ω . This is required to meet the pin leakage specification.









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	195
ADCON1	ADFM		ADCS<2:0> —				ADPRE	F<1:0>	196
ADCON2		Т	RIGSEL<4:0	>		_	—		197
ADRESH	ADC Result Register High								198, 199
ADRESL	ADC Result	Register Low							198, 199
ANSELA	_	_	-	ANSA4	—	ANSA2	ANSA1	ANSA0	152
ANSELC	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	-	—	ANSC3	ANSC2	ANSC1	ANSC0	166
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
TRISA	_	_	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	151
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFV	′R<1:0>	ADFV	२<1:0>	186

TABLE 17-3: 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.

Note 1: Unimplemented, read as '1'.

2: PIC16(L)F1619 only.

18.0 8-BIT DIGITAL-TO-ANALOG CONVERTER (DAC1) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 256 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- · ADC input channel
- DACxOUT1 pin

The Digital-to-Analog Converter (DAC) is enabled by setting the DAC1EN bit of the DAC1CON0 register.

EQUATION 18-1: DAC OUTPUT VOLTAGE

 $\frac{IF DACIEN = 1}{VOUT} = \left((VSOURCE+ - VSOURCE-) \times \frac{DACIR[7:0]}{2^8} \right) + VSOURCE-$ VSOURCE+ = VDD, VREF, or FVR BUFFER 2 VSOURCE- = VSS

18.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in **Section35.0** "Electrical **Specifications**".

18.3 DAC Voltage Reference Output

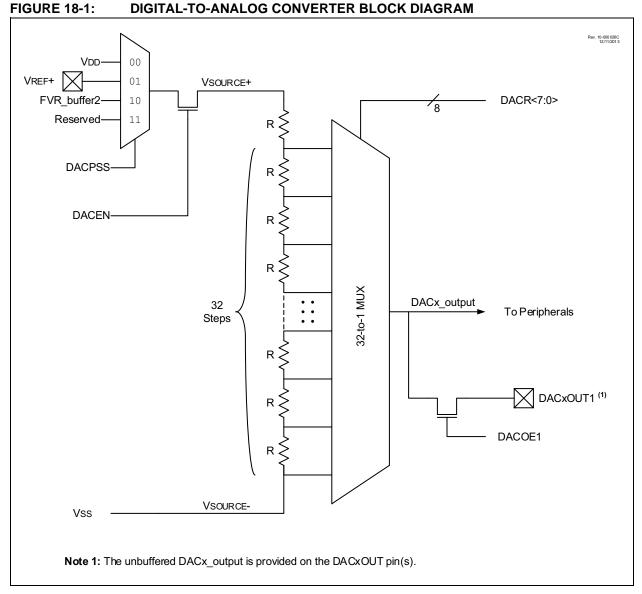
The DAC voltage can be output to the DACxOUT1 pin by setting the DAC1OE1 bit of the DAC1CON0 register. Selecting the DAC reference voltage for output on the DACxOUT1 pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACxOUT1 pin when it has been configured for DAC reference voltage output will always return a '0'.

Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to either DACxOUT1 pin. Figure 18-2 shows an example buffering technique.

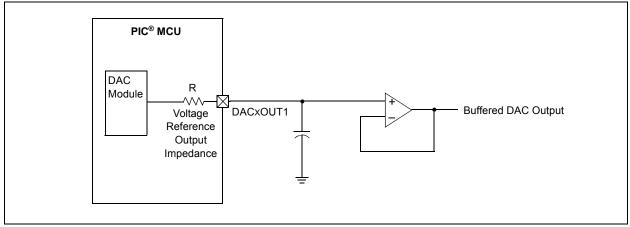
18.1 Output Voltage Selection

The DAC has 256 voltage level ranges. The 256 levels are set with the DAC1R<7:0> bits of the DAC1CON1 register.

The DAC output voltage is determined by Equation 18-1:







© 2014-2017 Microchip Technology Inc.

18.4 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DAC1CON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

18.5 Effects of a Reset

A device Reset affects the following:

- DAC is disabled.
- DAC output voltage is removed from the DACxOUT1 pin.
- The DAC1R<7:0> range select bits are cleared.

18.6 Register Definitions: DAC Control

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	
DAC1EN	_	DAC10E1	_	DAC1P	SS<1:0>	_		
bit 7				•			bit 0	
Legend:								
R = Readable bit		W = Writable bit	t	U = Unimpleme	ented bit, read as	'0'		
u = Bit is unchan	ged	x = Bit is unknow	wn	-n/n = Value at	POR and BOR/Va	alue at all other F	Resets	
'1' = Bit is set		'0' = Bit is cleare	ed					
bit 6	 1 = DAC is enabled 0 = DAC is disabled Unimplemented: Read as '0' 							
bit 6 bit 5								
	0 = DAC volta	ge level is discor	nected from the	e DACxOUT1 pir	ו			
bit 4	4 Unimplemented: Read as '0'							
bit 3-2	DAC1PSS<1:0 11 = Reserved, 10 = FVR Buffe 01 = VREF+ pin 00 = VDD	r2 output	e Source Select	bits				
bit 1-0	Unimplemente	d: Read as '0'						

REGISTER 18-1: DAC1CON0: DAC1 CONTROL REGISTER 0

REGISTER 18-2: DAC1CON1: DAC1 CONTROL REGISTER 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	R/W-0/0	
DAC1R<7:0>								
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 DAC1R<7:0>: DAC1 Voltage Output Select bits

TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC1 MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVF	۲<1:0>	186
DAC1CON0	DAC1EN	_	DAC10E1	_	DAC1PS	SS<1:0>	—	_	206
DAC1CON1	DAC1CON1 DAC1R<7:0>								206

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

19.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- · Independent comparator control
- · Programmable input selection
- · Comparator output is available internally/externally
- · Programmable output polarity
- Interrupt-on-change
- · Wake-up from Sleep
- Programmable Speed/Power optimization
- PWM shutdown
- Programmable and Fixed Voltage Reference

19.1 Comparator Overview

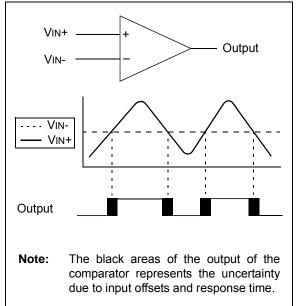
A single comparator is shown in Figure 19-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

The comparators available for this device are located in Table 19-1.

TABLE 19-1: COMPARATOR AVAILABILITY PER DEVICE

Device	C1	C2
PIC16(L)F1619	•	٠
PIC16(L)F1615	٠	•

FIGURE 19-1: SINGLE COMPARATOR



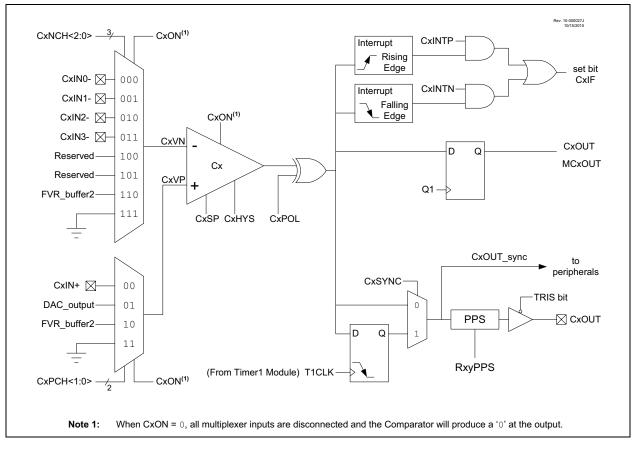


FIGURE 19-2: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM

19.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 registers (see Register 19-1) contain Control and Status bits for the following:

- Enable
- Output selection
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 registers (see Register 19-2) contain Control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

19.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

19.2.2 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- · CxOE bit of the CMxCON0 register must be set
- · Corresponding TRIS bit must be cleared
- · CxON bit of the CMxCON0 register must be set
 - Note 1: The CxOE bit of the CMxCON0 register overrides the PORT data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override.
 - 2: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

19.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 19-2shows the output state versus inputconditions, including polarity control.

TABLE 19-2: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

19.2.4 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1' which selects the Normal Speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

19.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See **Section35.0 "Electrical Specifications**" for more information.

19.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section22.5 "Timer1 Gate**" for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

19.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from a comparator can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 19-2) and the Timer1 Block Diagram (Figure 22-1) for more information.

19.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, the following bits must be set:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Note: Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register.

19.6 Comparator Positive Input Selection

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN+ analog pin
- DAC output
- FVR (Fixed Voltage Reference)
- Vss (Ground)

See Section15.0 "Fixed Voltage Reference (FVR)" for more information on the Fixed Voltage Reference module.

See Section18.0 "8-bit Digital-to-Analog Converter (DAC1) Module" for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

19.7 Comparator Negative Input Selection

The CxNCH<2:0> bits of the CMxCON1 register direct an analog input pin or analog ground to the inverting input of the comparator:

- CxIN0- pin
- CxIN1- pin
- CxIN2- pin
- CxIN3- pin
- Analog Ground
- FVR_buffer2

Some inverting input selections share a pin with the operational amplifier output function. Enabling both functions at the same time will direct the operational amplifier output to the comparator inverting input.

Note: To use CxINy+ and CxINy- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

19.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Section35.0 "Electrical Specifications" for more details.

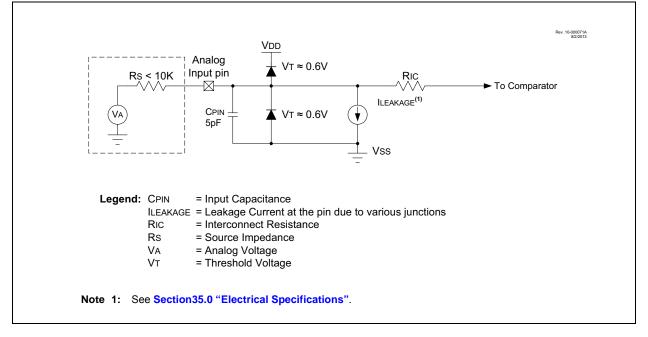
19.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 19-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

FIGURE 19-3: ANALOG INPUT MODEL

A maximum source impedance of $10 \text{ k}\Omega$ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

- Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.
 - 2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.



19.10 Register Definitions: Comparator Control

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC
bit 7							bit C
Legend:							
R = Readab		W = Writable	e bit	•	mented bit, read		
u = Bit is und	0	x = Bit is unk		-n/n = Value	at POR and BC	R/Value at all	other Resets
'1' = Bit is s€	et	'0' = Bit is cle	eared				
bit 7	CxON: Com	parator Enable	e bit				
	1 = Compar	ator is enabled					
	0 = Compar	ator is disabled	and consumes	s no active pow	ver		
bit 6	CxOUT: Co	mparator Outpu	ıt bit				
		1 (inverted pola	arity):				
	1 = CxVP < CxVP < CxVP > CxV	•••••					
	0 = CxVP >	0 (non-inverted	polarity):				
	1 = CxVP >		polarity).				
	0 = CxVP <	CxVN					
bit 5	Unimpleme	nted: Read as	' 0'				
bit 4	CxPOL: Co	mparator Outpu	ut Polarity Sele	ct bit			
	1 = Compar	ator output is ir	nverted				
	0 = Compar	ator output is n	ot inverted				
bit 3	Unimpleme	nted: Read as	'0'				
bit 2	CxSP: Com	parator Speed/	Power Select b	bit			
	1 = Compar	ator operates ir	n normal power	, higher speed	mode		
	0 = Compar	ator operates ir	n Low-power, L	ow-speed mod	le		
bit 1	CxHYS: Co	mparator Hyste	resis Enable b	it			
		rator hysteresis					
	0 = Compa	rator hysteresis	disabled				
bit 0	CxSYNC: C	omparator Out	put Synchrono	us Mode bit			
					ronous to chang	ges on Timer1	clock source
		updated on the					
	0 = Compa	rator output to	imeri and I/O	pin is asynchro	onous		

REGISTER 19-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 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	R/W-0/0
CxINTP	CxINTN			K/VV-0/0	K/ W-0/0	CxNCH<2:0>	rt/ VV-U/U
							L: 1 O
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplei	mented bit, rea	d as '0'	
u = Bit is unch	nanged	x = Bit is unkr	nown	-n/n = Value	at POR and BC	R/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7	CxINTP: Cor	nparator Interru	pt on Positive	Going Edge E	nable bits		
		F interrupt flag v rupt flag will be		· ·	0 0		
bit 6	CxINTN: Cor	nparator Interru	ipt on Negativ	e Going Edge I	Enable bits		
		F interrupt flag v rupt flag will be					
bit 5-4	CxPCH<1:0>	-: Comparator I	Positive Input	Channel Select	t bits		
		connects to AGI					
		connects to FVF					
		connects to VDA connects to CxII					
bit 3		ted: Read as '0'					
bit 2-0	•	Comparator I		t Channel Sele	ct bits		
		I connects to A	•				
	110 = CxVN	I connects to F	/R Buffer 2				
	101 = Rese						
	100 = Reserved		vINI2 nin				
		I connects to C: I connects to C:					
		I connects to C	•				
	000 = CxVN	I connects to C	xIN0- pin				

REGISTER 19-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

REGISTER 19-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
_	_	—	—	_	_	MC2OUT	MC1OUT
bit 7							bit 0

Legend:			
R = Reada	able bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is u	nchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is :	set	'0' = Bit is cleared	
bit 7-2	Unimpler	nented: Read as '0'	
bit 1	MC2OUT:	: Mirror Copy of C2OUT bit	

bit 0	MC1OUT: Mirror Copy of C1OUT bit
	15

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	-	_	ANSA4	-	ANSA2	ANSA1	ANSA0	152
CM1CON0	C1ON	C10UT	_	C1POL	_	C1SP	C1HYS	C1SYNC	212
CM1CON1	C1INTP	C1INTN	C1PCH	1<1:0>	_		C1NCH<2:0>		213
CM2CON0	C2ON	C2OUT	_	C2POL	_	C2SP	C2HYS	C2SYNC	212
CM2CON1	C2INTP	C2INTN	C2PCH	1<1:0>	_		C2NCH<2:0>		213
CMOUT	—	_	_		_	_	MC2OUT	MC10UT	213
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFV	′R<1:0>	ADFVF	R<1:0>	186
DAC1CON0	DAC1EN	_	DAC10E1	_	DAC1P	SS<1:0>	—	_	206
DAC1CON1				DAC1R	<7:0>				206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE2	OSFIE	C2IE	C1IE	_	BCL1IE	TMR6IE	TMR4IE	CCP2IE	99
PIR2	OSFIF	C2IF	C1IF	_	BCL1IF	TMR6IF	TMR4IF	CCP2IF	104
TRISA	—	_	TRISA5	TRISA4	_(1)	TRISA2	TRISA1	TRISA0	151
TRISC ⁽²⁾	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165

TABLE 19-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Legend: — = unimplemented location, read as '0'. Shaded cells are unused by the comparator module.

Note 1: Unimplemented, read as '1'.

2: PIC16F1619 only.

20.0 ZERO-CROSS DETECTION (ZCD) MODULE

The ZCD module detects when an A/C signal crosses through the ground potential. The actual zero crossing threshold is the zero crossing reference voltage, VCPINV, which is typically 0.75V above ground.

The connection to the signal to be detected is through a series current limiting resistor. The module applies a current source or sink to the ZCD pin to maintain a constant voltage on the pin, thereby preventing the pin voltage from forward biasing the ESD protection diodes. When the applied voltage is greater than the reference voltage, the module sinks current. When the applied voltage is less than the reference voltage, the module sources current. The current source and sink action keeps the pin voltage constant over the full range of the applied voltage. The ZCD module is shown in the simplified block diagram Figure 20-2.

The ZCD module is useful when monitoring an A/C waveform for, but not limited to, the following purposes:

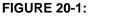
- A/C period measurement
- · Accurate long term time measurement
- Dimmer phase delayed drive
- Low EMI cycle switching

20.1 External Resistor Selection

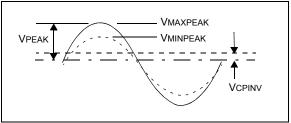
The ZCD module requires a current limiting resistor in series with the external voltage source. The impedance and rating of this resistor depends on the external source peak voltage. Select a resistor value that will drop all of the peak voltage when the current through the resistor is nominally 300 μ A. Refer to Equation 20-1 and Figure 20-1. Make sure that the ZCD I/O pin internal weak pull-up is disabled so it does not interfere with the current source and sink.

EQUATION 20-1: EXTERNAL RESISTOR

$$RSERIES = \frac{VPEAK}{3 \times 10^{-4}}$$

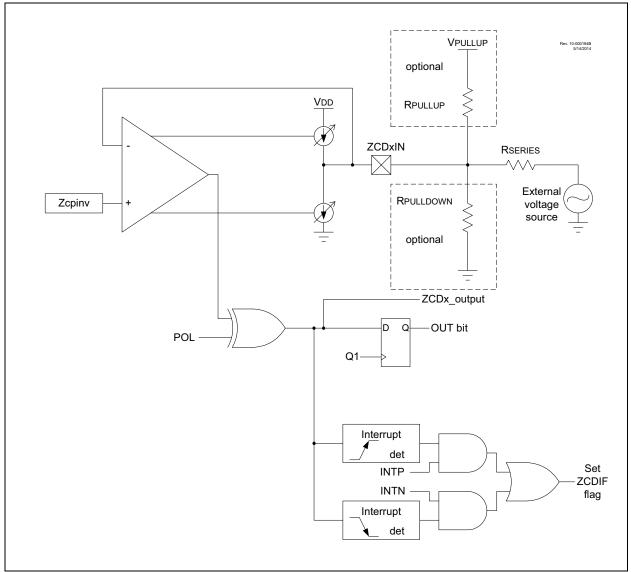


EXTERNAL VOLTAGE



PIC16(L)F1615/9





20.2 ZCD Logic Output

The ZCD module includes a Status bit, which can be read to determine whether the current source or sink is active. The ZCDxOUT bit of the ZCDxCON register is set when the current sink is active, and cleared when the current source is active. The ZCDxOUT bit is affected by the polarity bit.

20.3 ZCD Logic Polarity

The ZCDxPOL bit of the ZCDxCON register inverts the ZCDxOUT bit relative to the current source and sink output. When the ZCDxPOL bit is set, a ZCDxOUT high indicates that the current source is active, and a low output indicates that the current sink is active.

The ZCDxPOL bit affects the ZCD interrupts. See **Section20.4 "ZCD Interrupts"**.

20.4 ZCD Interrupts

An interrupt will be generated upon a change in the ZCD logic output when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in the ZCD for this purpose.

The ZCDIF bit of the PIR3 register will be set when either edge detector is triggered and its associated enable bit is set. The ZCDxINTP enables rising edge interrupts and the ZCDxINTN bit enables falling edge interrupts. Both are located in the ZCDxCON register.

To fully enable the interrupt, the following bits must be set:

- ZCDIE bit of the PIE3 register
- ZCDxINTP bit of the ZCDxCON register (for a rising edge detection)
- ZCDxINTN bit of the ZCDxCON register (for a falling edge detection)
- · PEIE and GIE bits of the INTCON register

Changing the ZCDxPOL bit will cause an interrupt, regardless of the level of the ZCDxEN bit.

The ZCDIF bit of the PIR3 register must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

20.5 Correcting for VCPINV offset

The actual voltage at which the ZCD switches is the reference voltage at the non-inverting input of the ZCD op amp. For external voltage source waveforms other than square waves, this voltage offset from zero causes the zero-cross event to occur either too early or too late.

20.5.1 CORRECTION BY AC COUPLING

When the external voltage source is sinusoidal, the effects of the ZCPINV offset can be eliminated by isolating the external voltage source from the ZCD pin with a capacitor, in addition to the voltage reducing resistor. The capacitor will cause a phase shift resulting in the ZCD output switch in advance of the actual zerocrossing event. The phase shift will be the same for both rising and falling zero crossings, which can be compensated for by either delaying the CPU response to the ZCD switch by a timer or other means, or selecting a capacitor value large enough that the phase shift is negligible.

To determine the series resistor and capacitor values for this configuration, start by computing the impedance, Z, to obtain a peak current of 300 uA. Next, arbitrarily select a suitably large non-polar capacitor and compute its reactance, Xc, at the external voltage source frequency. Finally, compute the series resistor, capacitor peak voltage, and phase shift by the formulas shown in Equation 20-2.

When this technique is used and the input signal is not present, the ZCD will tend to oscillate. To avoid this oscillation, connect the ZCD pin to VDD or GND with a high-impedance resistor such as 200K.

EQUATION 20-2: R-C CALCULATIONS

VPEAK = External voltage source peak voltage

- f = External voltage source frequency
- C = Series capacitor
- R = Series resistor
- $V_{\rm C}$ = Peak capacitor voltage
- Φ = Capacitor induced zero crossing phase advance in radians
- $T_\Phi\,$ = Time ZC event occurs before actual zero crossing

$$Z = \frac{VPEAK}{3 \times 10^{-4}}$$
$$XC = \frac{1}{2\pi fC}$$
$$R = \sqrt{Z^2 - Xc^2}$$
$$VC = XC(3 \times 10^{-4})$$
$$\Phi = Tan^{-1}\left(\frac{XC}{R}\right)$$
$$T\Phi = \frac{\Phi}{2\pi f}$$

EXAMPLE 20-1: R-C CALCULATIONS

VRMS = 120
VPEAK = VRMS *
$$\sqrt{2}$$
 = 169.7
f = 60 Hz
C = 0.1 µF
 $Z = \frac{VPEAK}{3 \times 10^{-4}} = \frac{169.7}{3 \times 10^{-4}} = 565.7 k\Omega$
 $XC = \frac{1}{2\pi fC} = \frac{1}{(2\pi \times 60 \times 1 \times 10^{-7})} = 26.53 k\Omega$
 $R = \sqrt{(Z^2 \times XC^2)} = 565.1 k\Omega \text{ (computed)}$
 $R = 560 k\Omega \text{ (used)}$
 $ZR = \sqrt{R^2 + Xc^2} = 560.6 k\Omega \text{ (using actual resistor)}$
IPEAK = $\frac{VPEAK}{ZR} = 302.7 \times 10^{-6}$
 $VC = XC \times Ipeak = 8.0V$
 $\Phi = Tan^{-1}(\frac{XC}{R}) = 0.047 \text{ radians}$
 $T\Phi = \frac{\Phi}{2\pi f} = 125.6 \mu s$

20.5.2 CORRECTION BY OFFSET CURRENT

When the waveform is varying relative to Vss, then the zero cross is detected too early as the waveform falls and too late as the waveform rises. When the waveform is varying relative to VDD, then the zero cross is detected too late as the waveform rises and too early as the waveform falls. The actual offset time can be determined for sinusoidal waveforms with the corresponding equations shown in Equation 20-3.

EQUATION 20-3: ZCD EVENT OFFSET

When External Voltage Source is relative to Vss:

$$TOFFSET = \frac{\operatorname{asin}\left(\frac{Vcpinv}{VPEAK}\right)}{2\pi \bullet Freq}$$

When External Voltage Source is relative to VDD:

$$TOFFSET = \frac{\operatorname{asin}\left(\frac{VDD - Vcpinv}{VPEAK}\right)}{2\pi \bullet Freq}$$

This offset time can be compensated for by adding a pull-up or pull-down biasing resistor to the ZCD pin. A pull-up resistor is used when the external voltage source is varying relative to Vss. A pull-down resistor is used when the voltage is varying relative to VDD. The resistor adds a bias to the ZCD pin so that the target external voltage source must go to zero to pull the pin voltage to the VCPINV switching voltage. The pull-up or pull-down value can be determined with the equations shown in Equation 20-4 or Equation 20-5.

EQUATION 20-4: ZCD PULL-UP/DOWN

When External Signal is relative to Vss:

$$RPULLUP = \frac{RSERIES(VPULLUP - Vcpinv)}{Vcpinv}$$

When External Signal is relative to VDD:

$$R_{PULLDOWN} = \frac{R_{SERIES}(Vcpinv)}{(VDD - Vcpinv)}$$

20.6 Handling VPEAK variations

If the peak amplitude of the external voltage is expected to vary, the series resistor must be selected to keep the ZCD current source and sink below the design maximum range of $\pm 600 \ \mu$ A and above a reasonable minimum range. A general rule of thumb is that the maximum peak voltage can be no more than six times the minimum peak voltage. To ensure that the maximum current does not exceed $\pm 600 \ \mu$ A and the minimum is at least $\pm 100 \ \mu$ A, compute the series resistance as shown in Equation 20-5. The compensating pull-up for this series resistance can be determined with Equation 20-4 because the pull-up value is independent from the peak voltage.

EQUATION 20-5: SERIES R FOR V RANGE

$$RSERIES = \frac{V_{MAXPEAK} + V_{MINPEAK}}{7 \times 10^{-4}}$$

20.7 Operation During Sleep

The ZCD current sources and interrupts are unaffected by Sleep.

20.8 Effects of a Reset

The ZCD circuit can be configured to default to the active or inactive state on Power-On-Reset (POR). When the \overline{ZCD} Configuration bit is cleared, the ZCD circuit will be active at POR. When the \overline{ZCD} Configuration bit is set, the ZCDxEN bit of the ZCDxCON register must be set to enable the ZCD module.

20.9 Register Definitions: ZCD Control

R/W-q/q	U-0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ZCDxEN	—	ZCDxOUT	ZCDxPOL	—	—	ZCDxINTP	ZCDxINTN
bit 7							bit (
Legend:							
R = Readab		W = Writable		•	nented bit, read		
u = Bit is un	changed	x = Bit is unkr	iown	-n/n = Value a	at POR and BO	R/Value at all c	other Resets
'1' = Bit is se	et	'0' = Bit is clea	ared	q = value dep	ends on config	uration bits	
		0 D /					
bit 7		o-Cross Detec		-			
						and sink currer and TRIS contr	
bit 6	Unimplemen	ted: Read as '	כי				
bit 5	ZCDxOUT: Z	ero-Cross Dete	ction Logic Le	evel bit			
	ZCDxPOL bit						
		is sinking curre is sourcing cur					
	ZCDxPOL bit		ent				
		is sourcing cur	rent				
	0 = ZCD pin	is sinking curre	nt				
bit 4		ero-Cross Dete	0	utput Polarity b	bit		
		c output is inve c output is not i					
bit 3-2	0	•					
	-	ted: Read as '		amunt Enchla b	:.		
bit 1		ero-Cross Pos					
		it is set on low- it is unaffected	· · ·	- '			
bit 0		Zero-Cross Neg					
		t is set on high	•	•			
	0 = ZCDIF bi						

REGISTER 20-1: ZCDxCON: ZERO-CROSS DETECTION CONTROL REGISTER

TABLE 20-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PIE3	—	-	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	108
PIR3	_	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	113
ZCD1CON	ZCD1EN		ZCD10UT	ZCD1POL	_	_	ZCD1INTP	ZCD1INTN	220

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

TABLE 20-2:SUMMARY OF CONFIGURATION WORD WITH THE ZCD MODULE

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
CONFIG2	13:8	_		LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	69
	7:0	ZCD	_	_	_	_	PPS1WAY	WRT	<1:0>	

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

21.0 TIMER0 MODULE

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

- 8-bit timer/counter register (TMR0)
- 3-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 21-1 is a block diagram of the Timer0 module.

21.1 Timer0 Operation

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

21.1.1 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.

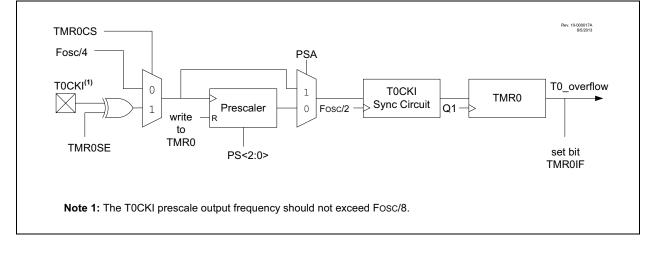
FIGURE 21-1: TIMER0 BLOCK DIAGRAM

21.1.2 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 for either input source is determined by the TMR0SE bit in the OPTION_REG register.



21.1.3 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.

21.1.4 TIMER0 INTERRUPT

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

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

21.1.5 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 **Section35.0** "Electrical **Specifications**".

21.1.6 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.

21.2 Register Definitions: Option Register

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 (
Legend:										
R = Readable	e bit	W = Writable	bit	U = Unimplen	nented bit, read	1 as '0'				
u = Bit is unc	hanged	x = Bit is unk	nown	-n/n = Value a	at POR and BO	R/Value at all c	ther Resets			
'1' = Bit is se	t	'0' = Bit is cle	ared							
bit 7	WPUEN: We	ak Pull-Up Ena	able bit							
		pull-ups are dis	· · ·		,					
	•	ll-ups are enab	-	al WPUx latch	values					
bit 6		INTEDG: Interrupt Edge Select bit								
		1 = Interrupt on rising edge of INT pin								
bit 5		0 = Interrupt on falling edge of INT pin TMR0CS: Timer0 Clock Source Select bit								
DIL D										
		1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (Fosc/4)								
bit 4		ner0 Source E		,						
		1 = Increment on high-to-low transition on T0CKI pin								
	0 = Incremen	0 = Increment on low-to-high transition on T0CKI pin								
bit 3	PSA: Presca	ler Assignment	bit							
		 1 = Prescaler is not assigned to the Timer0 module 0 = Prescaler is assigned to the Timer0 module 								
		•		odule						
bit 2-0	PS<2:0>: Pre	PS<2:0>: Prescaler Rate Select bits								
	Bit	Value Timer0	Rate							
		000 1:2								
		D01 1:4 D10 1:8								
		011 1:16								
	· · ·)							

REGISTER 21-1: OPTION_REG: OPTION REGISTER

TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

1:128

1:256

110

111

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON2	ON2 TRIGSEL<4:0> —							-	197
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			223
TMR0	Holding Register for the 8-bit Timer0 Count						221*		
TRISA	_		TRISA5	TRISA4	_(1)	TRISA2	TRISA1	TRISA0	151

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

Note 1: Unimplemented, read as '1'.

22.0 TIMER1/3/5 MODULE WITH GATE CONTROL

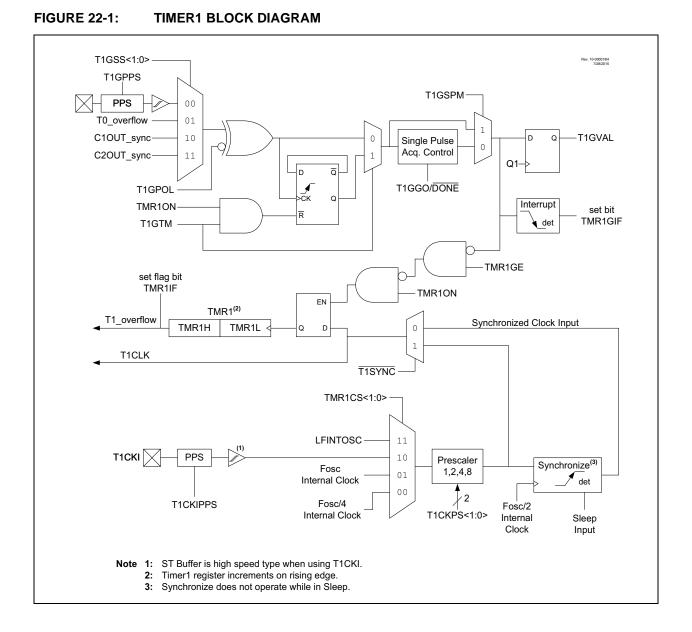
The Timer1/3/5 modules are a 16-bit timers/counters with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- · 2-bit prescaler
- Optionally synchronized comparator out
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- ADC Auto-Conversion Trigger(s)

- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-Pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 22-1 is a block diagram of the Timer1 module.

Note: Three identical Timer1 modules are implemented on this device. The timers are named Timer1, Timer3, and Timer5. All references to Timer1 apply as well to Timer3 and Timer5, as well as references to their associated registers.



22.1 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 22-1 displays the Timer1 enable selections.

TABLE 22-1: TIMER1 ENABLE SELECTIONS

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

22.2 Clock Source Selection

The TMR1CS<1:0> bits of the T1CON register are used to select the clock source for Timer1. Table 22-2 displays the clock source selections.

22.2.1 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 sources may be used:

- Asynchronous event on the T1G pin to Timer1
 gate
- C1 or C2 comparator input to Timer1 gate

22.2.2 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. The external clock source can be synchronized to the microcontroller system clock or it can run asynchronously.

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.

TABLE 22-2: CLOCK SOURCE SELECTIONS

TMR1CS<1:0>	Clock Source
11	LFINTOSC
10	External Clocking on T1CKI Pin
01	System Clock (Fosc)
0 0	Instruction Clock (Fosc/4)

22.3 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.

22.4 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see Section22.4.1 "Reading and Writing Timer1 in Asynchronous Counter Mode").

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

22.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

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

T1GSS	Timer1 Gate Source
00	Timer1 Gate pin (T1G)
01	Overflow of Timer0 (T0_overflow) (TMR0 increments from FFh to 00h)
10	Comparator 1 Output (C1_OUT_sync) ⁽¹⁾
11	Comparator 2 Output (C2_OUT_sync) ⁽¹⁾

Note 1: Optionally synchronized comparator output.

22.5.2.1 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

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.

22.5 Timer1 Gate

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

Timer1 gate can also be driven by multiple selectable sources.

22.5.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 22-3 for timing details.

TABLE 22-3: TIMER1 GATE ENABLE SELECTIONS

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

22.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 22-4. Source selection is controlled by the T1GSS<1:0> bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

gate circuitry.

22.5.2.2 Timer0 Overflow Gate Operation

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

22.5.3 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 22-4 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.						

22.5.4 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 22-5 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 22-6 for timing details.

22.5.5 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).

22.5.6 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).

22.6 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, the following bits must be set:

- 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.

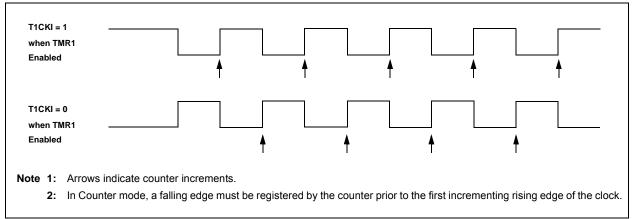
22.7 Timer1 Operation During Sleep

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
- T1SYNC bit of the T1CON register must be set
- TMR1CS bits 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.

FIGURE 22-2: TIMER1 INCREMENTING EDGE



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



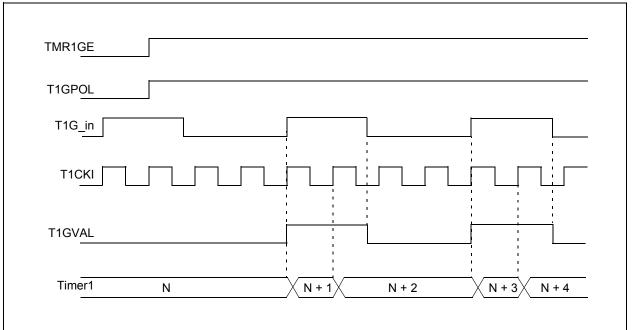
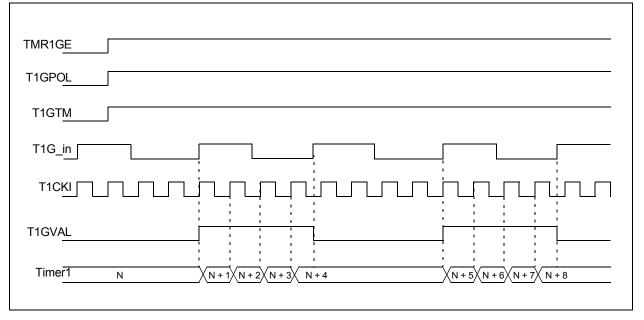


FIGURE 22-4: TIMER1 GATE TOGGLE MODE



100NL 22-3.		
TMR1GE		
T1GPOL		
T1GSPM		
T1GG <u>O/</u> DONE	✓ Set by software Counting enabled on	Cleared by hardware on falling edge of T1GVAL
T1G_in	rising edge of T1G	
T1CKI		
T1GVAL		
Timer1	N N + 1	N + 2
TMR1GIF	— Cleared by software	Cleared by Set by hardware on falling edge of T1GVAL

FIGURE 22-5: TIMER1 GATE SINGLE-PULSE MODE

FIGURE 22-6:	TIMER1 GATE SINGL	E-PULSE AND TOGGLE COMBINED MODE
TMR1GE		
T1GPOL		
T1GSPM		
T1GTM		
T1GG <u>O/</u> DONE T1G_in	← Set by software Counting enabled rising edge of T1	on G
тіскі		
T1GV <u>AL</u>		
Timer1	N	<u>N+1</u> <u>N+2</u> <u>N+3</u> <u>N+4</u>
TMR1GIF	 Cleared by software 	Set by hardware on Cleared by falling edge of T1GVAL Cleared by software

22.8 Register Definitions: Timer1 Control

REGISTER 22-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u	U-0	R/W-0/u		
TMR1CS<1:0>		T1CKPS<1:0>			T1SYNC	—	TMR10N		
bit 7							bit 0		
Legend:									
R = Readable	bit	W = Writable I	oit	U = Unimpler	nented bit, read	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						
bit 7-6	TMR1CS<1:0	D>: Timer1 Cloc	k Source Sele	ect bits					
	11 =LFINTOS	SC							
		10 =T1CKI							
	01 =Fosc 00 =Fosc/4								
bit 5-4		- Timer1 Input	Clock Presca	ale Select hits					
	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	Unimplemen	ted: Read as ')'						
bit 2	T1SYNC: Tin	T1SYNC: Timer1 Synchronization Control bit							
	1 = Do not synchronize asynchronous clock input								
	-	nize asynchrono		t with system c	lock (Fosc)				
bit 1	Unimplemen	ted: Read as 'o)'						
bit 0	TMR1ON: Timer1 On bit								
	1 = Enables			a: a					
	0 = Stops Tir	mer1 and clears	s Timer1 gate	tlip-tlop					

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	S<1:0>				
bit 7			•				bit (
<u> </u>											
Legend:	. 1.9		1.11								
R = Readabl		W = Writable		•	nented bit, read						
u = Bit is und	-	x = Bit is unkr			at POR and BO		other Resets				
'1' = Bit is se	l .	'0' = Bit is cle	areo	HC = BILIS CIE	eared by hardw	are					
bit 7	TMR1GE: Ti	mer1 Gate Ena	ble bit								
	If TMR10N =										
	This bit is ign										
		<u>If TMR1ON = 1</u> : 1 = Timer1 counting is controlled by the Timer1 gate function									
) = Timer1 counts regardless of Timer1 gate function									
bit 6	T1GPOL: Tir	T1GPOL: Timer1 Gate Polarity bit									
		1 = Timer1 gate is active-high (Timer1 counts when gate is high)									
	-	0 = Timer1 gate is active-low (Timer1 counts when gate is low)									
bit 5		T1GTM: Timer1 Gate Toggle Mode bit									
		 1 = Timer1 Gate Toggle mode is enabled 0 = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared 									
		Timer1 gate flip-flop toggles on every rising edge.									
bit 4		T1GSPM: Timer1 Gate Single-Pulse Mode bit									
		gate Single-Puls			ntrolling Timer1	gate					
bit 3		NE: Timer1 Gat			atus bit						
	1 = Timer1 g	gate single-puls	e acquisition is	ready, waiting	for an edge						
	-	gate single-puls		as completed of	or has not been	started					
bit 2		T1GVAL: Timer1 Gate Value Status bit									
		Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L. Unaffected by Timer1 Gate Enable (TMR1GE).									
bit 0	T1GSS<1:0>	T1GSS<1:0>: Timer1 Gate Source Select bits									
		ator 2 optionally									
		ator 1 optionally		output (C1_O	JT_sync)						
		01 =Timer0 overflow output (T0_overflow) 00 =Timer1 gate pin (T1G)									

REGISTER 22-2: 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
ANSELA	_	—	—	ANSA4	_	ANSA2	ANSA1	ANSA0	152
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
TMR1H	Holding Regi	ister for the M	ost Significan	t Byte of the 1	6-bit TMR1 C	ount			228*
TMR1L	Holding Regi	ister for the Le	east Significar	nt Byte of the	16-bit TMR1 (Count			228*
TMR3H	Holding Regi	ister for the M	ost Significan	t Byte of the 1	6-bit TMR3 C	ount			228*
TMR3L	Holding Regi	ister for the Le	east Significar	t Byte of the	16-bit TMR3 (Count			228*
TMR5H	Holding Regi	Holding Register for the Most Significant Byte of the 16-bit TMR5 Count							228*
TMR5L	Holding Regi	ister for the Le	east Significar	t Byte of the	16-bit TMR5 (Count			228*
TRISA	_	_	TRISA5	TRISA4	_(1)	TRISA2	TRISA1	TRISA0	151
T1CON	TMR1C	S<1:0>	T1CKP	S<1:0>	_	T1SYNC	_	TMR10N	232
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GS	S<1:0>	233
T3CON	TMR3C	S<1:0>	T3CKP	S<1:0>	_	T3SYNC	_	TMR3ON	232
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ DONE	T3GVAL	T3GS	S<1:0>	233
T5CON	TMR5C	S<1:0>	T5CKP	S<1:0>	—	T5SYNC	—	TMR5ON	232
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/ DONE	T5GVAL	T5GS	S<1:0>	233

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

Note 1: Unimplemented, read as '1'.

23.0 TIMER2/4/6 MODULE

The Timer2/4/6 modules are 8-bit timers that can operate as free-running period counters or in conjunction with external signals that control start, run, freeze, and reset operation in One-Shot and Monostable modes of operation. Sophisticated waveform control such as pulse density modulation are possible by combining the operation of these timers with other internal peripherals such as the comparators and CCP modules. Features of the timer include:

- 8-bit timer register
- · 8-bit period register
- · Selectable external hardware timer Resets
- Programmable prescaler (1:1 to 1:128)
- Programmable postscaler (1:1 to 1:16)
- Selectable synchronous/asynchronous operation
- · Alternate clock sources
- Interrupt-on-period

FIGURE 23-1: TIMER2 BLOCK DIAGRAM

- Three modes of operation:
 - Free Running Period
 - One-shot
 - Monostable

See Figure 23-1 for a block diagram of Timer2. See Figure 23-2 for the clock source block diagram.

Note: Three identical Timer2 modules are implemented on this device. The timers are named Timer2, Timer4, and Timer6. All references to Timer2 apply as well to Timer4 and Timer6. All references to T2PR apply as well to T4PR and T6PR.

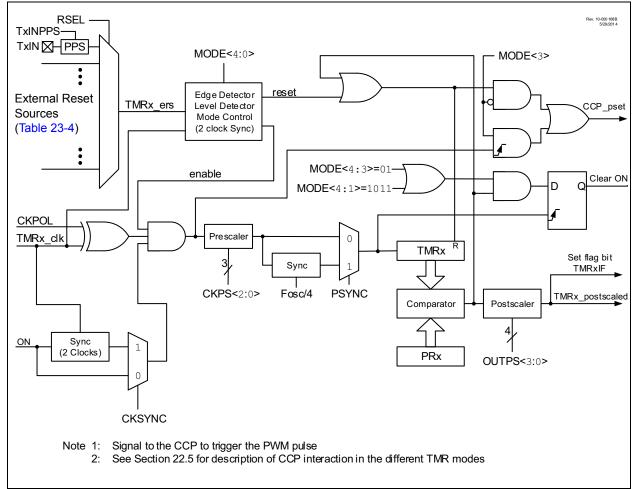
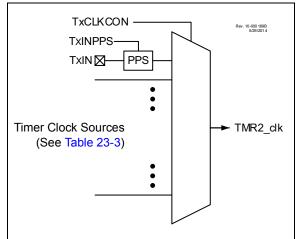


FIGURE 23-2: TIMER2 CLOCK SOURCE BLOCK DIAGRAM



23.1 Timer2 Operation

Timer2 operates in three major modes:

- Free Running Period
- One-shot
- Monostable

Within each mode there are several options for starting, stopping, and reset. Table 23-1 lists the options.

In all modes, the TMR2 count register is incremented on the rising edge of the clock signal from the programmable prescaler. When TMR2 equals T2PR, a high level is output to the postscaler counter. TMR2 is cleared on the next clock input.

An external signal from hardware can also be configured to gate the timer operation or force a TMR2 count Reset. In Gate modes the counter stops when the gate is disabled and resumes when the gate is enabled. In Reset modes the TMR2 count is reset on either the level or edge from the external source.

The TMR2 register is directly readable and writable. The TMR2 register is cleared on any device Reset. The T2PR register is double-buffered and initializes to 0xFF on any device Reset. The SFR is directly readable and writable, but the actual period buffer is only updated with the SFR value when the following events occur:

- · a write to the TMR2 register
- a write to the T2CON register
- · a write to the T2HLT register
- TMR2 = T2PR and the prescaler is full
- · External Reset Source event that resets the timer

Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- · any device Reset
- External Reset Source event that resets the timer.

Note:	TMR2	is	not	cleared	when	T2CON	is
	written.						

23.1.1 FREE RUNNING PERIOD MODE

The value of TMR2 is compared to that of the Period register, T2PR, on each clock cycle. When the two values match, the comparator resets the value of TMR2 to 00h on the next cycle and increments the output postscaler counter. When the postscaler count equals the value in the OUTPS<4:0> bits of the TMRxCON1 register then a one clock period wide pulse occurs on the TMR2_postscaled output, and the postscaler count is cleared.

23.1.2 ONE-SHOT MODE

The One-Shot mode is identical to the Free Running Period mode except that the ON bit is cleared and the timer is stopped when TMR2 matches T2PR and will not restart until the T2ON bit is cycled off and on. Postscaler OUTPS<4:0> values other than 0 are meaningless in this mode because the timer is stopped at the first period event and the postscaler is reset when the timer is restarted.

23.1.3 MONOSTABLE MODE

Monostable modes are similar to One-Shot modes except that the ON bit is not cleared and the timer can be restarted by an external Reset event.

23.2 PRx Period Register

The PRx period register is double buffered. Software reads and writes the PRx register. However, the timer uses a buffered PRx register for operation. Software does not have direct access to the buffered PRx register. The contents of the PRx register is transferred to the buffer by any of the following events:

- A write to the TMRx register
- · A write to the TMRxCON register
- When TMRx = PRx buffer and the prescaler rolls over
- An external Reset event

23.3 Timer2 Output

The Timer2 module's primary output is TMR2_postscaled, which pulses for a single TMR2_clk period when the postscaler counter matches the value in the OUTPS bits of the TMR2xCON register. The T2PR postscaler is incremented each time the TMR2 value matches the T2PR value. This signal can be selected as an input to several other input modules:

- The ADC module, as an Auto-conversion Trigger
- · CWG, as an auto-shutdown source

In addition, the Timer2 is also used by the CCP module for pulse generation in PWM mode. Both the actual TMR2 value as well as other internal signals are sent to the CCP module to properly clock both the period and pulse width of the PWM signal. See Section 26.4 "CCP/PWM Clock Selection" for more details on setting up Timer2 for use with the CCP, as well as the timing diagrams in Section 23.6 "Operation Examples" for examples of how the varying Timer2 modes affect CCP PWM output.

23.4 External Reset Sources

In addition to the clock source, the Timer2 also takes in an external Reset source. This external Reset source is selected for Timer2, Timer4, and Timer6 with the T2RST, T4RST, and T6RST registers, respectively. This source can control starting and stopping of the timer, as well as resetting the timer, depending on which mode the timer is in. The mode of the timer is controlled by the MODE<4:0> bits of the TMRxHLT register. Edge-Triggered modes require six Timer clock periods between external triggers. Level-Triggered modes require the triggering level to be at least three Timer clock periods long. External triggers are ignored while in Debug Freeze mode.

Mada	MODE	<4:0>	Output	Onerstien		Timer Control				
Mode	<4:3>	<2:0>	Operation	Operation	Start	Reset	Stop			
		000		Software gate (Figure 23-4)	ON = 1	_	ON = 0			
		001	Period Pulse	Hardware gate, active-high (Figure 23-5)	ON = 1 and TMRx_ers = 1	_	ON = 0 or TMRx_ers = 0			
		010	T uise	Hardware gate, active-low	ON = 1 and TMRx_ers = 0	_	ON = 0 or TMRx_ers = 2			
Free Running	00	011		Rising or falling edge Reset		TMRx_ers				
Period	00	100	Period	Rising edge Reset (Figure 23-6)		TMRx_ers ↑	ON = 0			
		101	Pulse	Falling edge Reset		TMRx_ers ↓				
		110	with Hardware	Low level Reset	ON = 1	TMRx_ers = 0	ON = 0 or TMRx_ers = 0			
		111	Reset	High level Reset (Figure 23-7)		TMRx_ers = 1	ON = 0 or TMRx_ers = 1			
		000	One-shot	Software start (Figure 23-8)	ON = 1					
		001	Edge	Rising edge start (Figure 23-9)	ON = 1 and TMRx_ers ↑	_				
	01	010	triggered start (Note 1) Edge triggered start and hardware Reset (Note 1)	Falling edge start	ON = 1 and TMRx_ers ↓		ON = 0 or Next clock after TMRx = PRx (Note 2)			
		011		Any edge start	ON = 1 and TMRx_ers	—				
One-shot		100		Rising edge start and Rising edge Reset (Figure 23-10)	ON = 1 and TMRx_ers ↑	TMRx_ers ↑				
		101		Falling edge start and Falling edge Reset	ON = 1 and TMRx_ers ↓	TMRx_ers \downarrow				
		110		Rising edge start and Low level Reset (Figure 23-11)	ON = 1 and TMRx_ers ↑	TMRx_ers = 0				
		111		Falling edge start and High level Reset	ON = 1 and TMRx_ers ↓	TMRx_ers = 1				
		000		Rese	rved		1			
		001	Edge	Rising edge start (Figure 23-12)	ON = 1 and TMRx_ers ↑		ON = 0 or			
Mono-stable		010	triggered start	Falling edge start	ON = 1 and TMRx_ers ↓	_	Next clock after			
		011	(Note 1)	Any edge start	ON = 1 and TMRx_ers	_	TMRx = PRx (Note 3)			
Reserved	10	100		Reserved						
Reserved		101	-	Rese	rved					
One-shot		110	Level triggered	High level start and Low level Reset (Figure 23-13)	ON = 1 and TMRx_ers = 1	TMRx_ers = 0	ON = 0 or			
		111	start and hardware Reset	Low level start & High level Reset	ON = 1 and TMRx_ers = 0	TMRx_ers = 1	Held in Rese (Note 2)			
Reserved	11	xxx		Rese	rved					

TABLE 23-1: TIMER2 OPERATING MODES

Note 1: If ON = 0 then an edge is required to restart the timer after ON = 1.

2: When TMRx = PRx then the next clock clears ON and stops TMRx at 00h.

3: When TMRx = PRx then the next clock stops TMRx at 00h but does not clear ON.

23.5 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches one of 16 postscale options (from 1:1 through 1:16), which are selected with the postscaler control bits, OUTPS<3:0> of the T2CON register. The interrupt is enabled by setting the TMR2IE interrupt enable bit of the PIE1 register. Interrupt timing is illustrated in Figure 23-3.



_		Rev. 10-000205A 4/7/2016
CKPS	0b010	
PRx	1	
OUTPS	060001	
TMRx_clk		
TMRx		0
TMRx_postscaled		1
TMRxIF	(1) (2)	(1)
Note 1: 2:	Setting the interrupt flag is synchronized with the instruction clock. Synchronization may take as many as 2 instruction cycles Cleared by software.	

23.6 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the TxCON register are cleared).
- The diagrams illustrate any clock except Fosc/4 and show clock-sync delays of at least two full cycles for both ON and Timer2_ers. When using Fosc/4, the clocksync delay is at least one instruction period for Timer2_ers; ON applies in the next instruction period.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in Section 26.4 "CCP/PWM Clock Selection". The signals are not a part of the Timer2 module.

23.6.1 SOFTWARE GATE MODE

This mode corresponds to legacy Timer2 operation. The timer increments with each clock input when ON = 1 and does not increment when ON = 0. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 23-4. With PRx = 5, the counter advances until TMRx = 5, and goes to zero with the next clock.



	Rev. 10-0001958 5002014
MODE	0Ь00000
TMRx_clk	
Instruction ⁽¹⁾ -	BSF BSF BSF
ON	
PRx	5
TMRx	0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 2
TMRx_postscaled	
PWM Duty Cycle	3
PWM Output	
	BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

23.6.2 HARDWARE GATE MODE

The Hardware Gate modes operate the same as the Software Gate mode except the TMRx_ers external signal can also gate the timer. When used with the CCP the gating extends the PWM period. If the timer is stopped when the PWM output is high then the duty cycle is also extended.

When MODE<4:0> = 00001 then the timer is stopped when the external signal is high. When MODE<4:0> = 00010 then the timer is stopped when the external signal is low.

Figure 23-5 illustrates the Hardware Gating mode for MODE<4:0> = 00001 in which a high input level starts the counter.

FIGURE 23-5:	HARDWARE GATE MODE TIMING DIAGRAM (MODE = 00001)	
1 IGUNE 23-J.	HARDWARE GATE MODE HIMING DIAGRAM	WODL = 00001	

	Rev: 10.000 1968 5000001 4	
MODE	0b00001	
TMRx_clk		
TMRx_ers		
PRx	5	
TMRx	$0 \qquad \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2$	
TMRx_postscaled		
PWM Duty Cycle PWM Output	3	

23.6.3 EDGE-TRIGGERED HARDWARE LIMIT MODE

In Hardware Limit mode the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge
- (MODE<4:0>= 00011)
- Reset on rising edge (MODE<4:0> = 00100)
- Reset on falling edge (MODE<4:0> = 00101)

When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two-clock delay. Refer to Figure 23-6.

FIGURE 23-6: EDGE-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE = 00100)

	Rev. 10-000 1978 5300201 4	
MODE	0b00100	
TMRx_clk		
PRx	5	
Instruction ⁽¹⁾ -	BSF BSF	
ON		
TMRx_ers		
TMRx	0 \ 1 \ 2 \ \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 0 \ 1	
TMRx_postscaled		
PWM Duty Cycle PWM Output	3	
Note 1 s	: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the et or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input	CPU to ut.

23.6.4 LEVEL-TRIGGERED HARDWARE LIMIT MODE

In the Level-Triggered Hardware Limit Timer modes the counter is reset by high or low levels of the external signal TMRx_ers, as shown in Figure 23-7. Selecting MODE<4:0> = 0.0110 will cause the timer to reset on a low level external signal. Selecting MODE<4:0> = 0.0111 will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMRx_ers = 1. ON is controlled by BSF and BCF instructions. When ON = 0 the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the PRx value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting, and the PWM output is set high, on either the clock following the PRx match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.



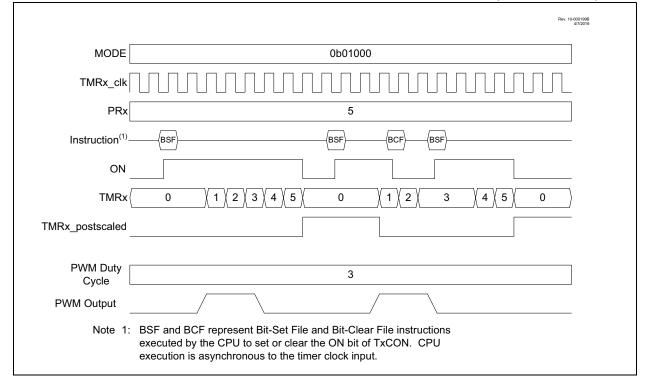
	Rev. 10-0001988 505/0014
MODE	0b00111
TMRx_clk	
PRx	5
Instruction ⁽¹⁾	(BSF)_(BSF)(BSF)_(
ON	
TMRx_ers	
TMRx	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $
TMRx_postscaled	
PWM Duty Cycle	3
PWM Output	
	1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

23.6.5 SOFTWARE START ONE-SHOT MODE

In One-Shot mode the timer resets and the ON bit is cleared when the timer value matches the PRx period value. The ON bit must be set by software to start another timer cycle. Setting MODE<4:0> = 01000 selects One-Shot mode which is illustrated in Figure 23-8. In the example, ON is controlled by BSF and BCF instructions. In the first case, a BSF instruction sets ON and the counter runs to completion and clears ON. In the second case, a BSF instruction starts the cycle, BCF/BSF instructions turn the counter off and on during the cycle, and then it runs to completion.

When One-Shot mode is used in conjunction with the CCP PWM operation the PWM pulse drive starts concurrent with setting the ON bit. Clearing the ON bit while the PWM drive is active will extend the PWM drive. The PWM drive will terminate when the timer value matches the CCPRx pulse width value. The PWM drive will remain off until software sets the ON bit to start another cycle. If software clears the ON bit after the CCPRx match but before the PRx match then the PWM drive will be extended by the length of time the ON bit remains cleared. Another timing cycle can only be initiated by setting the ON bit after it has been cleared by a PRx period count match.

FIGURE 23-8: SOFTWARE START ONE-SHOT MODE TIMING DIAGRAM (MODE = 01000)



23.6.6 EDGE-TRIGGERED ONE-SHOT MODE

The Edge-Triggered One-Shot modes start the timer on an edge from the external signal input, after the ON bit is set, and clear the ON bit when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = 01001)
- Falling edge (MODE<4:0> = 01010)
- Rising or Falling edge (MODE<4:0> = 01011)

If the timer is halted by clearing the ON bit then another TMRx_ers edge is required after the ON bit is set to resume counting. Figure 23-9 illustrates operation in the rising edge One-Shot mode.

When Edge-Triggered One-Shot mode is used in conjunction with the CCP then the edge-trigger will activate the PWM drive and the PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated when the timer halts at the PRx period count match.

FIGURE 23-9: EDGE-TRIGGERED ONE-SHOT MODE TIMING DIAGRAM (MODE = 01001)

MODE	0b01001
TMRx_clk	
PRx	5
Instruction ⁽¹⁾ -	(BSF) (BCF)
ON	
TMRx_ers	
TMRx	$0 \qquad 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \qquad 1 \\ 2$
CCP_pset	
TMRx_postscaled _	
PWM Duty Cycle	3
PWM Output	

23.6.7 EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE

In Edge-Triggered Hardware Limit One-Shot modes the timer starts on the first external signal edge after the ON bit is set and resets on all subsequent edges. Only the first edge after the ON bit is set is needed to start the timer. The counter will resume counting automatically two clocks after all subsequent external Reset edges. Edge triggers are as follows:

- Rising edge start and Reset (MODE<4:0> = 01100)
- Falling edge start and Reset (MODE<4:0> = 01101)

The timer resets and clears the ON bit when the timer value matches the PRx period value. External signal edges will have no effect until after software sets the ON bit. Figure 23-10 illustrates the rising edge hardware limit one-shot operation.

When this mode is used in conjunction with the CCP then the first starting edge trigger, and all subsequent Reset edges, will activate the PWM drive. The PWM drive will deactivate when the timer matches the CCPRx pulse-width value and stay deactivated until the timer halts at the PRx period match unless an external signal edge resets the timer before the match occurs.

FIGURE 23-10: EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE = 01100)

MODE 0b01100 TMRx_clk 5 PRx Instruction⁽¹⁾ BSF BSF ON TMRx_ers TMRx 0 1 2 \ 3 \ 4 \ 5 0 1) 2 0 2) 3 🛛 4 ≬ 5 1 TMRx_postscaled PWM Duty 3 Cycle PWM Output 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to Note set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

23.6.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

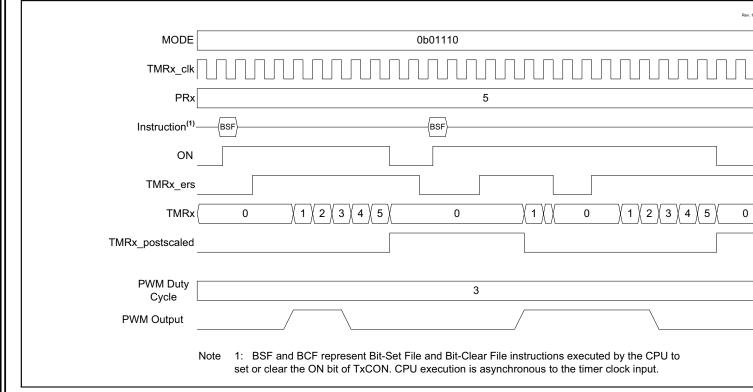
In Level -Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = 01110)
- High Reset level (MODE<4:0> = 01111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.





23.6.9 EDGE-TRIGGERED MONOSTABLE MODES

The Edge-Triggered Monostable modes start the timer on an edge from the external Reset signal input, after the ON bit is set, and stop incrementing the timer when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = 10001)
- Falling edge (MODE<4:0> = 10010)
- Rising or Falling edge (MODE<4:0> = 10011)

When an Edge-Triggered Monostable mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external Reset signal edge that starts the timer, but will not go active when the timer matches the PRx value. While the timer is incrementing, additional edges on the external Reset signal will not affect the CCP PWM.

FIGURE 23-12: RISING EDGE-TRIGGERED MONOSTABLE MODE TIMING DIAGRAM (MODE = 10001)

MODE	0b10001
TMRx_clk	
PRx	5
Instruction ⁽¹⁾	BSF BSF BCF
ON	
TMRx_ers	
TMRx	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$
TMRx_postscaled	
PWM Duty Cycle	3
PWM Output	
	1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

23.6.10 LEVEL-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

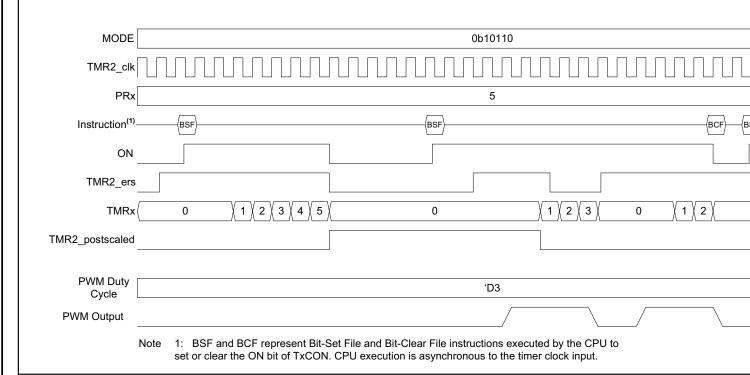
The Level-Triggered Hardware Limit One-Shot modes hold the timer in Reset on an external Reset level and start counting when both the ON bit is set and the external signal is not at the Reset level. If one of either the external signal is not in Reset or the ON bit is set then the other signal being set/made active will start the timer. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = 10110)
- High Reset level (MODE<4:0> = 10111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control the timer will stay in Reset until both the ON bit is set and the external signal is not at the Reset level.

When Level-Triggered Hardware Limit One-Shot modes are used in conjunction with the CCP PWM operation the PWM drive goes active with either the external signal edge or the setting of the ON bit, whichever of the two starts the timer.

FIGURE 23-13: LEVEL-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE = 10110)



23.7 Timer2 Operation During Sleep

When PSYNC = 1, Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and T2PR registers will remain unchanged while processor is in Sleep mode.

When PSYNC = 0, Timer2 will operate in Sleep as long as the clock source selected is also still running. Selecting the LFINTOSC, MFINTOSC, or HFINTOSC oscillator as the timer clock source will keep the selected oscillator running during Sleep.

23.8 Register Definitions: Timer2/4/6 Control

Long bit name prefixes for the Timer2/4/6 peripherals are shown in Table 23-2. Refer to Section 1.1.2.2 "Long Bit Names" for more information TABLE 23-2:

Peripheral	Bit Name Prefix
Timer2	T2
Timer4	T4
Timer6	T6

REGISTER 23-1: TxCLKCON: TIMERx CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
	—	—	_		CS<	3:0>	
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 3-0 **CS<3:0>:** Timerx Clock Selection bits See Table 23-3.

TABLE 23-3: TIMERX CLOCK SOURCES

CS<3:0>	Timer2 Timer4		Timer6
1101-1111	Reserved	Reserved	Reserved
1011	AT1_perclk	AT1_perclk	AT1_perclk
1010	LC4_out	LC4_out	LC4_out
1001	LC3_out	LC3_out	LC3_out
1000	LC2_out	LC2_out	LC2_out
0111	LC1_out	LC1_out	LC1_out
0110	Pin selected by T2INPPS	Pin selected by T2INPPS	Pin selected by T2INPPS
0101	MFINTOSC 31.25 kHz	MFINTOSC 31.25 kHz	MFINTOSC 31.25 kHz
0100	ZCD1_output	ZCD1_output	ZCD1_output
0011	LFINTOSC	LFINTOSC	LFINTOSC
0010	HFINTOSC 16 MHz	HFINTOSC 16 MHz	HFINTOSC 16 MHz
0001	Fosc	Fosc	Fosc
0000	Fosc/4	Fosc/4	Fosc/4

R/W/HC-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	
0N ⁽¹⁾	(1) CKPS<2:0>			OUTP	S<3:0>			
bit 7							bit 0	
Legend:								
R = Readable		W = Writable I		-	nented bit, rea			
u = Bit is unch	anged	x = Bit is unkn				R/Value at all	other Resets	
'1' = Bit is set		'0' = Bit is clea	ared	HC = Bit is cle	eared by hardv	vare		
hit 7		On hit						
bit 7	ON: Timerx							
	1 = Timerx 0 = Timerx	is off: all counter	rs and state m	nachines are re	set			
bit 6-4		: Timer2-type Cl						
	111 = 1 : 128	• •						
	110 = 1:64	Prescaler						
	101 = 1:32 Prescaler							
	100 = 1:16 Prescaler							
	011 = 1:8 P							
	010 = 1:4 P 001 = 1:2 P							
	001 = 1.2 P							
bit 3-0	OUTPS<3:0	>: Timerx Outpu	t Postscaler S	Select bits				
	1111 = 1:16							
	1110 = 1:15							
	1101 = 1:14							
	1100 = 1:13 1011 = 1:12							
	1011 - 1.12 1010 - 1:11							
	1001 = 1:10							
	1000 = 1:9 F							
	0111 = 1:8 F	Postscaler						
	0110 = 1:7 F							
	0101 = 1:6 F							
	0100 = 1:5 F 0011 = 1:4 F							
	0010 = 1.3 F							
	0001 = 1:2 F							
	0000 = 1:1 F	De este e el e e						

REGISTER 23-2: TxCON: TIMERx CONTROL REGISTER

Note 1: In certain modes, the ON bit will be auto-cleared by hardware. See Section 23.6 "Operation Examples".

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
PSYNC ^{(1,}	²⁾ CKPOL ⁽³⁾	CKSYNC ^(4, 5)			MODE<4:0>(6, 7)		
bit 7	÷						bit 0
Legend:						(0)	
R = Readabl		W = Writable bit			ented bit, read as		-
u = Bit is und	0	x = Bit is unknov		-n/n = Value at	POR and BOR/V	alue at all other	Resets
'1' = Bit is se	et	'0' = Bit is cleare	ed				
bit 7	1 = TMRx Pr	rx Prescaler Synch escaler Output is s escaler Output is n	synchronized to	o Fosc/4			
bit 6	1 = Falling ed	rx Clock Polarity S dge of input clock o lge of input clock c	clocks timer/pr				
bit 5	1 = ON regist	nerx Clock Synchro ter bit is synchroniz ter bit is not synch	zed to TMR2_	clk input			
bit 4-0	MODE<4:0>: See Table 23-1	Timerx Control Mo	de Selection b	_{bits} (6, 7)			
Note 1:	Setting this bit ens	ures that reading 1	MRx will retui	rn a valid value.			
2:	When this bit is '1'	, Timer2 cannot op	erate in Sleep	mode.			
3:	CKPOL should not	t be changed while	ON = 1.				
4:	Setting this bit ens	ures glitch-free op	eration when t	he ON is enabled	or disabled.		
5:	When this bit is set	t then the timer op	eration will be	delayed by two T	MRx input clocks	after the ON bit	is set.
6:	Unless otherwise in of TMRx).	ndicated, all modes	s start upon ON	N = 1 and stop upo	on ON = 0 (stops o	occur without aff	ecting the value

REGISTER 23-3: TxHLT: TIMERx HARDWARE LIMIT CONTROL REGISTER

7: When TMRx = PRx, the next clock clears TMRx, regardless of the operating mode.

REGISTER 23-4: TxRST: TIMERx EXTERNAL RESET SIGNAL SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
	—	—	—		RSEL	.<3:0>	
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 7-4 Unimplemented: Read as '0'

bit 3-0 **RSEL<4:0>:** TimerX External Reset Signal Source Selection bits See Table 23-4.

TABLE 23-4: EXTERNAL RESET SOURCES

RSEL<4:0>	Timer2	Timer4	Timer6
1111	Reserved	Reserved	Reserved
1110	PWM4_out	PWM4_out	PWM4_out
1101	PWM3_out	PWM3_out	PWM3_out
1100	LC4_out	LC4_out	LC4_out
1011	LC3_out	LC3_out	LC3_out
1010	LC2_out	LC2_out	LC2_out
1001	LC1_out	LC1_out	LC1_out
1000	ZCD1_out	ZCD1_out	ZCD1_out
0111	TMR6_postscaled	TMR6_postscaled	Reserved
0110	TMR4_postscaled	Reserved	TMR4_postscaled
0101	Reserved	TMR2_postscaled	TMR2_postscaled
0100	CCP2_out	CCP2_out	CCP2_out
0011	CCP1_out	CCP1_out	CCP1_out
0010	C2OUT_sync	C2OUT_sync	C2OUT_sync
0001	C1OUT_sync	C1OUT_sync	C1OUT_sync
0000	Pin selected by T2INPPS	Pin selected by T2INPPS	Pin selected by T2INPPS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	EN	_	OUT	FMT		MODE	=<3:0>		354
CCP2CON	EN	— OUT FMT			MODE<3:0>				354
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	_	_	_	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
PR2	Timer2 Modu	Timer2 Module Period Register							244*
TMR2	Holding Reg	ister for the 8	-bit TMR2 Re	gister					236*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				256
T2CLKCON	—	_	_	_	CS<3:0>			255	
T2RST	—				RSEL<3:0>				258
T2HLT	PSYNC	CKPOL CKSYNC MODE<4:0>					257		
PR4	Timer4 Mode	ner4 Module Period Register					244*		
TMR4	Holding Register for the 8-bit TMR4 Register						236*		
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				256
T4CLKCON	_	_	_	_		CS<	<3:0>		255
T4RST	—		_	_		RSEL	_<3:0>		258
T4HLT	PSYNC	CKPOL CKSYNC			MODE<4:0>				257
PR6	Timer6 Module Period Register						244*		
TMR6	Holding Register for the 8-bit TMR6 Register						236*		
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				256
T6CLKCON	—		_		— T6CS<2:0>			255	
T6RST	—	_		_	RSEL<3:0>			258	
T6HLT	PSYNC	CKPOL	CKSYNC		MODE<4:0>			257	

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

* Page provides register information.

24.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

24.1 MSSP Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

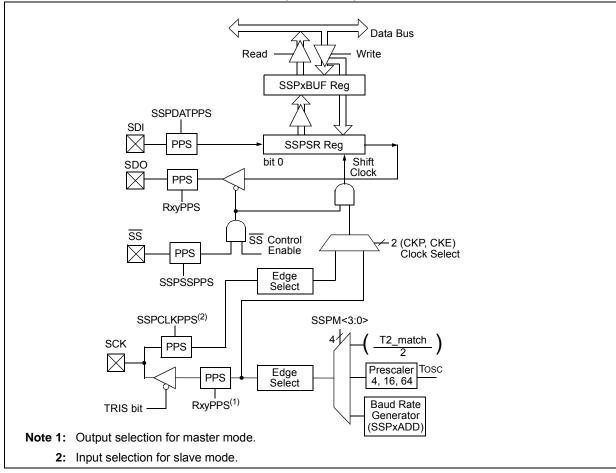
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- · Daisy-chain connection of slave devices

Figure 24-1 is a block diagram of the SPI interface module.

FIGURE 24-1: MSSP BLOCK DIAGRAM (SPI MODE)

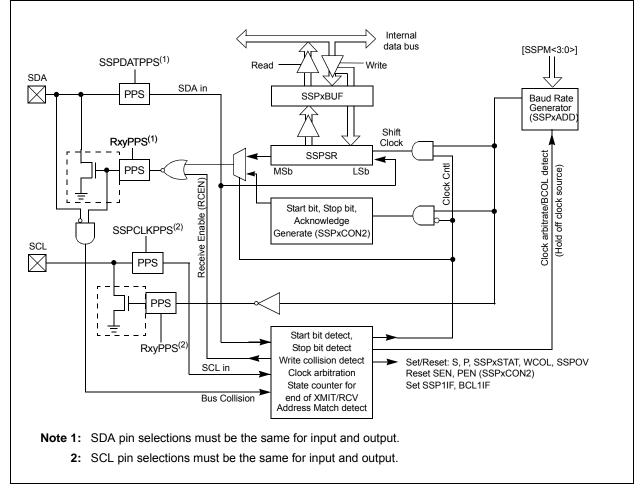


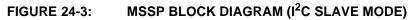
The I^2C interface supports the following modes and features:

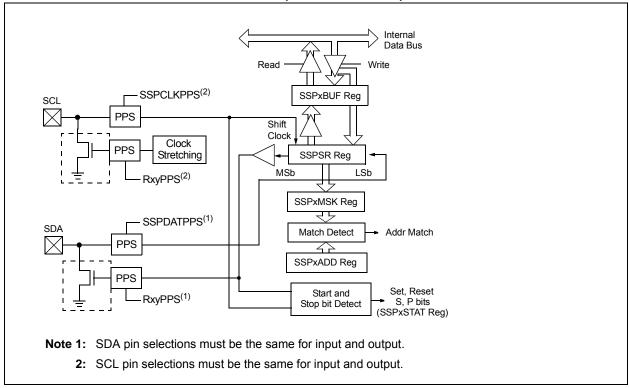
- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- · 7-bit and 10-bit addressing
- Start and Stop interrupts
- · Interrupt masking
- Clock stretching
- · Bus collision detection
- General call address matching
- Address masking
- · Address Hold and Data Hold modes
- · Selectable SDA hold times

Figure 24-2 is a block diagram of the I^2C interface module in Master mode. Figure 24-3 is a diagram of the I^2C interface module in Slave mode.









24.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select (SS)

Figure 24-1 shows the block diagram of the MSSP module when operating in SPI mode.

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

Figure 24-4 shows a typical connection between a master device and multiple slave devices.

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

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 24-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on

its SDO pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

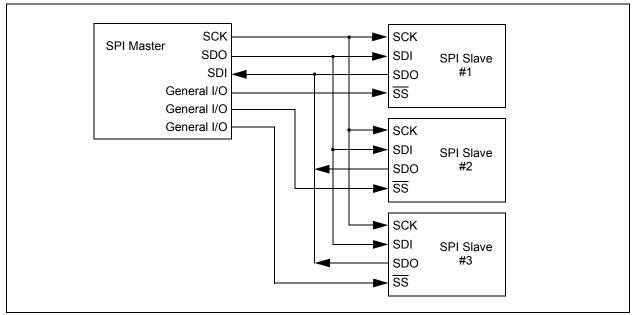
Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

FIGURE 24-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION



24.2.1 SPI MODE REGISTERS

The MSSP module has five registers for SPI mode operation. These are:

- MSSP STATUS register (SSPxSTAT)
- MSSP Control register 1 (SSPxCON1)
- MSSP Control register 3 (SSPxCON3)
- MSSP Data Buffer register (SSPxBUF)
- MSSP Address register (SSPxADD)
- MSSP Shift register (SSPSR) (Not directly accessible)

SSPxCON1 and SSPxSTAT are the control and STATUS registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

In one SPI master mode, SSPxADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in Section 24.7 "Baud Rate Generator".

SSPSR is the shift register used for shifting data in and out. SSPxBUF provides indirect access to the SSPSR register. SSPxBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPSR and SSPxBUF together create a buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not buffered. A write to SSPxBUF will write to both SSPxBUF and SSPSR.

24.2.2 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- · Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- · Slave Select mode (Slave mode only)

To enable the serial port, SSP Enable bit, SSPEN of the SSPxCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCONx registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and \overline{SS} pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- · SDI must have corresponding TRIS bit set
- · SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding
 TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- SS must have corresponding TRIS bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPxBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full Detect bit, BF of the SSPxSTAT register, and the interrupt flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the write collision detect bit WCOL of the SSPxCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF of the SSPxSTAT register, indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPxBUF register. Additionally, the SSPxSTAT register indicates the various Status conditions.

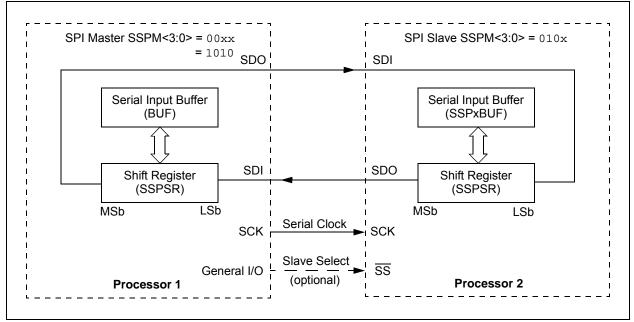


FIGURE 24-5: SPI MASTER/SLAVE CONNECTION

24.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 24-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set). The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 24-6, Figure 24-8, Figure 24-9 and Figure 24-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

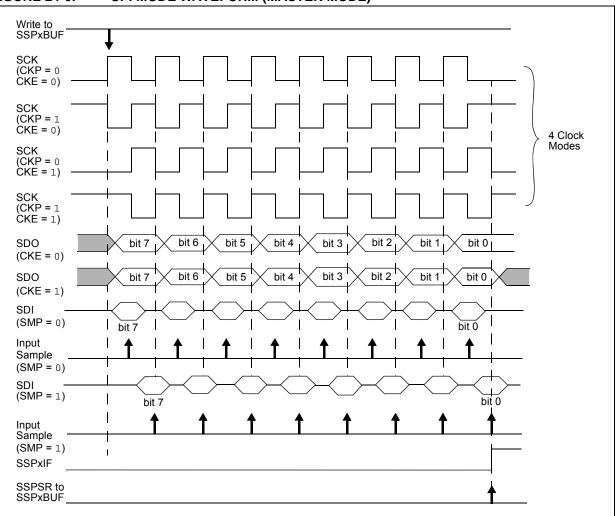
- · Fosc/4 (or Tcy)
- Fosc/16 (or 4 * Tcy)
- Fosc/64 (or 16 * Tcy)
- Timer2 output/2
- Fosc/(4 * (SSPxADD + 1))

Figure 24-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

Note: In Master mode the clock signal output to the SCK pin is also the clock signal input to the peripheral. The pin selected for output with the RxyPPS register must also be selected as the peripheral input with the SSPCLKPPS register.





24.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

24.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisychain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisychain feature only requires a single Slave Select line from the master device.

Figure 24-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

24.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The \overline{SS} pin allows a Synchronous Slave mode. The SPI must be in Slave mode with \overline{SS} pin control enabled (SSPxCON1<3:0> = 0100).

FIGURE 24-7: SPI DAISY-CHAIN CONNECTION

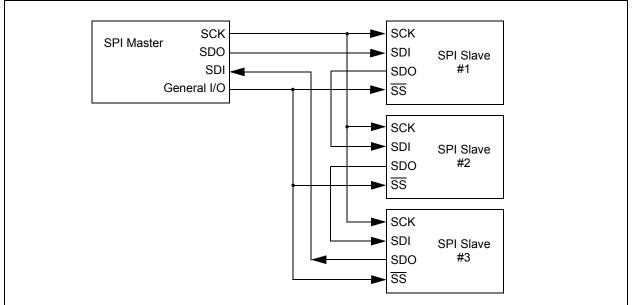
When the \overline{SS} pin is low, transmission and reception are enabled and the SDO pin is driven.

When the \overline{SS} pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

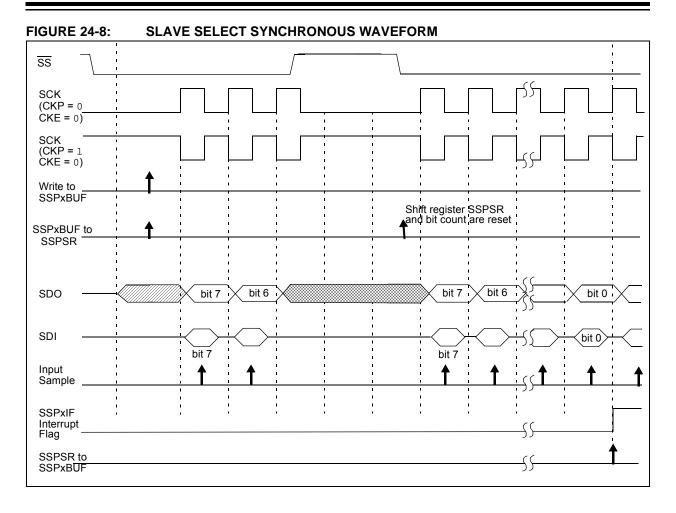
Note 1:	When the SPI is in Slave mode with \overline{SS} pin					
	control enabled (SSPxCON1<3:0> =					
	0100), the SPI module will reset if the \overline{SS}					
	pin is set to VDD.					

- 2: When the SPI is used in Slave mode with CKE set; the user must enable SS pin control.
- While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the \overline{SS} pin to a high level or clearing the SSPEN bit.



PIC16(L)F1615/9





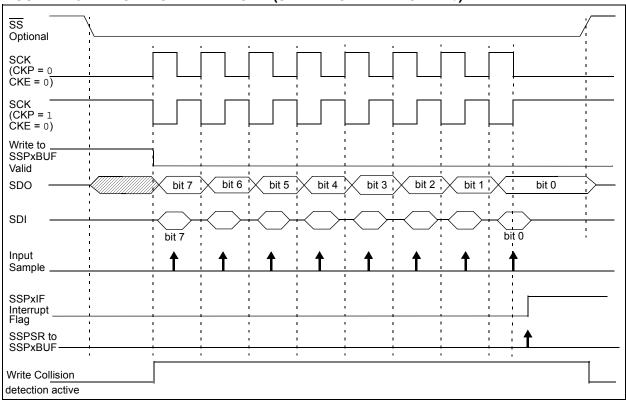
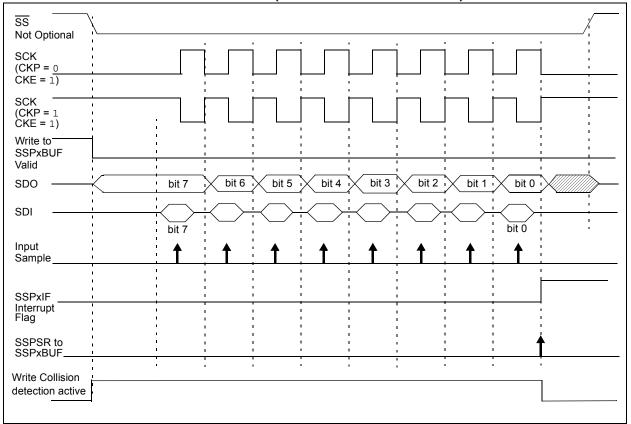


FIGURE 24-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



24.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, module clocks may be operating at a different speed than when in Full-Power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSP clock is much faster than the system clock.

In Slave mode, when MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSP interrupts should be disabled.

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/ reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	_	—	_	ANSA4	—	ANSA2	ANSA1	ANSA0	152
ANSELC	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	_	_	ANSC3	ANSC2	ANSC1	ANSC0	166
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RxyPPS	_	_	_	RxyPPS<4:0>				172	
SSPCLKPPS	_	_	_	SSPCLKPPS<4:0>					174, 172
SSPDATPPS	_	_	_	SSPDATPPS<4:0>				174, 172	
SSPSSPPS	_	_	_	SSPSSPPS<4:0>				174, 172	
SSP1BUF	Synchronous Serial Port Receive Buffer/Transmit Register						264*		
SSP1CON1	WCOL	SSPOV	SSPEN	CKP SSPM<3:0>					309
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	308
SSP1STAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	308
TRISA	_	_	TRISA5	TRISA4	_(1)	TRISA2	TRISA1	TRISA0	151
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165

TABLE 24-1: SUMMARY OF REGISTERS ASSOCIATED WITH SPI OPERATION

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP in SPI mode. * Page provides register information.

Note 1: Unimplemented, read as '1'.

2: PIC16(L)F1619 only.

24.3 I²C MODE OVERVIEW

The Inter-Integrated Circuit (I^2C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

The I²C bus specifies two signal connections:

- · Serial Clock (SCL)
- Serial Data (SDA)

Figure 24-11 shows the block diagram of the MSSP module when operating in I²C mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 24-11 shows a typical connection between two processors configured as master and slave devices.

The I²C bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

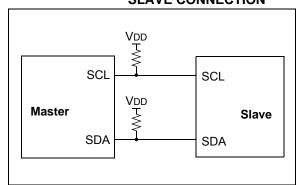
- Master Transmit mode (master is transmitting data to a slave)
- Master Receive mode (master is receiving data from a slave)
- Slave Transmit mode (slave is transmitting data to a master)
- Slave Receive mode (slave is receiving data from the master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an ACK. The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDA line while the SCL line is held high. Address and data bytes are sent out, Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

FIGURE 24-11: I²C MASTER/ SLAVE CONNECTION



The Acknowledge bit (\overline{ACK}) is an active-low signal, which holds the SDA line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of a data bit is always performed while the SCL line is held low. Transitions that occur while the SCL line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, then it repeatedly sends out a byte of data, with the slave responding after each byte with an \overline{ACK} bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, then it repeatedly receives a byte of data from the slave, and responds after each byte with an ACK bit. In this example, the master device is in Master Receive mode and the slave is Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last ACK bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last ACK bit when it is in receive mode.

The I²C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.

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

24.3.1 CLOCK STRETCHING

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

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

24.3.2 ARBITRATION

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

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

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

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

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

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

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

24.4 I²C MODE OPERATION

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC[®] microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

24.4.1 BYTE FORMAT

All communication in I^2C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

24.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I^2C communication that have definitions specific to I^2C . That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I^2C specification.

24.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

- Note 1: Data is tied to output zero when an I²C mode is enabled.
 - 2: Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

24.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 24-2: I²C BUS TERMS

TABLE 24-2:	I ² C BUS TERMS				
TERM	Description				
Transmitter	The device which shifts data out onto the bus.				
Receiver	The device which shifts data in from the bus.				
Master	The device that initiates a transfer, generates clock signals and termi- nates a transfer.				
Slave	The device addressed by the master.				
Multi-master	A bus with more than one device that can initiate data transfers.				
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.				
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.				
Idle	No master is controlling the bus, and both SDA and SCL lines are high.				
Active	Any time one or more master devices are controlling the bus.				
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.				
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSPxADD.				
Write Request	Slave receives a matching address with R/W bit clear, and is ready to clock in data.				
Read Request	Master sends an address byte with the R/\overline{W} bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.				
Clock Stretching	When a device on the bus holds SCL low to stall communication.				
Bus Collision	Any time the SDA line is sampled low by the module while it is out- putting and expected high state.				

© 2014-2017 Microchip Technology Inc.

24.4.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 24-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I^2C Specification that states no bus collision can occur on a Start.

24.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

Note: At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

24.4.7 RESTART CONDITION

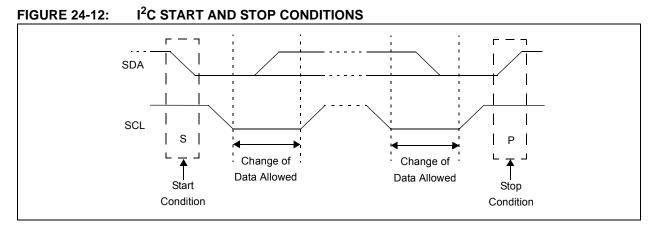
A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 24-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/W bit set. The slave logic will then hold the clock and prepare to clock out data.

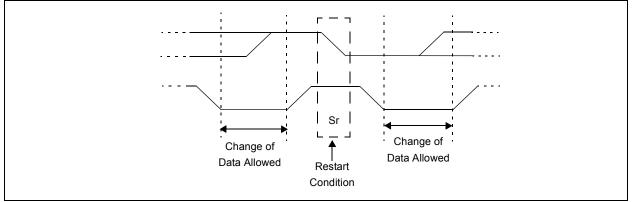
After a full match with R/\overline{W} clear in 10-bit mode, a prior match flag is set and maintained until a Stop condition, a high address with R/\overline{W} clear, or high address match fails.

24.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.







24.4.9 ACKNOWLEDGE SEQUENCE

The 9th SCL pulse for any transferred byte in I^2C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge (ACK) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an \overline{ACK} is placed in the ACKSTAT bit of the SSPxCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the ACK value sent back to the transmitter. The ACKDT bit of the SSPxCON2 register is set/cleared to determine the response.

Slave hardware will generate an ACK response if the AHEN and DHEN bits of the SSPxCON3 register are clear.

There are certain conditions where an \overline{ACK} will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the eighth falling edge of SCL on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

24.5 I²C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected by the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

24.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 24-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 24-5) affects the address matching process. See **Section 24.5.8 "SSP Mask Register**" for more information.

24.5.1.1 I²C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

24.5.1.2 I²C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/W bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

24.5.2 SLAVE RECEPTION

When the R/\overline{W} bit of a matching received address byte is clear, the R/\overline{W} bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPxSTAT register is set, or bit SSPOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 24-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSPxIF, must be cleared by software.

When the SEN bit of the SSPxCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPxCON1 register, except sometimes in 10-bit mode. See Section 24.5.6.2 "10bit Addressing Mode" for more detail.

24.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I^2C slave in 7-bit Addressing mode. Figure 24-14 and Figure 24-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I^2C communication.

- 1. Start bit detected.
- 2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- 3. Matching address with R/W bit clear is received.
- 4. The slave pulls SDA low sending an ACK to the master, and sets SSPxIF bit.
- 5. Software clears the SSPxIF bit.
- 6. Software reads received address from SSPxBUF clearing the BF flag.
- 7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
- 8. The master clocks out a data byte.
- Slave drives SDA low sending an ACK to the master, and sets SSPxIF bit.
- 10. Software clears SSPxIF.
- 11. Software reads the received byte from SSPxBUF clearing BF.
- 12. Steps 8-12 are repeated for all received bytes from the master.
- 13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes idle.

24.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the eighth falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to ACK the receive address or data byte, rather than the hardware. This functionality adds support for PMBus[™] that was not present on previous versions of this module.

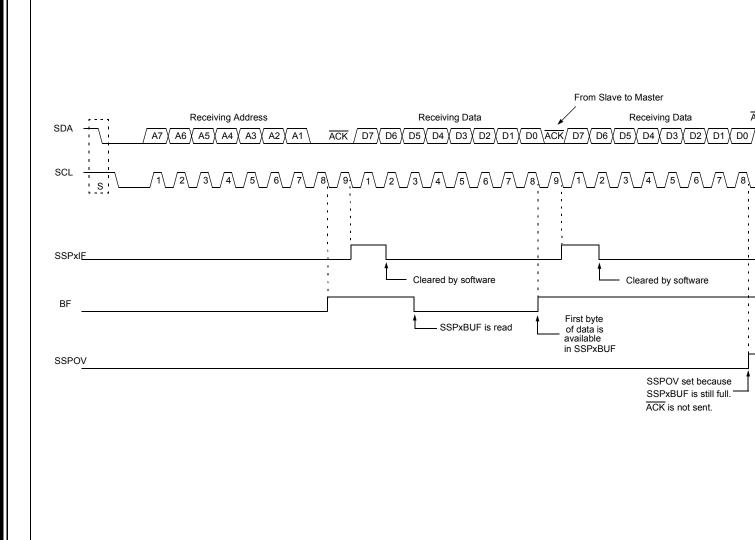
This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 24-16 displays a module using both address and data holding. Figure 24-17 includes the operation with the SEN bit of the SSPxCON2 register set.

- 1. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- Matching address with R/W bit clear is clocked in. SSPxIF is set and CKP cleared after the eighth falling edge of SCL.
- 3. Slave clears the SSPxIF.
- Slave can look at the ACKTIM bit of the SSPxCON3 register to determine if the SSPxIF was after or before the ACK.
- 5. Slave reads the address value from SSPxBUF, clearing the BF flag.
- 6. Slave sets ACK value clocked out to the master by setting ACKDT.
- 7. Slave releases the clock by setting CKP.
- 8. SSPxIF is set after an \overline{ACK} , not after a NACK.
- 9. If SEN = 1 the slave hardware will stretch the clock after the ACK.
- 10. Slave clears SSPxIF.

Note: SSPxIF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSPxIF not set

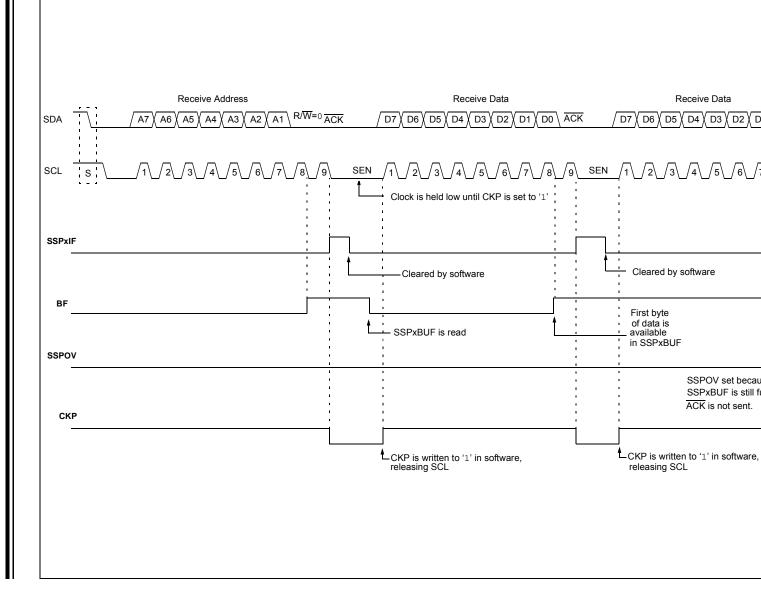
- 11. SSPxIF set and CKP cleared after eighth falling edge of SCL for a received data byte.
- 12. Slave looks at ACKTIM bit of SSPxCON3 to determine the source of the interrupt.
- 13. Slave reads the received data from SSPxBUF clearing BF.
- 14. Steps 7-14 are the same for each received data byte.
- 15. Communication is ended by either the slave sending an ACK = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

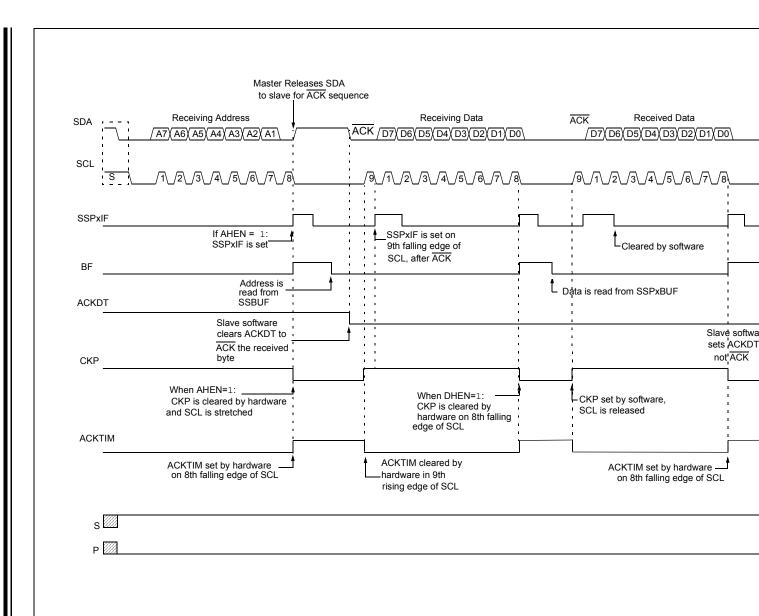




DS40001770D-page 278

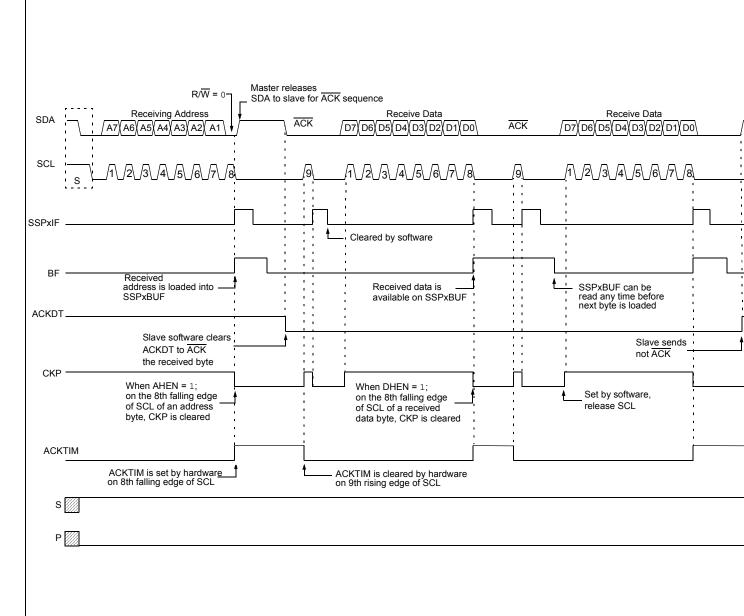












111 2017 Microchin Tochno

24.5.3 SLAVE TRANSMISSION

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an ACK pulse is sent by the slave on the ninth bit.

Following the ACK, slave hardware clears the CKP bit and the SCL pin is held low (see Section 24.5.6 "Clock Stretching" for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit of the SSPxCON1 register. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The \overline{ACK} pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This \overline{ACK} value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not \overline{ACK}), then the data transfer is complete. In this case, when the not \overline{ACK} is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low (\overline{ACK}), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

24.5.3.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCL1IF bit of the PIR2 register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. User software can use the BCL1IF bit to handle a slave bus collision.

24.5.3.2 7-bit Transmission

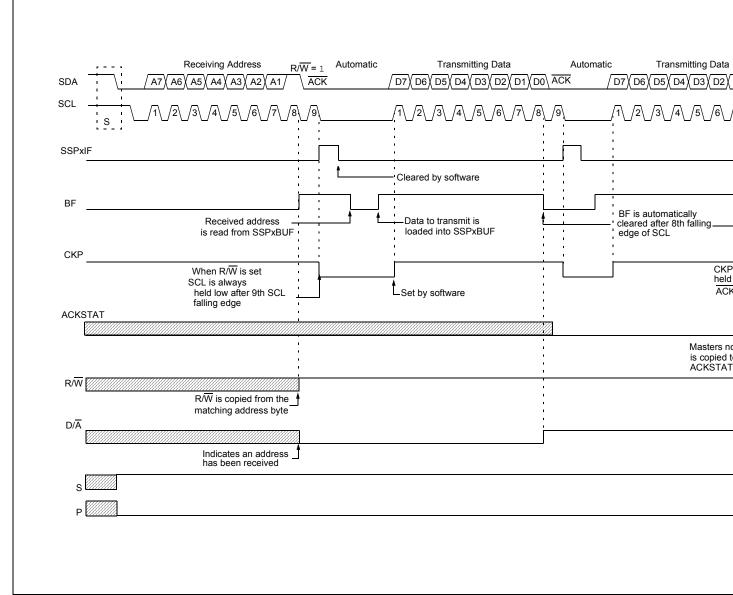
A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 24-18 can be used as a reference to this list.

- 1. Master sends a Start condition on SDA and SCL.
- 2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- 3. Matching address with R/W bit set is received by the Slave setting SSPxIF bit.
- 4. Slave hardware generates an ACK and sets SSPxIF.
- 5. SSPxIF bit is cleared by user.
- 6. Software reads the received address from SSPxBUF, clearing BF.
- 7. R/\overline{W} is set so CKP was automatically cleared after the ACK.
- 8. The slave software loads the transmit data into SSPxBUF.
- 9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
- 10. SSPxIF is set after the ACK response from the master is loaded into the ACKSTAT register.
- 11. SSPxIF bit is cleared.
- 12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master ACKs the clock will be stretched.

- ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.
- 13. Steps 9-13 are repeated for each transmitted byte.
- 14. If the master sends a not ACK; the clock is not held, but SSPxIF is still set.
- 15. The master sends a Restart condition or a Stop.
- 16. The slave is no longer addressed.





24.5.3.3 7-bit Transmission with Address Hold Enabled

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

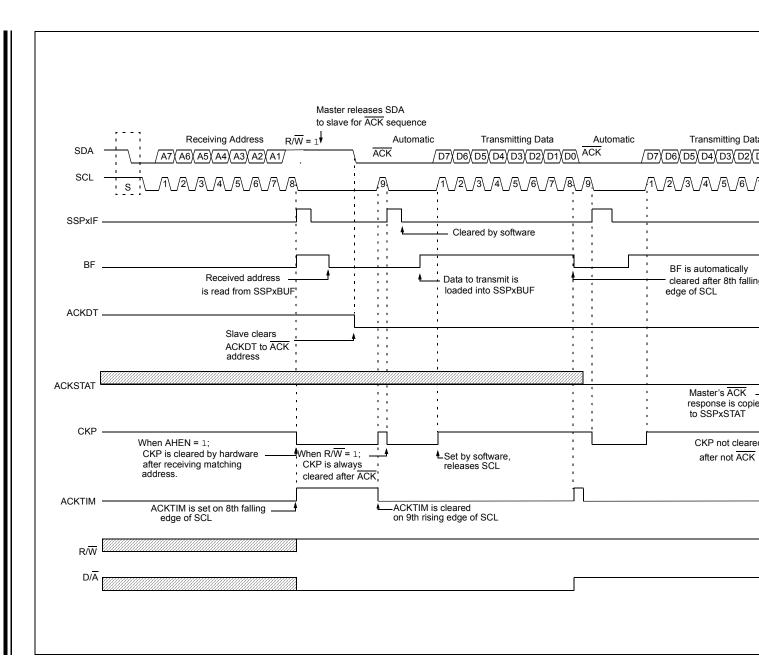
Figure 24-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

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

Note: SSPxBUF cannot be loaded until after the ACK.

- 13. Slave sets the CKP bit releasing the clock.
- 14. Master clocks out the data from the slave and sends an ACK value on the 9th SCL pulse.
- 15. Slave hardware copies the ACK value into the ACKSTAT bit of the SSPxCON2 register.
- 16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
- 17. If the master sends a not ACK the slave releases the bus allowing the master to send a Stop and end the communication.

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



24.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSP module configured as an I^2C slave in 10-bit Addressing mode.

Figure 24-20 is used as a visual reference for this description.

This is a step by step process of what must be done by slave software to accomplish I^2C communication.

- 1. Bus starts Idle.
- 2. Master sends Start condition; S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- 3. Master sends matching high address with R/\overline{W} bit clear; UA bit of the SSPxSTAT register is set.
- 4. Slave sends ACK and SSPxIF is set.
- 5. Software clears the SSPxIF bit.
- 6. Software reads received address from SSPxBUF clearing the BF flag.
- 7. Slave loads low address into SSPxADD, releasing SCL.
- 8. Master sends matching low address byte to the slave; UA bit is set.

Note: Updates to the SSPxADD register are not allowed until after the ACK sequence.

9. Slave sends ACK and SSPxIF is set.

Note: If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

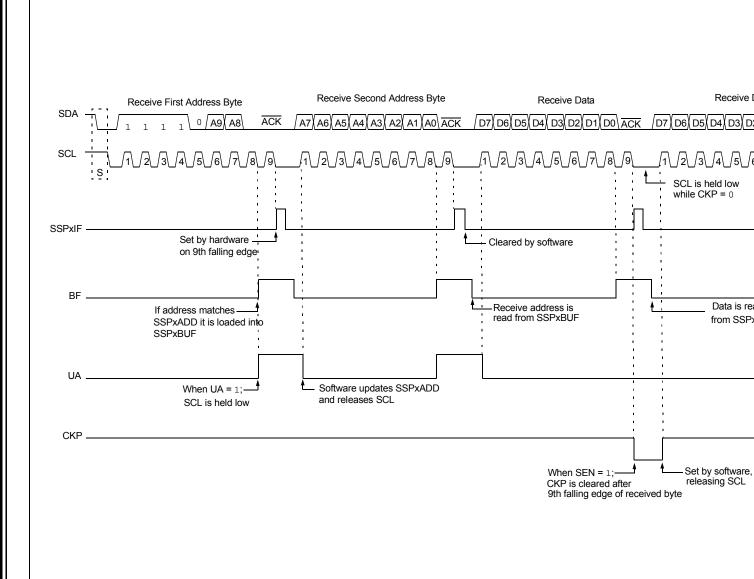
- 10. Slave clears SSPxIF.
- 11. Slave reads the received matching address from SSPxBUF clearing BF.
- 12. Slave loads high address into SSPxADD.
- 13. Master clocks a data byte to the slave and clocks out the slaves ACK on the 9th SCL pulse; SSPxIF is set.
- 14. If SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
- 15. Slave clears SSPxIF.
- 16. Slave reads the received byte from SSPxBUF clearing BF.
- 17. If SEN is set the slave sets CKP to release the SCL.
- 18. Steps 13-17 repeat for each received byte.
- 19. Master sends Stop to end the transmission.

24.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 24-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

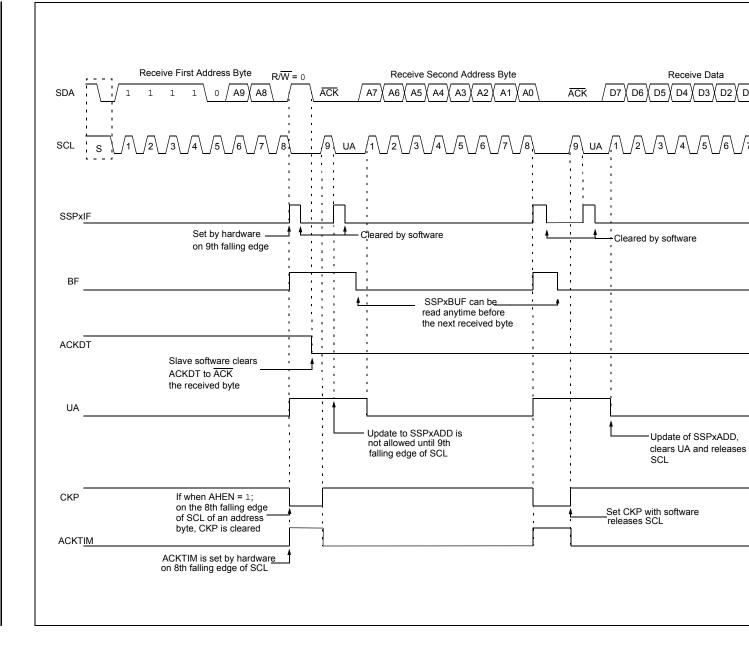
Figure 24-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.



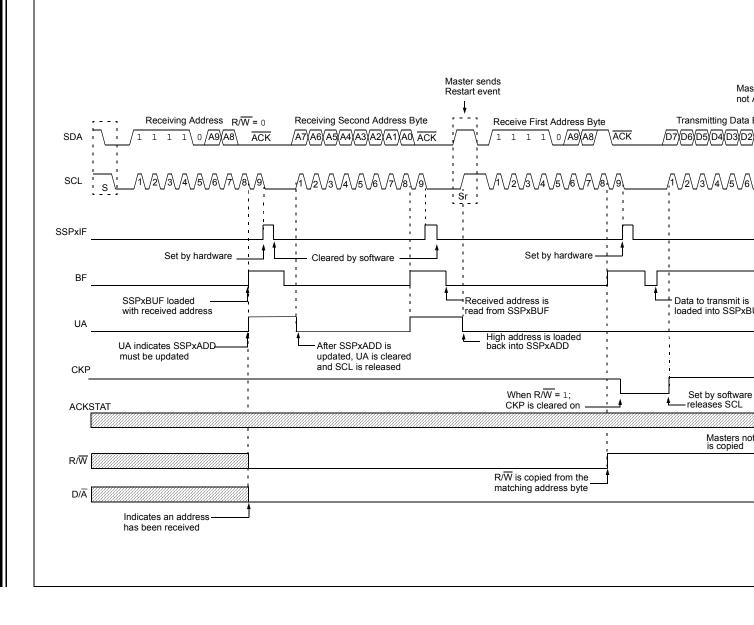












24.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

24.5.6.1 Normal Clock Stretching

Following an \overline{ACK} if the R/W bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the \overline{ACK} sequence. Once the slave is ready; CKP is set by software and communication resumes.

- Note 1:The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the 9th falling edge of SCL.
 - 2: Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the 9th falling edge of SCL. It is now always cleared for read requests.

24.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPxADD.

Note:	Previous versions of the module did not
	stretch the clock if the second address byte
	did not match.

24.5.6.3 Byte NACKing

When the AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When the DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

24.5.6.4 Clock Synchronization and the CKP Bit

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 24-23).

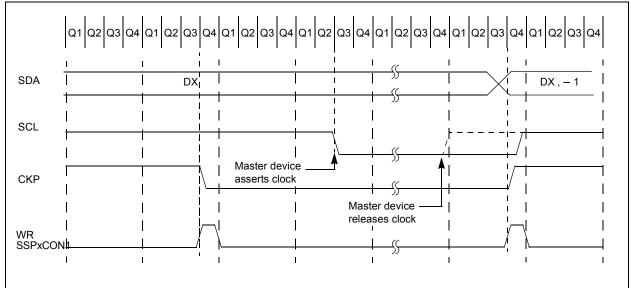


FIGURE 24-23: CLOCK SYNCHRONIZATION TIMING

24.5.7 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

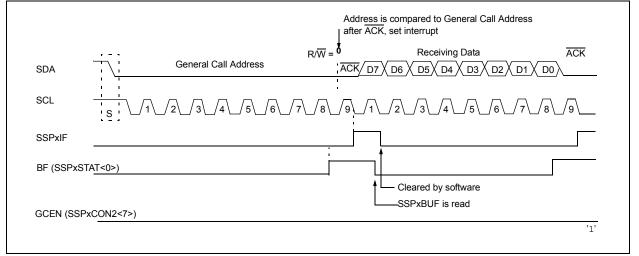
The general call address is a reserved address in the $I^{2}C$ protocol, defined as address 0x00. When the GCEN bit of the SSPxCON2 register is set, the slave module will automatically ACK the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the

R/W bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. Figure 24-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 24-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



24.5.8 SSP MASK REGISTER

An SSP Mask (SSPxMSK) register (Register 24-5) is available in I²C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPxMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

24.6 I²C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPxIF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- · Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated
 - Note 1:The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur
 - 2: Master mode suspends Start/Stop detection when sending the Start/Stop condition by means of the SEN/PEN control bits. The SSPxIF bit is set at the end of the Start/Stop generation when hardware clears the control bit.

24.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

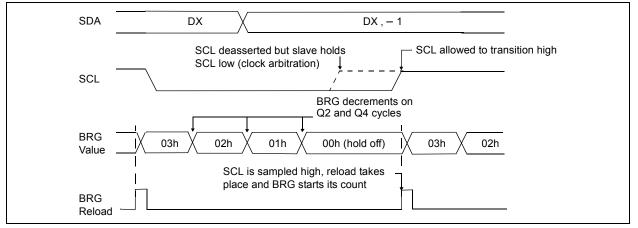
In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/\overline{W} bit. In this case, the R/\overline{W} bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See Section 24.7 "Baud Rate Generator" for more detail.

24.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 24-25).

FIGURE 24-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



24.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not idle.

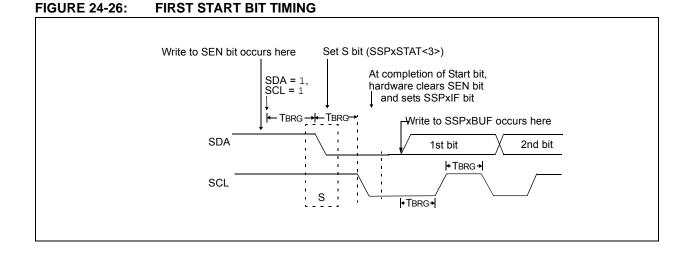
Note:	Because queuing of events is not allowed, writing to the lower five bits of SSPxCON2
	is disabled until the Start condition is complete.

24.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 24-26), the user sets the Start Enable bit, SEN bit of the SSPxCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator is reloaded with the start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared

by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

- Note 1: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCL1IF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.
 - 2: The Philips I²C specification states that a bus collision cannot occur on a Start.



24.6.5 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition (Figure 24-27) occurs when the RSEN bit of the SSPxCON2 register is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPxCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPxSTAT register will be set. The SSPxIF bit will not be set until the Baud Rate Generator has timed out.

- Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.
 - 2: A bus collision during the Repeated Start condition occurs if:
 - •SDA is sampled low when SCL goes from low-to-high.
 - •SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

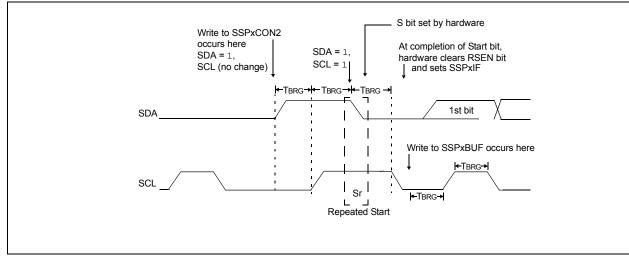


FIGURE 24-27: REPEATED START CONDITION WAVEFORM

24.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time if an address match occurred, or if data was received properly. The status of ACK is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 24-28).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT Status bit of the SSPxCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPxIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

24.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPxSTAT register is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

24.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

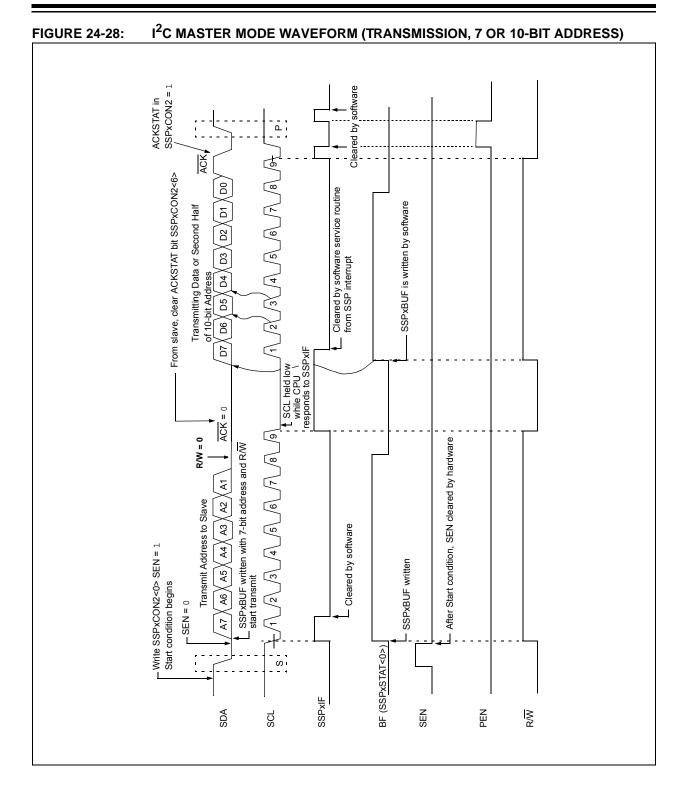
WCOL must be cleared by software before the next transmission.

24.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPxCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK} = 0$) and is set when the slave does not Acknowledge ($\overline{ACK} = 1$). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

24.6.6.4 Typical Transmit Sequence:

- 1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
- 2. SSPxIF is set by hardware on completion of the Start.
- 3. SSPxIF is cleared by software.
- 4. The MSSP module will wait the required start time before any other operation takes place.
- 5. The user loads the SSPxBUF with the slave address to transmit.
- 6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
- The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
- The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
- 9. The user loads the SSPxBUF with eight bits of data.
- 10. Data is shifted out the SDA pin until all eight bits are transmitted.
- 11. The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
- 12. Steps 8-11 are repeated for all transmitted data bytes.
- 13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPxCON2 register. Interrupt is generated once the Stop/Restart condition is complete.



24.6.7 I²C MASTER MODE RECEPTION

Master mode reception (Figure 24-29) is enabled by programming the Receive Enable bit, RCEN bit of the SSPxCON2 register.

Note:	The MSSP module must be in an Idle
	state before the RCEN bit is set or the
	RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/ low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPxBUF, the BF flag bit is set, the SSPxIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable, ACKEN bit of the SSPxCON2 register.

24.6.7.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPxBUF from SSPSR. It is cleared when the SSPxBUF register is read.

24.6.7.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when eight bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

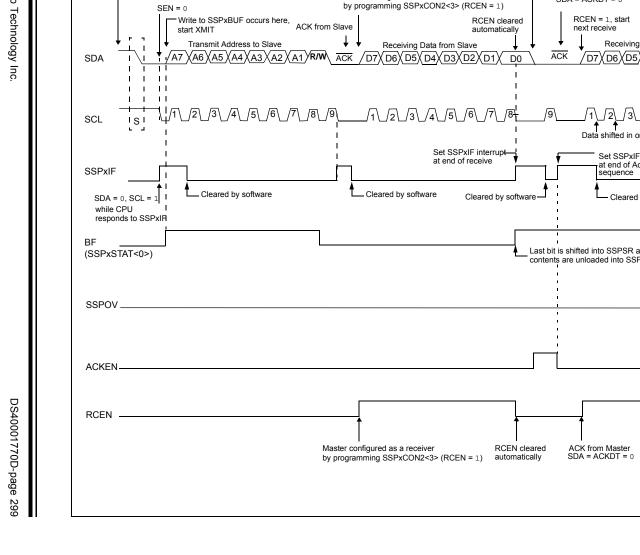
24.6.7.3 WCOL Status Flag

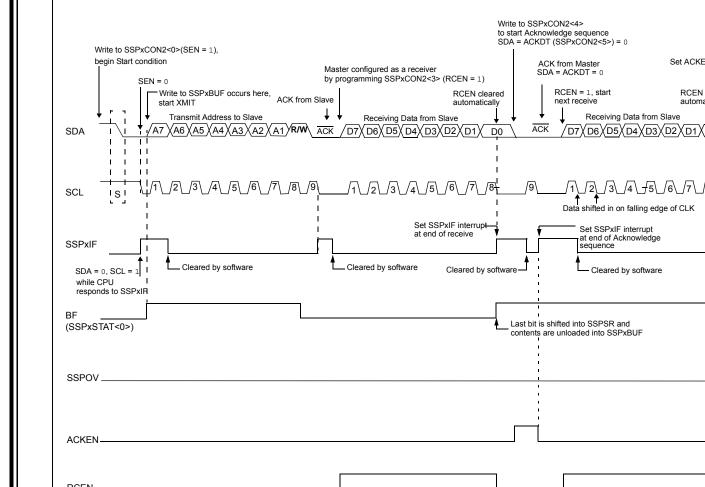
If the user writes the SSPxBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

24.6.7.4 Typical Receive Sequence:

- 1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
- 2. SSPxIF is set by hardware on completion of the Start.
- 3. SSPxIF is cleared by software.
- 4. User writes SSPxBUF with the slave address to transmit and the R/W bit set.
- 5. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
- The MSSP module shifts in the ACK bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
- 7. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
- 8. User sets the RCEN bit of the SSPxCON2 register and the master clocks in a byte from the slave.
- 9. After the eighth falling edge of SCL, SSPxIF and BF are set.
- 10. Master clears SSPxIF and reads the received byte from SSPxBUF, clears BF.
- 11. Master sets ACK value sent to slave in ACKDT bit of the SSPxCON2 register and initiates the ACK by setting the ACKEN bit.
- 12. Master's ACK is clocked out to the slave and SSPxIF is set.
- 13. User clears SSPxIF.
- 14. Steps 8-13 are repeated for each received byte from the slave.
- 15. Master sends a not ACK or Stop to end communication.







Set ACKEN, start Acknowl

D0

RCEN cleared automatically

/7\ 8

6\

SDA = ACKDT

ACK

ACK is

/9\

Set SSPxIF of receive

SSPOV is set b SSPxBUF is sti

RCEN cleared automatically

24.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 24-30).

24.6.8.1 WCOL Status Flag

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

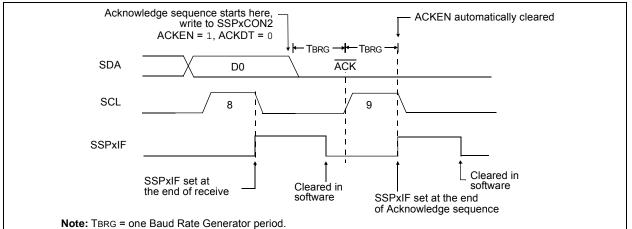
24.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 24-31).

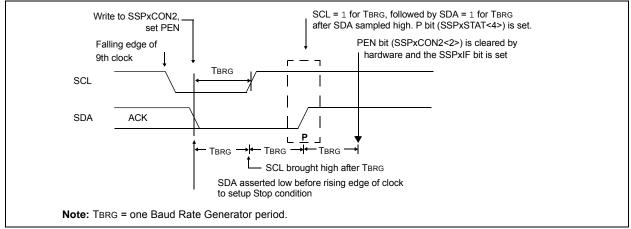
24.6.9.1 WCOL Status Flag

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

FIGURE 24-30: ACKNOWLEDGE SEQUENCE WAVEFORM







24.6.10 SLEEP OPERATION

While in Sleep mode, the I²C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

24.6.11 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

24.6.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit of the SSPxSTAT register is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCL1IF bit.

The states where arbitration can be lost are:

- · Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

24.6.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCL1IF and reset the I²C port to its Idle state (Figure 24-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

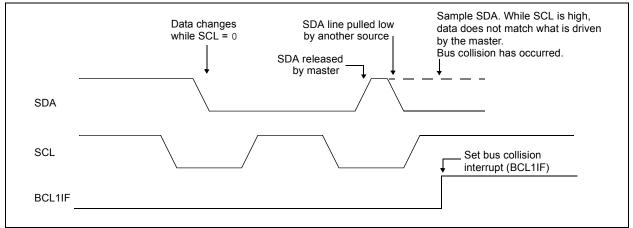
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 24-32: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



© 2014-2017 Microchip Technology Inc.

24.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 24-33).
- b) SCL is sampled low before SDA is asserted low (Figure 24-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

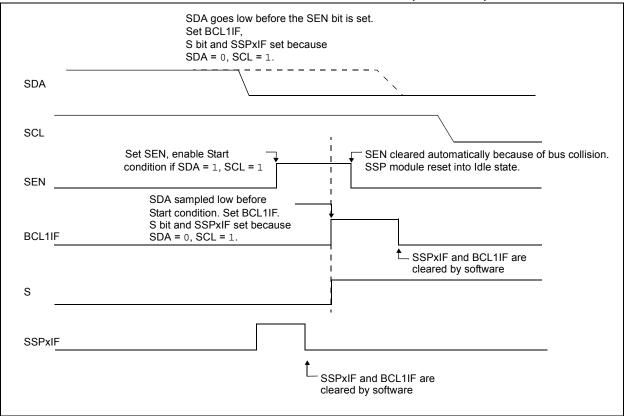
- · the Start condition is aborted,
- the BCL1IF flag is set and
- the MSSP module is reset to its Idle state (Figure 24-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

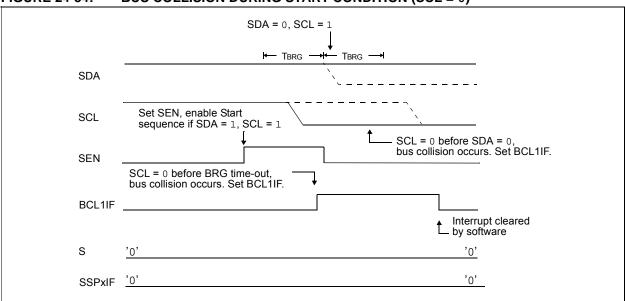
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 24-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note: The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

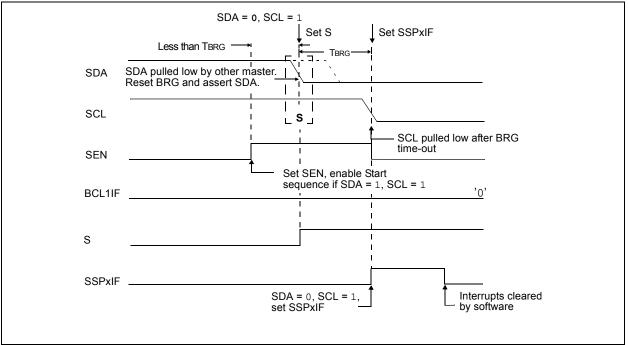












24.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

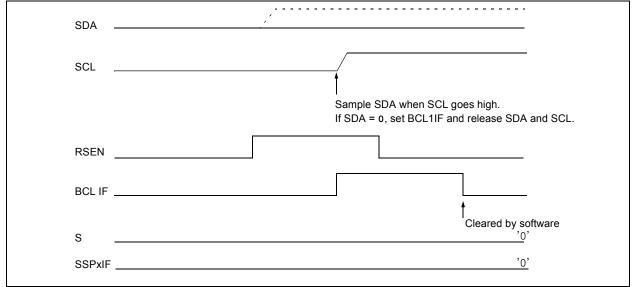
- a) A low level is sampled on SDA when SCL goes from low level to high level (Case 1).
- b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 24-36). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

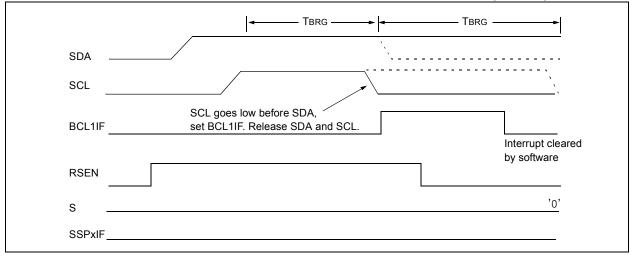
If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 24-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 24-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)







24.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

- a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).
- b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 24-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 24-39).

FIGURE 24-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)

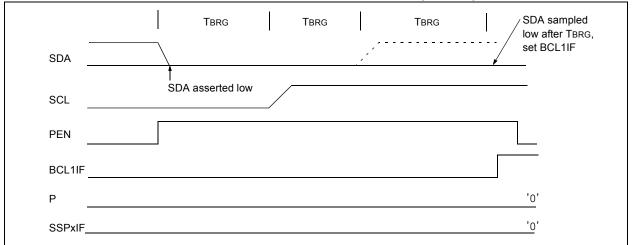
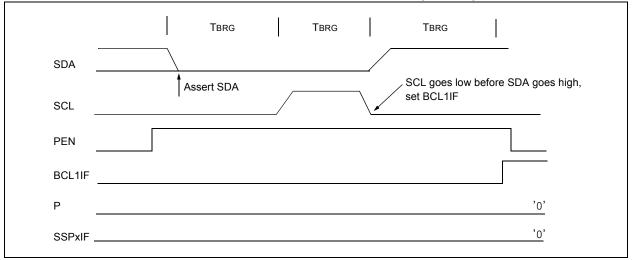


FIGURE 24-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
ANSELA	—	_	-	ANSA4	_	ANSA2	ANSA1	ANSA0	152
ANSELB ⁽¹⁾	—	_	ANSB5	ANSB4	_	—	_	—	159
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾			ANSC3	ANSC2	ANSC1	ANSC0	166
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIE2	OSFIE	C2IE	C1IE	_	BCL1IE	TMR6IE	TMR4IE	CCP2IE	99
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
PIR2	OSFIF	C2IF	C1IF	_	BCL1IF	TMR6IF	TMR4IF	CCP2IF	104
RxyPPS	_	_	_	RxyPPS<4:0>					
SSPCLKPPS	—	_	_		SS	PCLKPPS<4	:0>		174, 172
SSPDATPPS	_	_	_		SS	PDATPPS<4	:0>		174, 172
SSPSSPPS	_	_	_		S	SPSSPPS<4:)>		174, 172
SSP1ADD				ADD	<7:0>				312
SSP1BUF	Synchronous	s Serial Port F	Receive Buffe	r/Transmit Re	egister				264*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP		SSPN	<3:0>		309
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	310
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	311
SSP1MSK				MSK	<7:0>				312
SSP1STAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	308
TRISA	—	—	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	151
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165

	TABLE 24-3:	SUMMARY OF REGISTERS ASSOCIATED WITH I ² C OPERATION
--	-------------	---

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the MSSP module in I²C mode. * Page provides register information.

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

24.7 BAUD RATE GENERATOR

The MSSP module has a Baud Rate Generator available for clock generation in both I²C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 24-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

An internal signal "Reload" in Figure 24-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the

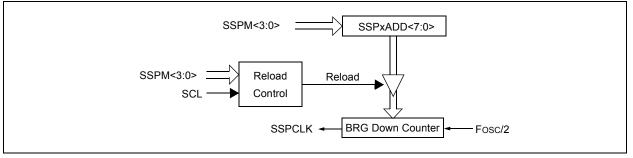
module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 24-4demonstratesclockratesbasedoninstructioncyclesandtheBRGvalueloadedintoSSPxADD.

EQUATION 24-1:

$$FCLOCK = \frac{FOSC}{(SSPxADD + 1)(4)}$$

FIGURE 24-40: BAUD RATE GENERATOR BLOCK DIAGRAM



Note: Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for I²C. This is an implementation limitation.

TABLE 24-4: MSSP CLOCK RATE W/BRG

Fosc	Fcy	BRG Value	FCLOCK (2 Rollovers of BRG)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

Note: Refer to the I/O port electrical specifications in Figure 35-7 and Table 35-10 to ensure the system is designed to support the I/O timing requirements.

24.8 Register Definitions: MSSP Control

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
SMP	CKE	D/A	Р	S	R/W	UA	BF
bit 7							bit C
Legend:	••						
R = Readable b		W = Writable bi		•	nted bit, read as '0		
u = Bit is unchar	nged	x = Bit is unkno		-n/n = Value at I	POR and BOR/Val	ue at all other Res	ets
'1' = Bit is set		'0' = Bit is clear	ed				
bit 7	<u>SPI Master mo</u> 1 = Input data	a Input Sample bit <u>ode:</u> sampled at end of sampled at middle		ne			
	In I ² C Master of	l <u>e:</u> cleared when SPI i <u>or Slave mode:</u>					
		control disabled fo control enabled for			and 1 MHz)		
bit 6	CKE: SPI Cloc In SPI Master of 1 = Transmit o 0 = Transmit o In l^2C^{TM} mode 1 = Enable inp	k Edge Select bit (or Slave mode: ccurs on transition ccurs on transition	SPI mode only) from active to Idl from Idle to activ esholds are comp	e clock state e clock state	specification		
bit 5	 D/A: Data/Address bit (I²C mode only) 1 = Indicates that the last byte received or transmitted was data 0 = Indicates that the last byte received or transmitted was address 						
bit 4	 P: Stop bit (I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.) 1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset) 0 = Stop bit was not detected last 						
bit 3	 S: Start bit (I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.) 1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset) 0 = Start bit was not detected last 						
bit 2	This bit holds ti next Start bit, S In l^2C Slave m 1 = Read 0 = Write In l^2C Master r 1 = Transmit 0 = Transmit		tion following the				ress match to the
bit 1	UA: Update Address bit (10-bit I ² C mode only) 1 = Indicates that the user needs to update the address in the SSP1ADD register 0 = Address does not need to be updated						
bit 0	0 = Receive no <u>Transmit (I²C r</u> 1 = Data transi	and I ² C modes): omplete, SSP1BUF ot complete, SSP1I	BUF is empty es not include the				

REGISTER 24-2: SSP1CON1: SSP CONTROL REGISTER 1

R/C/HS-0/0	R/C/HS-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
WCOL	SSPOV ⁽¹⁾	SSPEN	CKP		SSPN	/<3:0>	
pit 7							bit
Legend:							
R = Readable bit		W = Writable bit			ted bit, read as '0'		
u = Bit is unchange	d	x = Bit is unknow			OR and BOR/Value		
1' = Bit is set		'0' = Bit is cleared	d	HS = Bit is set by	/ hardware	C = User cleared	
bit 7	0 = No collision Slave mode:	ne SSP1BUF regist			itions were not valio word (must be clear	l for a transmission ed in software)	to be started
bit 6	SSPOV: Receive In SPI mode: 1 = A new byte i Overflow ca setting overf SSP1BUF rr 0 = No overflow In I ² C mode: 1 = A byte is re	Overflow Indicator is received while the n only occur in Slav flow. In Master mode egister (must be cle v ceived while the S eared in software).	e SSP1BUF registe re mode. In Slave r e, the overflow bit is ared in software). SP1BUF register	node, the user must s not set since each	t read the SSP1BUF, new reception (and tr	e of overflow, the data even if only transmi ransmission) is initiat DV is a "don't care"	tting data, to avoid ad by writing to the
bit 5	In both modes, w In SPI mode: 1 = Enables ser 0 = Disables ser $\ln l^2 C$ mode: 1 = Enables the	ial port and configuerial port and configue	e pins must be pro res SCK, SDO, SE gures these pins a figures the SDA an	s I/O port pins d SCL pins as the so	s input or output arce of the serial port purce of the serial po		
bit 4	0 = Idle state for $1000000000000000000000000000000000000$	clock is a high leve clock is a low level <u>e:</u> rol ow (clock stretch). <u>de:</u>		ata setup time.)			
bit 3-0	1111 = I^2C Slave 1110 = I^2C Slave 1101 = Reserved 1101 = Reserved 1011 = I^2C firmw 1010 = SPI Mast 1001 = Reserved 1000 = I^2C Slave 0111 = I^2C Slave 0110 = SPI Slave 0101 = SPI Mast 0001 = SPI Mast	e mode, 7-bit addre are controlled Mas er mode, clock = F er mode, clock = F e mode, clock addr e mode, 10-bit addre e mode, 7-bit addre	ess with Start and ss with Start and osc/(4 * (SSP1AE osc / (4 * (SSP1AE osc / (4 * (SSP1AE osc / (4 * (SSP1A osc / (4 * (SSP1A css Sk pin, <u>SS</u> pin cor 2 match/2 osc/64 osc/16	I Stop bit interrupts Stop bit interrupts e lle) DD+1)) ⁽⁵⁾ DD+1)) ⁽⁴⁾ trol disabled, SS ci		in	
2: Wher to sel 3: Wher	n enabled, these pi ect the pins. n enabled, the SDA	ins must be proper	ly configured as ir	put or output. Use is inputs. Use SSP	SSPSSPPS, SSPC	by writing to the SS LKPPS, SSPDATPI PPS, and RxyPPS to	PS, and RxyPPS

4: SSP1ADD values of 0, 1 or 2 are not supported for I²C mode.
 5: SSP1ADD value of '0' is not supported. Use SSPM = 0000 instead.

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7				•			bit
Legend:							
R = Readable	hit	W = Writable	bit	II = Unimpler	mented bit, read	1 as 'N'	
u = Bit is unch		x = Bit is unk		-	at POR and BO		ther Resets
'1' = Bit is set	angeo	'0' = Bit is cle			d by hardware		
						0 - 0301 301	
bit 7	1 = Enable in			• •	or 00h) is receiv	red in the SSPS	SR
bit 6	1 = Acknowle	cknowledge S edge was not r edge was rece		mode only)			
bit 5	ACKDT: Ack	nowledge Data	a bit (in I ² C mo	de only)			
	In Receive m Value transm 1 = Not Ackn 0 = Acknowle	itted when the owledge	user initiates a	an Acknowledg	ge sequence at	the end of a re	ceive
bit 4		•	uence Enable	bit (in I ² C Mas	ter mode only)		
	Automat		by hardware.	SDA and S	CL pins, and	transmit ACF	KDT data b
bit 3	RCEN: Rece	ive Enable bit Receive mode	(in I ² C Master	mode only)			
bit 2	SCKMSSP R	elease Contro	_		y) atically cleared	by bordworo	
	1 = 1111111111111111111111111111111111		III SDA aliu SC	L pins. Autom		by hardware.	
bit 1	RSEN: Repe	ated Start Cor			er mode only) ins. Automatica	Illy cleared by h	nardware.
bit 0	In Master mo	<u>de:</u> tart condition c	e/Stretch Enab		atically cleared	by hardware.	
				ave transmit a	nd slave receive	e (stretch enabl	ed)
Note 1: For	bits ACKEN F	RCEN PEN R	SEN. SEN: If t	he l ² C module	is not in the Idl	e mode this bi	t may not be

REGISTER 24-3: SSP1CON2: SSP CONTROL REGISTER 2⁽¹⁾

Note 1: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I²C module is not in the Idle mode, this bit may not be set (no spooling) and the SSP1BUF may not be written (or writes to the SSP1BUF are disabled).

R-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	
ACKTIM ⁽³⁾	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	
bit 7				1		1	bit C	
Legend:								
R = Readable		W = Writable		•	mented bit, read			
u = Bit is unch	•	x = Bit is unk		-n/n = Value	Value at POR and BOR/Value at all other Resets			
'1' = Bit is set		'0' = Bit is cle	eared					
bit 7	ACKTIM: Ac	knowledge Tim	e Status bit (l	2 C mode only)	(3)			
					e, set on eighth	falling edge of	SCL clock	
	0 = Not an A	cknowledge se	quence, clear	ed on 9 th rising	edge of SCL c	lock		
bit 6		Condition Interru	•	•	de only)			
		nterrupt on detention interrupt						
hit E	•	Condition Interrupt						
bit 5		nterrupt on dete	•	•	• /			
		ection interrupt						
bit 4	BOEN: Buffe	er Overwrite En	able bit					
	In SPI Slave							
					oyte is shifted in			
		ew byte is rece 1CON1 registe			STAT register a	aiready set, SS	POV bit of the	
		r mode and SP			apaaloa			
		s ignored.						
	<u>In I²C Slave I</u>		tod and $\overline{\Lambda CK}$ i	s concrated fo	or a received ad	ldross/data byt	a ignoring the	
		of the SSPOV			a received ad		e, ignoring the	
		1BUF is only u			ır			
bit 3	SDAHT: SDA	A Hold Time Se	election bit (I ² C	c mode only)				
		of 300 ns hold						
hit 2		of 100 ns hold		-	g edge of SCL C Slave mode o			
bit 2						3,	aiah atata tha	
		f the PIR2 regis			en the module i	s outputting a l	nigh state, the	
		lave bus collisi		0				
		s collision inter		bled				
bit 1	AHEN: Addre	ess Hold Enabl	le bit (I ² C Slav	e mode only)				
					ching received	address byte; (CKP bit of the	
		DN1 register wi holding is disal		nd the SCL wil	ll be held low.			
bit 0		Hold Enable b		ode only)				
				• ·	ed data byte; sl	ave hardware o	lears the CKF	
		SSP1CON1 r						
	0 = Data holo	ding is disabled	l					
Note 1: For	r daisy-chained	SPI operation;	allows the use	er to ignore all	but the last rece	eived byte. SSP	OV is still set	
wh	en a new byte i				ues to write the			
	P1BUF.				1919 - J. J. J. M. 19			
2: Thi	is bit has no eff	ect in Slave mo	odes that Start	and Stop cond	dition detection	is explicitly liste	ed as enabled.	

REGISTER 24-4: SSP1CON3: SSP CONTROL REGISTER 3

3: The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

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	
			MSK	<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 a	at POR and BO	R/Value at all c	other Resets	
'1' = Bit is set '0' = Bit is cleared			ared					
				-n/n = Value a	at POR and BO	R/Value at all c	other Res	
bit 7-1	MSK<7:1>: 1 = The rec	eived address b			D <n> to detect</n>		atch	

REGISTER 24-5: SSP1MSK: SSP MASK REGISTER

	1 = The received address bit n is compared to SSP1ADD <n> to detect I²C address match</n>
	0 = The received address bit n is not used to detect I ² C address match
bit 0	MSK<0>: Mask bit for I ² C Slave mode, 10-bit Address
	I ² C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
	1 = The received address bit 0 is compared to SSP1ADD<0> to detect I ² C address match
	0 = The received address bit 0 is not used to detect I ² C address match
	I ² C Slave mode, 7-bit address, the bit is ignored

REGISTER 24-6: SSP1ADD: MSSP ADDRESS AND BAUD RATE REGISTER (I²C MODE)

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
ADD<7:0>							
bit 7							bit 0
Legend:							
R = Readable b	oit	W = Writable bi	t	U = Unimpler	nented bit, read	1 as '0'	

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	

Master mode:

bit 7-0	ADD<7:0>: Baud Rate Clock Divider bits
	SCL pin clock period = ((ADD<7:0> + 1) *4)/Fosc

10-Bit Slave mode – Most Significant Address Byte:

- bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a "don't care". Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.
- bit 2-1 ADD<2:1>: Two Most Significant bits of 10-bit address
- bit 0 Not used: Unused in this mode. Bit state is a "don't care".

<u>10-Bit Slave mode – Least Significant Address Byte:</u>

bit 7-0 ADD<7:0>: Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

bit 0 Not used: Unused in this mode. Bit state is a "don't care".

25.0 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 Full-Duplex is useful system. mode 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 polarity in synchronous modes
- · Sleep operation

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 25-1 and Figure 25-2.

The EUSART transmit output (TX_out) is available to the TX/CK pin and internally to the following peripherals:

Configurable Logic Cell (CLC)

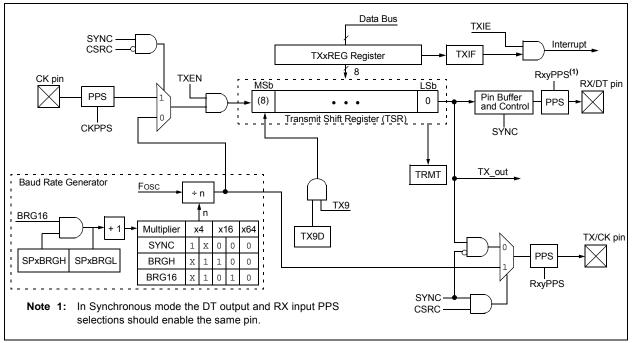
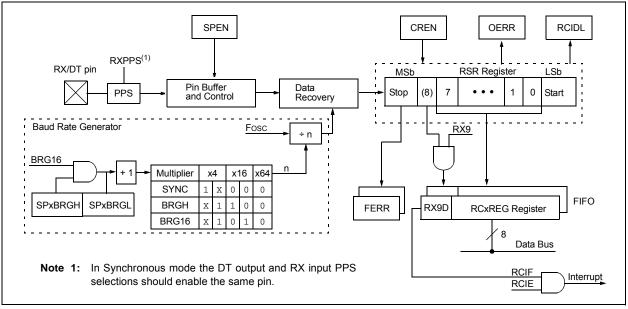


FIGURE 25-1: EUSART TRANSMIT BLOCK DIAGRAM





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

- Transmit Status and Control (TXxSTA)
- · Receive Status and Control (RCxSTA)
- Baud Rate Control (BAUDxCON)

These registers are detailed in Register 25-1, Register 25-2 and Register 25-3, respectively.

The RX and CK input pins are selected with the RXPPS and CKPPS registers, respectively. TX, CK, and DT output pins are selected with each pin's RxyPPS register. Since the RX input is coupled with the DT output in Synchronous mode, it is the user's responsibility to select the same pin for both of these functions when operating in Synchronous mode. The EUSART control logic will control the data direction drivers automatically.

25.1 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 25-5 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.

25.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 25-1. 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 TXxREG register.

25.1.1.1 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 TXxSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXxSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCxSTA register enables the EUSART and automatically configures 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.

25.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXxREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXxREG 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 TXxREG until the Stop bit of the previous character has been transmitted. The pending character in the TXxREG 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 TXxREG.

25.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDxCON 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 in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See Section 25.5.1.2 "Clock Polarity".

25.1.1.4 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 TXxREG. 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 TXxREG. The TXIF flag bit is not cleared immediately upon writing TXxREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXxREG 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 TXxREG is empty, regardless of the state of 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 TXxREG.

25.1.1.5 TSR Status

The TRMT bit of the TXxSTA 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 TXxREG. 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 has 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.

25.1.1.6 Transmitting 9-Bit Characters

The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXxSTA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TXxSTA 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 TXxREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXxREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 25.1.2.7 "Address Detection**" for more information on the Address mode.

25.1.1.7 Asynchronous Transmission Set-up:

- Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 25.4 "EUSART Baud Rate Generator (BRG)").
- 2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
- 3. 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.
- 4. Set SCKP bit if inverted transmit is desired.
- 5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
- If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
- 7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
- 8. Load 8-bit data into the TXxREG register. This will start the transmission.

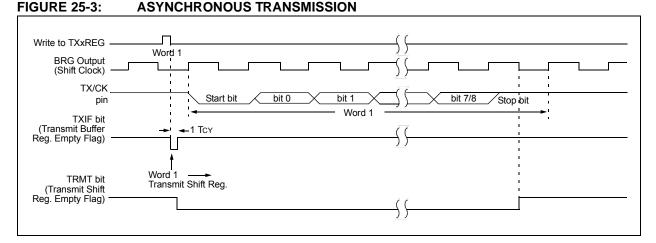
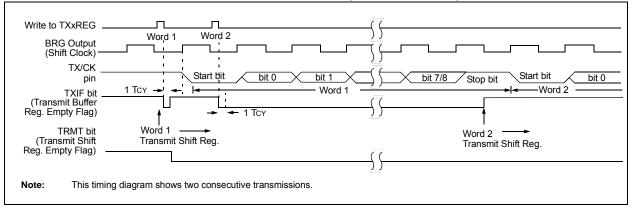


FIGURE 25-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—		ANSA4	—	ANSA2	ANSA1	ANSA0	152
ANSELB ⁽¹⁾	_	_	ANSB5	ANSB4	_	_	_		159
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	_	—	– ANSC3		ANSC1	ANSC0	166
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	_	WUE	ABDEN	325
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	324
RxyPPS	_	_	_		172				
SP1BRGL				BRG<	:7:0>				326*
SP1BRGH				BRG<	15:8>				326*
TRISA	_	_	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	149
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
TX1REG	EUSART Tra	nsmit Data R	egister	•	•	•	•	•	315*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323

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

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

25.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 25-2. 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 RCxREG register.

25.1.2.1 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.

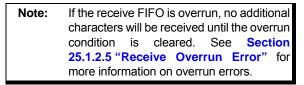
Setting the CREN bit of the RCxSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXxSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCxSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to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

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

25.1.2.2 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 25.1.2.4 "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 RCxREG register.



25.1.2.3 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 all of 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.

25.1.2.4 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 RCxSTA 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 RCxREG.

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 RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA 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 RCxREG will not clear the FERR bit.

25.1.2.5 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 RCxSTA 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 RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

25.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set, the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA 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 RCxREG.

25.1.2.7 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 RCxSTA 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.

25.1.2.8 Asynchronous Reception Set-up

- Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 25.4 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- 3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- 4. If interrupts are desired, set the RCIE bit of the PIE1 register and 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 RCxSTA 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 RCxREG register.
- 10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

25.1.2.9 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 SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 25.4 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- 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 bit of the PIE1 register and 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 RCxSTA 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 RCxREG 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.

Start Start Star RX/DT pin bit 0 bit ΄bit Ο` bit 1 /bit 7/8/ Stop bit (bit 7/8/ Stop bit , bit 7/8/ Stop bit bit Rcv Shift Word 2 Word 1 RCxREG RCxREG RCIDL Read Rcv Buffer Reg. RCxREG RCIF (Interrupt Flag) OERR bit CREN Note: This timing diagram shows three words appearing on the RX input. The RCxREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

FIGURE 25-5:

ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page	
ANSELA	—	—	_	ANSA4	—	ANSA2	ANSA1	ANSA0	152	
ANSELB ⁽¹⁾	_	_	ANSB5	ANSB4	_	_	-	_	159	
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	_	—	ANSC3	ANSC2	ANSC1	ANSC0	166	
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	325	
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97	
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	97	
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103	
RC1REG	EUSART Receive Data Register									
RC1STA	SPEN	RX9	SREN	CREN ADDEN FERR OERR RX9D			324			
RxyPPS	—	_			ŀ	RxyPPS<4:0	>		172	
SP1BRGL				BRG<	:7:0>				326	
SP1BRGH				BRG<	15:8>				326	
TRISA	—	—	TRISA5	TRISA4	_(2)	TRISA2	TRISA1	TRISA0	151	
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158	
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165	
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323	

TABLE 25-2: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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

* Page provides register information.

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

25.2 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. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 5.2.2.3 "Internal Oscillator Frequency Adjustment"** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see Section 25.4.1 "Auto-Baud Detect"). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

25.3 Register Definitions: EUSART Control

REGISTER 25-1: TX1STA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0				
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D				
bit 7							bit (
Legend:											
R = Readab	le bit	W = Writable	bit	U = Unimplemented bit, read as '0'							
u = Bit is un	changed	x = Bit is unk	nown	-n/n = Value a	at POR and BO	R/Value at all o	other Resets				
'1' = Bit is se	et	'0' = Bit is cle	ared								
bit 7	Asynchronou Don't care		t bit								
		<u>s mode</u> : mode (clock ge 1ode (clock fron)						
bit 6	1 = Selects	ansmit Enable 9-bit transmiss 8-bit transmiss	ion								
bit 5	TXEN: Trans 1 = Transmi 0 = Transmi		1)								
bit 4	SYNC: EUS 1 = Synchro 0 = Asynchr		ect bit								
bit 3	Asynchronou 1 = Send Sy	nc Break on ne eak transmissic	ext transmissio	n (cleared by l	nardware upon o	completion)					
bit 2	BRGH: High Asynchronou 1 = High spe 0 = Low spe Synchronous Unused in th	eed eed <u>s mode:</u>	ect bit								
bit 1		smit Shift Regis pty	ter Status bit								
bit 0	TX9D: Ninth	bit of Transmit ess/data bit or a									
Note 1: S	REN/CREN ove	rrides TXEN in	Sync mode.								

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0					
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D					
bit 7							bit					
Legend:												
R = Readable	e bit	W = Writable	bit	U = Unimpler	nented bit, read	l as '0'						
u = Bit is uncl	nanged	x = Bit is unk	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									
	SPEN: Serial Port Enable bit											
bit 7			It									
		L = Serial port enabled D = Serial port disabled (held in Reset)										
bit 6	RX9: 9-Bit Receive Enable bit											
		9-bit reception	on									
	0 = Selects 8-bit reception											
bit 5	SREN: Singl	e Receive Ena	ble bit									
	Asynchronous mode:											
	Don't care											
		mode – Maste	<u>er</u> :									
		single receive										
		s single receive ared after rece		ete								
		Synchronous mode – Slave										
	Don't care											
bit 4	CREN: Continuous Receive Enable bit											
	Asynchronous mode:											
	1 = Enables receiver											
	0 = Disables receiver											
	Synchronous mode:											
	 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN) 0 = Disables continuous receive 											
bit 3		lress Detect Er										
	Asynchronou	is mode 9-bit (F	RX9 = 1):									
	1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set											
		0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit										
		<u>is mode 8-bit (f</u>	<u> RX9 = 0)</u> :									
	Don't care											
bit 2	FERR: Fram	•										
	1 = Framing 0 = No frami		ipdated by rea	aling RUXREG	register and re	ceive next valid	i byte)					
bit 1	OERR: Over	-										
-		error (can be o	leared by clea	aring bit CREN)							
	0 = No overi		,	<u> </u>								
bit 0	RX9D: Ninth	bit of Received	l Data									

REGISTER 25-2: RC1STA: 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			•			·	bit (
Legend:													
R = Readable	e bit	W = Writable	e bit	U = Unimplem	nented bit, rea	d as '0'							
u = Bit is uncl	hanged	x = Bit is unl	known	-n/n = Value a	t POR and BO	OR/Value at all c	ther Resets						
'1' = Bit is set		'0' = Bit is cl	eared										
L:4 7		ute Devel Dete	et Overflevy bit										
bit 7	ABDOVF: Auto-Baud Detect Overflow bit Asynchronous mode:												
	<u>Asynchronous mode</u> : 1 = Auto-baud timer overflowed												
	0 = Auto-baud timer did not overflow												
	<u>Synchronou</u> Don't care	<u>s mode</u> :											
bit 6	RCIDL: Rec	eive Idle Flag b	pit										
	<u>Asynchrono</u>												
	 1 = Receiver is Idle 0 = Start bit has been received and the receiver is receiving 												
	0 = Start bit Synchronou		ved and the re	ceiver is receivi	ng								
	Don't care	<u>o modo</u> .											
bit 5	Unimpleme	Unimplemented: Read as '0'											
bit 4	SCKP: Synchronous Clock Polarity Select bit												
	<u>Asynchrono</u>	Asynchronous mode:											
	 1 = Transmit inverted data to the TX/CK pin 0 = Transmit non-inverted data to the TX/CK pin 												
	Synchronous mode:												
	 1 = Data is clocked on rising edge of the clock 0 = Data is clocked on falling edge of the clock 												
bit 3	BRG16: 16-	-bit Baud Rate Generator bit											
		aud Rate Gene											
bit 2	Unimpleme	nted: Read as	'0'										
bit 1	WUE: Wake-up Enable bit												
	Asynchronous mode:												
	will autor	 1 = Receiver is waiting for a falling edge. No character will be received, byte RCIF will be set. WUE will automatically clear after RCIF is set. 											
		r is operating n	ormally										
	<u>Synchronou</u> Don't care	<u>s mode</u> :											
bit 0		to-Baud Detec	Enable bit										
	ABDEN. Au Asynchrono												
			le is enabled (c	lears when aut	o-baud is com	plete)							
		aud Detect mod											
	<u>Synchronou</u>												
	Don't care												

REGISTER 25-3: BAUD1CON: BAUD RATE CONTROL REGISTER

25.4 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 BAUDxCON register selects 16-bit mode.

The SPxBRGH, SPxBRGL 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 TXxSTA register and the BRG16 bit of the BAUDxCON register. In Synchronous mode, the BRGH bit is ignored.

Table 25-3 contains the formulas for determining the baud rate. Example 25-1 provides a sample calculation for determining the 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 25-5. 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 SPxBRGH, SPxBRGL 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 25-1: CALCULATING BAUD RATE ERROR

For a device with Fosc of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate = $\frac{Fosc}{64([SPBRGH:SPBRGL] + 1)}$

Solving for SPxBRGH:SPxBRGL:

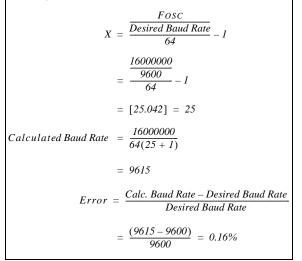


TABLE 25-3: BAUD RATE FORMULAS

C	Configuration Bits			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/Synchronous	1			

Legend: x = Don't care, n = value of SPxBRGH, SPxBRGL register pair.

TABLE 25-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page	
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16		WUE	ABDEN	325	
RC1STA	SPEN RX9 SREN CREN ADDEN FERR OERR RX9D									
SP1BRGL	BRG<7:0>									
SP1BRGH		BRG<15:8>								
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323	

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

* Page provides register information.

					SYNC	C = 0, BRG	l = 0, BRG	616 = 0					
BAUD	Fosc	; = 20.00	0 MHz	Fosc	; = 18.43	2 MHz	Fosc = 16.000 MHz			Fosc = 11.0592 MHz			
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	
300	_	_	_	_	_	_	_		_	_			
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 25-5:BAUD RATES FOR ASYNCHRONOUS MODES

			_		SYNC	C = 0, BRGH	I = 0, BRO	616 = 0				
BAUD	Fos	c = 8.000) MHz	Fos	c = 4.000) MHz	Fosc	= 3.686	4 MHz	Fos	c = 1.000) MHz
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	_		_	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, BRO	616 = 0				
BAUD	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
RATE			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

					SYNC	C = 0, BRGH	l = 1, BRC	G16 = 0				
BAUD	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	_	_	-	_	_	_	_	_	_	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 25-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONT	INUED)
---	--------

					SYNC	C = 0, BRGH	l = 0, BRG	616 = 1				
BAUD	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	-0.01	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, BRC	616 = 1				
BAUD	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	_	_
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	_	_	_
57.6k	55556	-3.55	8	—	—	_	57.60k	0.00	3	—	_	_
115.2k	—	_	—		_	—	115.2k	0.00	1	—	_	—

				SYNC = 0	, BRGH	= 1, BRG16	= 1 or SY	'NC = 1,	BRG16 = 1			
BAUD	Foso	: = 20.00	0 MHz	Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	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 25-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

				SYNC = 0	, BRGH	= 1, BRG16	= 1 or SY	'NC = 1,	BRG16 = 1			
BAUD	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	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	—	_	—

25.4.1 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 BAUDxCON register starts the auto-baud calibration sequence. 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 SPxBRG begins counting up using the BRG counter clock as shown in Figure 25-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPxBRGH, SPxBRGL register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCxREG needs to be read to clear the RCIF interrupt. RCxREG content should be discarded. When calibrating for modes that do not use the SPxBRGH register the user can verify that the SPxBRGL register did not overflow by checking for 00h in the SPxBRGH register.

	TABLE 25-6:	BRG COUNTER CLOCK RA	ATES
--	-------------	----------------------	------

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, SPxBRGL and SPxBRGH registers are both used as a 16-bit counter, independent of the BRG16 setting.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 25-6. During ABD, both the SPxBRGH and SPxBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPxBRGH and SPxBRGL 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</u> 25.4.3 "Auto-Wake-up on <u>Break</u>").
 - 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 one. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPxBRGH:SPxBRGL register pair.

BRG Value	XXXXh	0000h		001Ch
RX pin		Start		- Edge #5 Stop bit
BRG Clock		huuuuuu		; Efinanononon
ABDEN bit	Set by User —	י י 	(``, ``,	Auto Cleared
ABDEN DI		J I	; 	1
RCIDL	. <u> </u>	ı ı — — — — — — — — — — — — — — — — — —		1 1
RCIF bit		ı <u>İ</u>		<u>;</u>
(Interrupt)	I	1		: /
Read	1	1 1		: >
RCxREG		! 		· ·
SPxBRGL		I I	XXh	1Ch
SPxBRGH		•	XXh X	00h

FIGURE 25-6: AUTOMATIC BAUD RATE CALIBRATION

25.4.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDxCON 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 SPxBRGH:SPxBRGL register pair. The overflow condition will set the RCIF flag. The counter continues to count until the fifth rising edge is detected on the RX pin. The RCIDL bit will remain false ('0') until the fifth rising edge at which time the RCIDL bit will be set. If the RCREG is read after the overflow occurs but before the fifth rising edge then the fifth rising edge will set the RCIF again.

Terminating the auto-baud process early to clear an overflow condition will prevent proper detection of the sync character fifth rising edge. If any falling edges of the sync character have not yet occurred when the ABDEN bit is cleared then those will be falsely detected as Start bits. The following steps are recommended to clear the overflow condition:

- 1. Read RCREG to clear RCIF.
- 2. If RCIDL is zero then wait for RCIF and repeat step 1.
- 3. Clear the ABDOVF bit.

25.4.3 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 BAUDxCON 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 25-7), and asynchronously if the device is in Sleep mode (Figure 25-8). The interrupt condition is cleared by reading the RCxREG 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.

25.4.3.1 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.

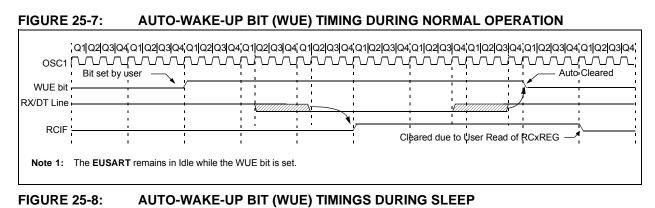
Oscillator Start-up Time

Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., 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 in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCxREG 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.



Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4 Q1 Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4 Bit Set by User Auto Cleared WUE bit RX/DT Line Note 1 RCIF Cleared due to User Read of RCxREG Sleep Command Executed Sleep Ends Note 1: If the wake-up event requires long oscillator warm-up time, the automatic clearing of the WUE bit can occur while the stpose signal is still active. This sequence should not depend on the presence of Q clocks.

2: The EUSART remains in Idle while the WUE bit is set.

25.4.4 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 TXxSTA register. The Break character transmission is then initiated by a write to the TXxREG. The value of data written to TXxREG 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 TXxSTA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 25-9 for the timing of the Break character sequence.

25.4.4.1 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 TXxREG with a dummy character to initiate transmission (the value is ignored).
- 4. Write '55h' to TXxREG 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 TXxREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXxREG.

25.4.5 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 RCxSTA register and the received data as indicated by RCxREG. 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
- RCxREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 25.4.3 "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 BAUDxCON register before placing the EUSART in Sleep mode.

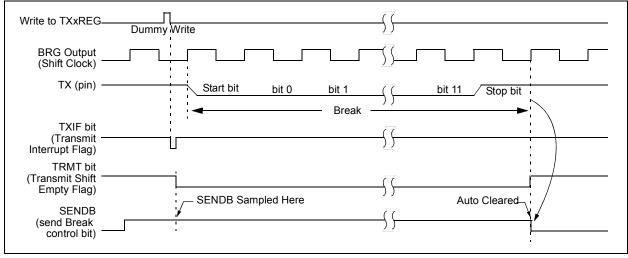


FIGURE 25-9: SEND BREAK CHARACTER SEQUENCE

25.5 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.

25.5.1 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 TXxSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXxSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCxSTA register enables the EUSART.

25.5.1.1 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.

25.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDxCON 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. 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.

25.5.1.3 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 TXxREG register. If the TSR still contains all or part of a previous character, the new character data is held in the TXxREG 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 TXxREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXxREG.

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.

- 25.5.1.4 Synchronous Master Transmission Set-up:
- Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 25.4 "EUSART Baud Rate Generator (BRG)").
- 2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 3. Disable Receive mode by clearing bits SREN and CREN.
- 4. Enable Transmit mode by setting the TXEN bit.
- 5. If 9-bit transmission is desired, set the TX9 bit.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
- 8. Start transmission by loading data to the TXxREG register.

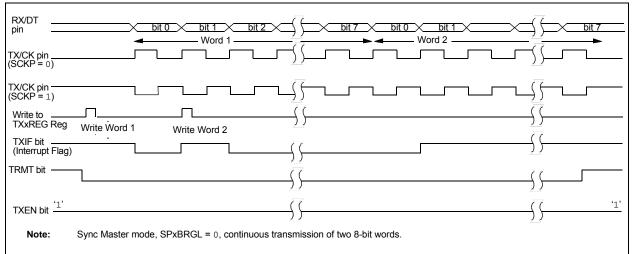


FIGURE 25-10: SYNCHRONOUS TRANSMISSION



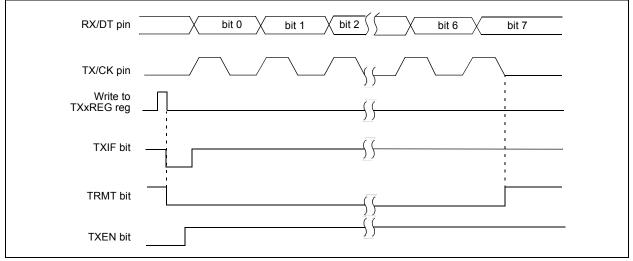


TABLE 25-7:SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER
TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	-	—		ANSA4	—	ANSA2	ANSA1	ANSA0	152
ANSELB ⁽¹⁾	_	_	ANSB5	ANSB4	_	_	_	_	159
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	-		ANSC3	ANSC2	ANSC1	ANSC0	166
BAUD1CON	ABDOVF	RCIDL		SCKP	BRG16		WUE	ABDEN	325
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	324
RxyPPS		—	-		F	RxyPPS<4:0	>		172
SP1BRGL				BRG<	:7:0>				326
SP1BRGH				BRG<	15:8>				326
TRISA	_	_	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	151
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	_	_	_	—	158
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5 TRISC4 TRISC3 TRISC2 TRISC1 TRISC0						
TX1REG	EUSART Transmit Data Register								315*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323

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

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

25.5.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled 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 RCxSTA register) or the Continuous Receive Enable bit (CREN of the RCxSTA 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 RCxREG. The RCIF bit remains set as long as there are unread characters in the receive FIFO.

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

25.5.1.6 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 is automatically disabled 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.

Note: If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSEL bit must be cleared.

25.5.1.7 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 RCxREG is read to access the FIFO. When this happens the OERR bit of the RCxSTA 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 RCxREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

25.5.1.8 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA 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 RCxREG.

- 25.5.1.9 Synchronous Master Reception Setup:
- 1. Initialize the SPxBRGH, SPxBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- 3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 4. Ensure bits CREN and SREN are clear.
- 5. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 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 RCxSTA 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 RCxREG register.
- 11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

1160KL 25-12.	STACHKOROUS RECEPTION (MASTER MODE, SREN)	
RX/DT pin TX/CK pin (SCKP = 0)	bit 0 bit 1 bit 2 bit 3 bit 4 bit 5 bit 6 bit 7	
TX/CK pin (SCKP = 1) Write to bit SREN		
SREN bit		
CREN bit		'0'
RCIF bit (Interrupt) ————		
Read RCxREG		
Note: Timing dia	agram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0 .	

FIGURE 25-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

TABLE 25-8: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	_	—		ANSA4	—	ANSA2	ANSA1	ANSA0	152
ANSELB ⁽¹⁾	_	—	ANSB5	ANSB4	—	—	_	—	159
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	_	—	ANSC3	ANSC2	ANSC1	ANSC0	166
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	325
CKPPS	-	—				CKPPS<4:0>			174, 172
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RC1REG	EUSART Receive Data Register							318*	
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	324
RXPPS	_	—	_	RXPPS<4:0>					174, 172
RxyPPS	_	—	_	RxyPPS<4:0>					172
SP1BRGL	BRG<7:0>						326*		
SP1BRGH	BRG<15:8>						326*		
TRISA	_	_	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	151
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous master reception.

* Page provides register information.

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

25.5.2 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 TXxSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXxSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCxSTA register enables the EUSART.

25.5.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see Section 25.5.1.3 "Synchronous Master Transmission"), except in the case of the Sleep mode.

If two words are written to the TXxREG 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 the TXxREG register.
- 3. The TXIF bit will not be set.
- After the first character has been shifted out of TSR, the TXxREG register will transfer the second character to the TSR and the TXIF bit will now be set.
- 5. 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.

- 25.5.2.2 Synchronous Slave Transmission Set-up:
- 1. Set the SYNC and SPEN bits and clear the CSRC bit.
- 2. Clear the ANSEL bit for the CK pin (if applicable).
- 3. Clear the CREN and SREN bits.
- 4. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 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 TXxREG register.

TABLE 25-9:SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE
TRANSMISSION

TRANSWISSION									
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	_	_	_	ANSA4	—	ANSA2	ANSA1	ANSA0	152
ANSELB ⁽¹⁾	_	_	ANSB5	ANSB4	—	_	-	_	159
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	_		ANSC3	ANSC2	ANSC1	ANSC0	166
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16		WUE	ABDEN	325
CKPPS	_	—	—			CKPPS<4:0>			174, 172
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	324
RXPPS	_	—	_			RXPPS<4:0>			174, 172
RxyPPS	_	_	_	RxyPPS<4:0>					172
TRISA	_	_	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	151
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	_	_	_	—	158
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
TX1REG	EUSART Transmit Data Register							315*	
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323

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

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

25.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (Section 25.5.1.5 "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 RCxREG 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.

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

TABLE 25-10: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	_	_	_	ANSA4	_	ANSA2	ANSA1	ANSA0	152
ANSELB ⁽¹⁾	_	_	ANSB5	ANSB4	—	_	_	—	159
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	_		ANSC3	ANSC2	ANSC1	ANSC0	166
BAUD1CON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	325
CKPPS	_	_	—	CKPPS<4:0>					174, 172
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	103
RC1REG	EUSART Receive Data Register							318*	
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	324
RXPPS	_	_	—	RXPPS<4:0>				174, 172	
TRISA	_	_	TRISA5	TRISA4	(2)	TRISA2	TRISA1	TRISA0	151
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	_		_	_	158
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	323

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave reception.

* Page provides register information.

Note 1: PIC16(L)F1619 only.

2: Unimplemented, read as '1'.

25.6 EUSART Operation During Sleep

The EUSART will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

25.6.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCxSTA and TXxSTA Control registers must be configured for Synchronous Slave Reception (see Section 25.5.2.4 "Synchronous Slave Reception Set-up:").
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RCxREG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set. Thereby, waking the processor from Sleep.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the Global Interrupt Enable (GIE) bit of the INTCON register is also set, then the Interrupt Service Routine at address 004h will be called.

25.6.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- The RCxSTA and TXxSTA Control registers must be configured for synchronous slave transmission (see Section 25.5.2.2 "Synchronous Slave Transmission Set-up:").
- The TXIF interrupt flag must be cleared by writing the output data to the TXxREG, thereby filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- Interrupt enable bits TXIE of the PIE1 register and PEIE of the INTCON register must set.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TXxREG will transfer to the TSR and the TXIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TXxREG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the Global Interrupt Enable (GIE) bit is also set then the Interrupt Service Routine at address 0004h will be called.

26.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains two standard Capture/ Compare/PWM modules (CCP1 and CCP2).

- Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.
 - 2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

26.1 Capture Mode

The Capture mode function described in this section is available and identical for all CCP modules.

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the CCPx input, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the MODE<3:0> bits of the CCPxCON register:

- Every edge (rising or falling)
- · Every falling edge
- Every rising edge
- Every 4th rising edge
- · Every 16th rising edge

The CCPx capture input signal is configured by the CTS bits of the CCPxCAP register with the following options:

- CCPx pin
- Comparator 1 output (C1_OUT_sync)
- Comparator 2 output (C2_OUT_sync)
- Interrupt-on-change interrupt trigger (IOC_interrupt)

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 26-1 shows a simplified diagram of the capture operation.

26.1.1 CCP PIN CONFIGURATION

In Capture mode, select the interrupt source using the CTS bits of the CCPxCAP register. If the CCPx pin is chosen, it should be configured as an input by setting the associated TRIS control bit.

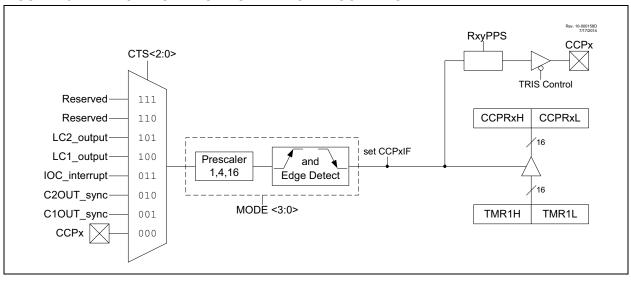


FIGURE 26-1: CAPTURE MODE OPERATION BLOCK DIAGRAM

26.1.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See Section22.0 "Timer1/3/5 Module with Gate Control" for more information on configuring Timer1.

26.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIEx register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIRx register following any change in Operating mode.

Note:	Clocking Timer1 from the system clock (Fosc) should not be used in Capture
	mode. In order for Capture mode to
	recognize the trigger event on the CCPx
	pin, Timer1 must be clocked from the
	instruction clock (Fosc/4) or from an
	external clock source.

26.1.4 CCP PRESCALER

There are four prescaler settings specified by the MODE<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the EN bit of the CCPxCON register before changing the prescaler.

26.1.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1 is clocked by Fosc/4, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep when Timer1 is clocked by an external clock source.

26.1.6 CAPTURE OUTPUT

Whenever a capture occurs, the output of the CCP will go high for a period equal to one system clock period (1/Fosc). This output is available as an input signal to the CWG, as an auto-conversion trigger for the ADC, as an External Reset Signal for the TMR2 modules, as a window input to the SMT, and as an input to the CLC module. In addition, the CCPx pin output can be mapped to output pins through the use of PPS (see **13.2 "PPS Outputs"**).

26.2 Compare Mode

The Compare mode function described in this section is available and identical for all CCP modules.

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMR1H:TMR1L register pair. When a match occurs, one of the following events can occur:

- · Toggle the CCPx output
- · Set the CCPx output
- Clear the CCPx output
- · Pulse the CCPx output
- · Generate a Software Interrupt
- Optionally Reset TMR1

The action on the pin is based on the value of the MODE<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set.

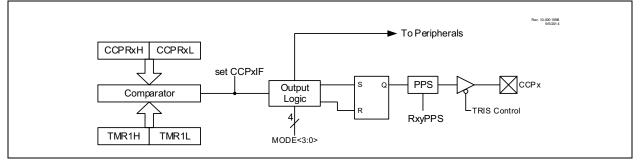
All Compare modes can generate an interrupt.

Figure 26-2 shows a simplified diagram of the compare operation.

26.2.1 CCPx PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

FIGURE 26-2: COMPARE MODE OPERATION BLOCK DIAGRAM



26.2.2 TIMER1 MODE RESOURCE

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See Section22.0 "Timer1/3/5 Module with Gate Control" for more information on configuring Timer1.

Note: Clocking Timer1 from the system clock (Fosc) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TImer1 must be clocked from the instruction clock (Fosc/4) or from an external clock source.

26.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (MODE<3:0> = 1010), the CCPx module does not assert control of the CCPx pin (see the CCPxCON register).

26.2.4 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (Fosc) for proper operation. Since Fosc is shut down during Sleep mode, the Compare mode will not function properly during Sleep.

26.2.5 CAPTURE OUTPUT

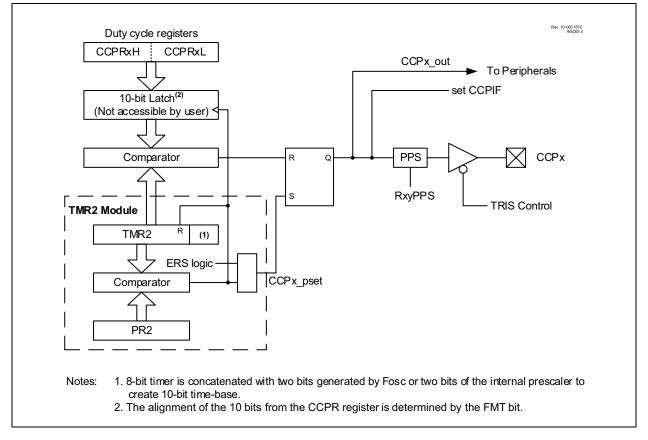
When in Compare mode, the CCP will provide an output upon the 16-bit value of the CCPRxH:CCPRxL register pair matching the TMR1H:TMR1L register pair. The compare output depends on which Compare mode the CCP is configured as. If the MODE bits of CCPxCON register are equal to '1011' or '1010', the CCP module will output high, while TMR1 is equal to CCPRxH:CCPRxL register pair. This means that the pulse width is determined by the TMR1 prescaler. If the MODE bits of CCPxCON are equal to '0001' or '0010', the output will toggle upon a match, going from '0' to '1' or vice-versa. If the MODE bits of CCPxCON are equal to '1001', the output is cleared on a match, and if the MODE bits are equal to '1000', the output is set on a match. This output is available as an input signal to the CWG, as an auto-conversion trigger for the ADC, as an external Reset signal for the TMR2 modules, as a window input to the SMT, and as an input to the CLC module. In addition, the CCPx pin output can be mapped to output pins through the use of PPS (see Section13.2 "PPS Outputs").

26.3 **PWM** Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the on state and the low portion of the signal is considered the off state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and in turn the power that is applied to the load. The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

FIGURE 26-3: SIMPLIFIED PWM BLOCK DIAGRAM



26.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- · PR2/4/6 registers
- T2CON/T4CON/T6CON registers
- CCPRxH:CCPRxL register pair

Figure shows a simplified block diagram of PWM operation.

- Note 1: The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.
 - 2: Clearing the CCPxCON register will relinquish control of the CCPx pin.

26.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

- 1. Disable the CCPx pin output driver by setting the associated TRIS bit.
- 2. Determine which timer will be used to clock the CCP; Timer2/4/6.
- 3. Load the associated PR2/4/6 register with the PWM period value.
- 4. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
- 5. Load the CCPRxH:CCPRxL register pair with the PWM duty cycle value.
- 6. Configure and start Timer2/4/6:
 - Clear the TMR2IF/TMR4IF/TMR6IF interrupt flag bit of the PIRx register. See Note below.
 - Configure the CKPS bits of the TxCON register with the Timer prescale value.
 - Enable the Timer by setting the ON bit of the TxCON register.
- 7. Enable PWM output pin:
 - Wait until the Timer overflows and the TMR2IF/TMR4IF/TMR6IF bit of the PIRx register is set. See Note below.
 - Enable the CCPx pin output driver by clearing the associated TRIS bit.
- **Note:** In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

26.4 CCP/PWM Clock Selection

The PIC16(L)F1615/9 allows each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

As there are up to three 8-bit timers with auto-reload (Timer2/4/6), PWM mode on the CCP and PWM modules can use any of these timers.

The CCPTMRS register is used to select which timer is used.

26.4.1 USING THE TMR2/4/6 WITH THE CCP MODULE

This device has a new version of the TMR2 module that has many new modes, which allow for greater customization and control of the PWM signals than older parts. Refer to **Section23.6 "Operation Examples"** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the Fosc/4 clock source selected.

26.4.2 PWM PERIOD

The PWM period is specified by the PR2/4/6 register of Timer2/4/6. The PWM period can be calculated using the formula of Equation 26-1.

EQUATION 26-1: PWM PERIOD

$$PWM Period = [(PR2) + 1] \bullet 4 \bullet Tosc \bullet$$
$$(TMR2 Prescale Value)$$

Note 1: Tosc = 1/Fosc

When TMR2/4/6 is equal to its respective PR2/4/6 register, the following three events occur on the next increment cycle:

- TMR2/4/6 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from the CCPRxH:CCPRxL pair into the internal 10-bit latch.

Note:	The Timer postscaler (see Figure) is not
	used in the determination of the PWM
	frequency.

26.4.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to two registers: the CCPRxH:CCPRxL register pair. Where the particular bits go is determined by the FMT bit of the CCPxCON register. If FMT = 0, the two Most Significant bits of the duty cycle value should be written to bits <1:0> of CCPRxH register and the remaining eight bits to the CCPRxL register. If FMT = 1, the Least Significant two bits of the duty cycle should be written to bits <7:6> of the CCPRxL register and the Most Significant eight bits to the CCPRxH register. This is illustrated in Figure 26-4. These bits can be written at any time. The duty cycle value is not latched into the internal latch until after the period completes (i.e., a match between PR2/4/6 and TMR2/4/6 registers occurs).

Equation 26-2 is used to calculate the PWM pulse width. Equation 26-3 is used to calculate the PWM duty cycle ratio.

EQUATION 26-2: PULSE WIDTH

• (TMR2 Prescale Value)

EQUATION 26-3: DUTY CYCLE RATIO

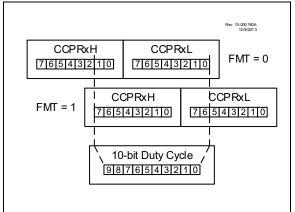
Duty Cycle Ratio =
$$\frac{(CCPRxH:CCPRxL)}{4(PRx+1)}$$

The PWM duty cycle registers are double buffered for glitchless PWM operation.

The 8-bit timer TMR2/4/6 register is concatenated with either the 2-bit internal system clock (Fosc), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2/4/6 prescaler is set to 1:1.

When the 10-bit time base matches the internal buffer register, then the CCPx pin is cleared (see Figure).

FIGURE 26-4: CCPx DUTY-CYCLE ALIGNMENT



26.4.4 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

© 2014-2017 Microchip Technology Inc.

The maximum PWM resolution is ten bits when PR2/4/6 is 255. The resolution is a function of the PR2/4/6 register value as shown by Equation 26-4.

EQUATION 26-4: PWM RESOLUTION

Resolution =
$$\frac{\log[4(PR2 + 1)]}{\log(2)}$$
 bits

Note: If the pulse width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

TADLE 20-1. EXAMIFLE FWW FREQUENCIES AND RESOLUTIONS (FUSC = 20 MITZ)	TABLE 26-1:	EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)
---	-------------	---

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6

TABLE 26-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

26.4.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section5.0 "Oscillator Module**" for additional details.

26.4.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

26.4.7 PWM OUTPUT

The output of the CCP in PWM mode is the PWM signal generated by the module and described above. This output is available as an input signal to the CWG, as an auto-conversion trigger for the ADC, as an external Reset signal for the TMR2 modules, as a window input to the SMT, and as an input to the CLC module. In addition, the CCPx pin output can be mapped to output pins through the use of PPS (see Section13.2 "PPS Outputs").

26.5 Register Definitions: CCP Control

R/W-0/0	U-0	R-x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	_	OUT	FMT		MODE	<3:0>	
bit 7							bit
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplen	nented bit, read	d as '0'	
u = Bit is unch	anged	x = Bit is unk	nown			R/Value at all o	other Reset
'1' = Bit is set	0	'0' = Bit is cle	ared				
bit 7		odule Enable b	.:4				
DIL 7	1 = CCPx is		אנ				
	1 = CCP x is 0 = CCP x is						
bit 6	Unimplemen	ted: Read as '	0'				
bit 5	-	Dutput Data bit					
bit 4		(Pulse-Width)					
	If MODE = P\	. ,	5				
				he MSb of the F the LSb of the F			
bit 3-0	MODE<3:0>:	CCPx Mode S	Selection bit				
	11xx = PWM	1 mode					
	1010 = Com	pare mode: Ρι	Ilse output, cle Ilse output (0 -				
			t output on cor				
	0110 = Capt 0101 = Capt	ure mode: eve ure mode: eve		dge			
	0100 = Capt	ure mode: eve	ry falling edge				
			ry rising or fall				
			ggle output on				
	0001 = Com 0000 = Capt			d clear TMR1 c			

REGISTER 26-1: CCPxCON: CCPx CONTROL 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
P4TS	EL<1:0>	P3TSE	L<1:0>	C2TSE	L<1:0>	C1TSE	EL<1:0>
bit 7		1					bit 0
Legend:							
R = Readabl	e bit	W = Writable I	oit	U = Unimplen	nented bit, read	d as '0'	
u = Bit is und	hanged	x = Bit is unkn	own	-n/n = Value a	at POR and BO	R/Value at all	other Resets
'1' = Bit is se	t	'0' = Bit is clea	ared				
bit 7-6	P4TSEL<1:0	>: PWM4 Time	r Selection bit	S			
	11 = Reserv						
		is based off Tir					
		is based off Tir					
bit 5-4		>: PWM3 Time					
DIL 3-4				5			
	11 = Reserv 10 = PWM3	veu 8 is based off Tir	mer6 in PWM	mode			
		is based off Tir					
		is based off Tir					
bit 3-2	C2TSEL<1:0	>: CCP2 (PWN	12) Timer Sele	ection bits			
	11 = Reserv	ved					
	10 = CCP2	is based off Tim	ner6 in PWM r	node			
01 = CCP2 is based off Timer4 in PWM mode							
	00 = CCP2 is based off Timer2 in PWM mode						
bit 1-0	C1TSEL<1:0	>: CCP1 (PWN	11) Timer Sele	ection bits			
	11 = Reserv						
		is based off Tim					
		01 = CCP1 is based off Timer4 in PWM mode					
	00 = 00PI	00 = CCP1 is based off Timer2 in PWM mode					

REGISTER 26-2: CCPTMRS: PWM TIMER SELECTION CONTROL REGISTER 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	R/W-0/0	
	CC							
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable	bit	U = Unimplen	nented bit, read	d as '0'		
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value a	other Reset			
'1' = Bit is set		'0' = Bit is clea	ared					
bit 7-0	MODE = Ca	<u>pture Mode</u>						
	CCPRxL<7:	0>: LSB of capt	ured TMR1 va	alue				
	MODE = Compare Mode							
	CCPRxL<7:0>: LSB compared to TMR1 value							
	MODE = PWM Mode && FMT = 0							
	CCPRxL<7:0>: CCPW<7:0> — Pulse width Least Significant eight bits							
	MODE = PW	/M Mode && FM	<u>1T = 1</u>					

CCPRxL<7:6>: CCPW<1:0> — Pulse width Least Significant two bits

REGISTER 26-3: CCPRxL: CCPx LOW BYTE REGISTER

CCPRxL<5:0>: Not used

© 2014-2017 Microchip Technology Inc.

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
			CCPR	<15:8>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	d as '0'	
u = Bit is unch	nanged	x = Bit is unkn	iown	-n/n = Value a	at POR and BC	R/Value at all	other Reset
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7-0	MODE = Ca	pture Mode:					
CCPRxH<7:0>: MSB of captured TMR1 value							

REGISTER 26-4: CCPRxH: CCPx HIGH BYTE REGISTER

MODE = Capture Mode:
 CCPRxH<7:0>: MSB of captured TMR1 value
 MODE = Compare Mode:
 CCPRxH<7:0>: MSB compared to TMR1 value
 MODE = PWM Mode && FMT = 0:
 CCPRxH<7:2>: Not used
 CCPRxH<7:2>: Not used
 CCPRxH<1:0>: CCPW<9:8> — Pulse width Most Significant two bits
 MODE = PWM Mode && FMT = 1:
 CCPRxH<7:0>: CCPW<9:2> — Pulse width Most Significant eight bits

REGISTER 26-5: CCPxCAP: CCPx CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—		CTS<2:0>	
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 Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-3	Unimplemented: Read as '0'
---------	----------------------------

bit 2-0 CTS<2:0>: Capture Trigger Input Selection bits

- 111 = LC4_out
 - 110 = LC3 out
 - 101 = LC2_out
 - 100 = LC1_out
 - 011 = IOC_interrupt
 - 010 = C2_OUT_sync
 - 001 = C1_OUT_sync
 - 000 = CCPx pin

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCAP	_	CTS<1:0>							
CCP1CON	EN	EN – OUT FMT MODE<3:0>							354
CCP2CON	EN	EN - OUT FMT MODE<3:0>							354
CCPRxL	Capture/Comp	Capture/Compare/PWM Register x (LSB)							
CCPRxH	Capture/Comp	are/PWM Regis	ter x (MSB)						357
CCPTMRS	P4TSE	L<1:0>	P3TSE	L<1:0>	C2TSE	L<1:0>	C1TSE	L<1:0>	355
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIE2	OSFIE	C2IE	C1IE	_	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PR2	Timer2 Period	Register							244*
T2CON	ON		CKPS<2:0>			OUTP	S<3:0>		256
TMR2	Timer2 Module	e Register							236*
PR4	Timer4 Period	Register							244*
T4CON	ON		CKPS<2:0>			OUTP	S<3:0>		256
TMR4	Timer4 Module	e Register							236*
PR6	Timer6 Period Register							244*	
T6CON	ON CKPS<2:0> OUTPS<3:0>							256	
TMR6	Timer6 Module Register							236*	
TRISA	_	_	TRISA5	TRISA4	_(1)	TRISA2	TRISA1	TRISA0	151

TABLE 26-3: SUMMARY OF REGISTERS ASSOCIATED WITH STANDARD PWM

Legend: -= Unimplemented location, read as '0'. Shaded cells are not used by the PWM. * Page provides register information. Note 1: Unimplemented, read as '1'.

© 2014-2017 Microchip Technology Inc.

27.0 PULSE-WIDTH MODULATION (PWM) MODULE

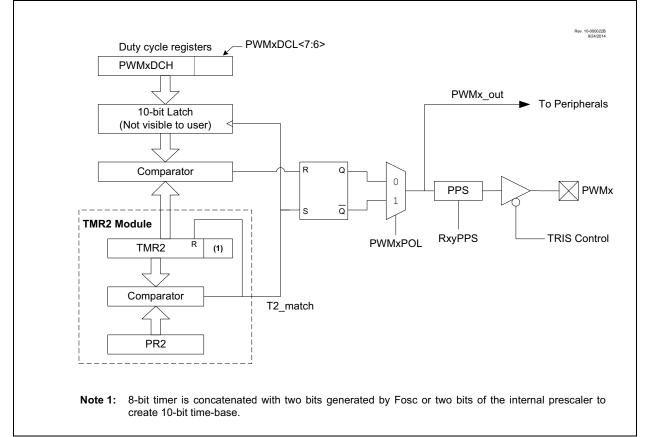
The PWM module generates a Pulse-Width Modulated signal determined by the duty cycle, period, and resolution that are configured by the following registers:

- PR2
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

Figure 27-1 shows a simplified block diagram of PWM operation.

For a step-by-step procedure on how to set up this module for PWM operation, refer to Section 27.1.9 "Setup for PWM Operation using PWMx Pins".

FIGURE 27-1: SIMPLIFIED PWM BLOCK DIAGRAM



27.1 **PWMx Pin Configuration**

All PWM outputs are multiplexed with the PORT data latch. The user must configure the pins as outputs by clearing the associated TRIS bits.

27.1.1 FUNDAMENTAL OPERATION

The PWM module produces a 10-bit resolution output. Timer2 and PR2 set the period of the PWM. The PWMxDCL and PWMxDCH registers configure the duty cycle. The period is common to all PWM modules, whereas the duty cycle is independently controlled.

Note:	The Timer2 postscaler is not used in the
	determination of the PWM frequency. The
	postscaler could be used to have a servo
	update rate at a different frequency than
	the PWM output.

All PWM outputs associated with Timer2 are set when TMR2 is cleared. Each PWMx is cleared when TMR2 is equal to the value specified in the corresponding PWMxDCH (8 MSb) and PWMxDCL<7:6> (2 LSb) registers. When the value is greater than or equal to PR2, the PWM output is never cleared (100% duty cycle).

Note: The PWMxDCH and PWMxDCL registers are double buffered. The buffers are updated when Timer2 matches PR2. Care should be taken to update both registers before the timer match occurs.

27.1.2 PWM OUTPUT POLARITY

The output polarity is inverted by setting the PWMxPOL bit of the PWMxCON register.

27.1.3 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 27-1.

EQUATION 27-1: PWM PERIOD

 $PWM Period = [(PR2) + 1] \bullet 4 \bullet Tosc \bullet$ (TMR2 Prescale Value)

Note: Tosc = 1/Fosc

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The PWM output is active. (Exception: When the PWM duty cycle = 0%, the PWM output will remain inactive.)
- The PWMxDCH and PWMxDCL register values are latched into the buffers.

Note:	The Timer2 postscaler has no effect on			
	the PWM operation.			

27.1.4 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDCH and PWMxDCL register pair. The PWMxDCH register contains the eight MSbs and the PWMxDCL<7:6>, the two LSbs. The PWMxDCH and PWMxDCL registers can be written to at any time.

Equation 27-2 is used to calculate the PWM pulse width.

Equation 27-3 is used to calculate the PWM duty cycle ratio.

EQUATION 27-2: PULSE WIDTH

 $Pulse Width = (PWMxDCH:PWMxDCL<7:6>) \bullet$

TOSC • (TMR2 Prescale Value)

Note: Tosc = 1/Fosc

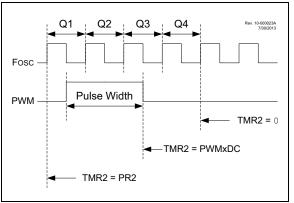
EQUATION 27-3: DUTY CYCLE RATIO

$$Duty Cycle Ratio = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(PR2+1)}$$

The 8-bit timer TMR2 register is concatenated with the two Least Significant bits of 1/Fosc, adjusted by the Timer2 prescaler to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

Figure 27-2 shows a waveform of the PWM signal when the duty cycle is set for the smallest possible pulse.

FIGURE 27-2: PWM OUTPUT



27.1.5 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 27-4.

EQUATION 27-4: PWM RESOLUTION

Resolution = $\frac{\log[4(PR2 + 1)]}{\log(2)}$ bits

Note: If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

TABLE 27-1:EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)

PWM Frequency	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	64	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 27-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

PWM Frequency	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	64	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

27.1.6 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

27.1.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency (Fosc). Any changes in the system clock frequency will result in changes to the PWM frequency. Refer to **Section 5.0 "Oscillator Module"** for additional details.

27.1.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the PWM registers to their Reset states.

27.1.9 SETUP FOR PWM OPERATION USING PWMx PINS

The following steps should be taken when configuring the module for PWM operation using the PWMx pins:

- 1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
- 2. Clear the PWMxCON register.
- 3. Load the PR2 register with the PWM period value.
- 4. Clear the PWMxDCH register and bits <7:6> of the PWMxDCL register.
- 5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR1 register. See note below.
 - Configure the CKPS bits of the T2CON register with the Timer2 prescale value.
 - Enable Timer2 by setting the ON bit of the T2CON register.
- 6. Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See note below.
- 7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the PWMxOE bit of the PWMxCON register.
- 8. Configure the PWM module by loading the PWMxCON register with the appropriate values.
 - Note 1: In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.
 - 2: For operation with other peripherals only, disable PWMx pin outputs.

27.2 Register Definitions: PWM Control

U-0	U-0	U-0	U-0	
_				
	—	—	—	
			bit 0	
U = Unimpler	nented bit, read	l as '0'		
-n/n = Value a	at POR and BO	R/Value at all o	other Resets	
0 = PWM module is disabled				
it				
bit				
1 = PWM output is active-low				

REGISTER 27-1: PWMxCON: PWM CONTROL REGISTER

bit 3-0 Unimplemented: Read as '0'

REGISTER 27-2: PWMxDCH: PWM DUTY CYCLE HIGH BITS

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
			PWMxD	CH<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

PWMxDCH<7:0>: PWM Duty Cycle Most Significant bits

These bits are the MSbs of the PWM duty cycle. The two LSbs are found in the PWMxDCL register.

REGISTER 27-3: PWMxDCL: PWM DUTY CYCLE LOW BITS

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxD	CL<7:6>	_	_	—	—	—	—
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	PWMxDCL<7:6>: PWM Duty Cycle Least Significant bits
	These bits are the LSbs of the PWM duty cycle. The MSbs are found in the PWMxDCH register.
bit 5-0	Unimplemented: Read as '0'

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PR2			Tin	ner2 module F	Period Registe	er			361*
PWM3CON	EN	_	OUT	POL	_	_	_		363
PWM3DCH				DC<9):2>				363
PWM3DCL	DC<	1:0>	—	—	_	_	_	_	363
PWM4CON	EN	_	OUT	POL	_	_	_	_	363
PWM4DCH				DC<9):2>				363
PWM4DCL	DC<	1:0>	—	—	_	_	_	_	363
T2CON	ON		CKPS<2:0>			OUTPS	S<3:0>		256
TMR2				Timer2 modu	le Register				236*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	151
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165
		6 II (C				<u> </u>			

TABLE 27-3: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM. * Page provides register information.

Note 1: Unimplemented, read as '1'.

2: PIC16(L)F1619 only.

28.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces half-bridge, full-bridge, and steering of PWM waveforms. It is backwards compatible with previous ECCP functions.

The CWG has the following features:

- · Six operating modes:
 - Synchronous Steering mode
 - Asynchronous Steering mode
 - Full-Bridge mode, Forward
 - Full-Bridge mode, Reverse
 - Half-Bridge mode
 - Push-Pull mode
- Output polarity control
- · Output steering
 - Synchronized to rising event
 - Immediate effect
- Independent 6-bit rising and falling event deadband timers
 - Clocked dead band
 - Independent rising and falling dead-band enables
- Auto-shutdown control with:
 - Selectable shutdown sources
 - Auto-restart enable
 - Auto-shutdown pin override control

28.1 Fundamental Operation

The CWG module can operate in six different modes, as specified by MODE of the CWGxCON0 register:

- Half-Bridge mode (Figure 28-9)
- Push-Pull mode (Figure 28-2)
 - Full-Bridge mode, Forward (Figure 28-3)
 - Full-Bridge mode, Reverse (Figure 28-3)
- Steering mode (Figure 28-10)
- Synchronous Steering mode (Figure 28-11)

It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. Thus, all output modes support auto-shutdown, which is covered in **28.10** "Auto-Shutdown".

28.1.1 HALF-BRIDGE MODE

In Half-Bridge mode, two output signals are generated as true and inverted versions of the input as illustrated in Figure 28-9. A non-overlap (dead-band) time is inserted between the two outputs to prevent shoot through current in various power supply applications. Dead-band control is described in **Section 28.5 "Dead-Band Control"**.

The unused outputs CWGxC and CWGxD drive similar signals, with polarity independently controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

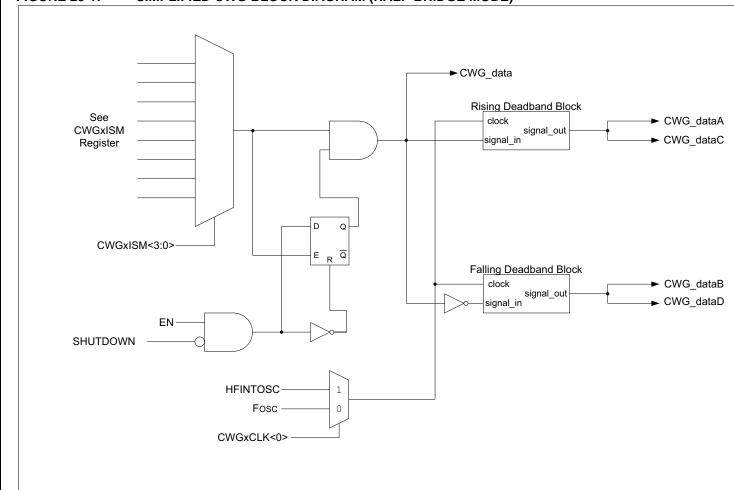


FIGURE 28-1: SIMPLIFIED CWG BLOCK DIAGRAM (HALF-BRIDGE MODE)

© 2014-2017 Microchip Technology Inc.

DS40001770D-page 366

28.1.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 28-2. This alternation creates the push-pull effect required for driving some transformer-based power supply designs.

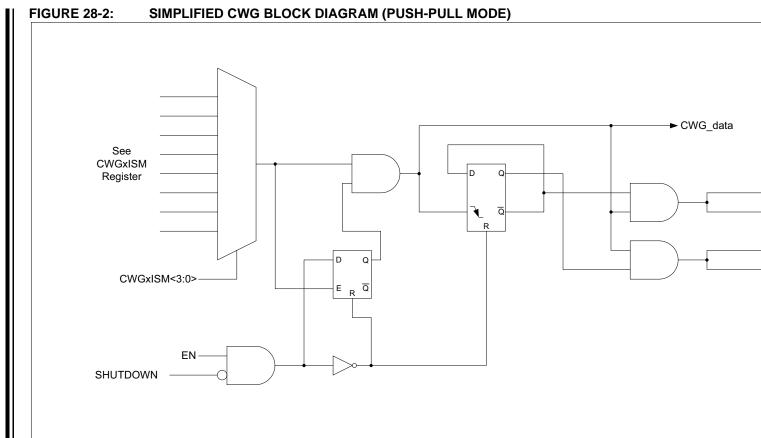
The push-pull sequencer is reset whenever EN = 0 or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWGxA.

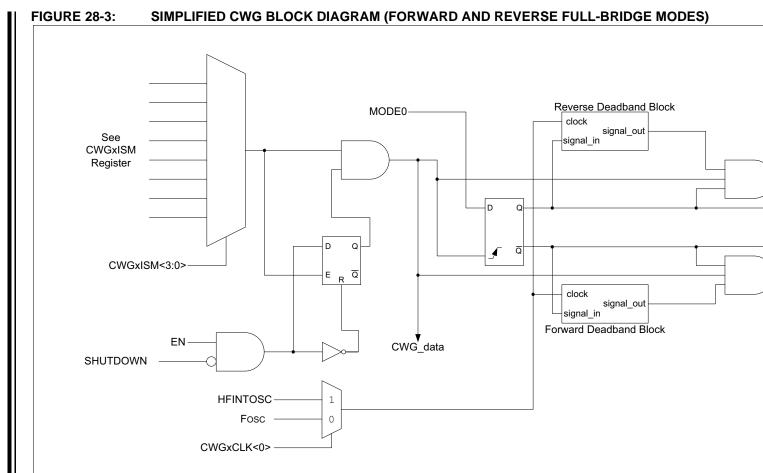
The unused outputs CWGxC and CWGxD drive copies of CWGxA and CWGxB, respectively, but with polarity controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

28.1.3 FULL-BRIDGE MODES

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the input data signal. In Forward Full-Bridge mode, CWGxA is driven to its active state, CWGxB and CWGxC are driven to their inactive state, and CWGxD is modulated by the input signal. In Reverse Full-Bridge mode, CWGxC is driven to its active state, CWGxA and CWGxD are driven to their inactive states, and CWGxB is modulated by the input signal. In Full-Bridge mode, the dead-band period is used when there is a switch from forward to reverse or vice-versa. This dead-band control is described in Section 28.5 "Dead-Band Control", with additional details in Section 28.6 "Rising Edge and Reverse Dead Band" and Section 28.7 "Falling Edge and Forward Dead Band".

The mode selection may be toggled between forward and reverse by toggling the MODE<0> bit of the CWGxCON0 while keeping MODE<2:1> static, without disabling the CWG module.



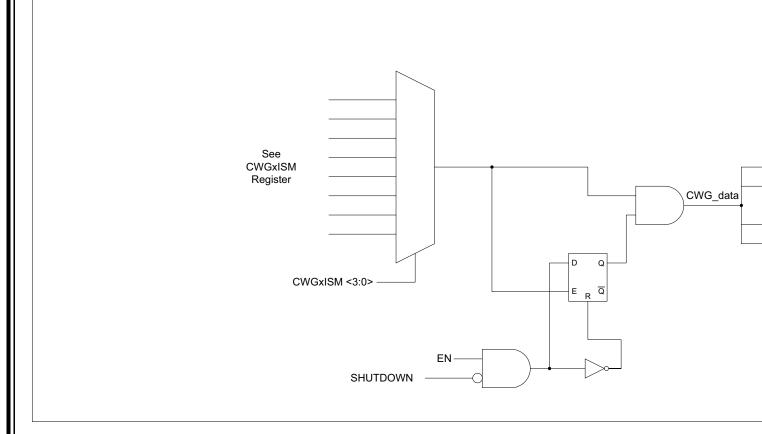


28.1.4 STEERING MODES

In Steering modes, the data input can be steered to any or all of the four CWG output pins. In Synchronous Steering mode, changes to steering selection registers take effect on the next rising input.

In Non-Synchronous mode, steering takes effect on the next instruction cycle. Additional details are provided in **Section 28.9 "CWG Steering Mode**".





28.2 Clock Source

The CWG module allows the following clock sources to be selected:

- Fosc (system clock)
- HFINTOSC (16 MHz only)

The clock sources are selected using the CS bit of the CWGxCLKCON register.

28.3 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 28-1.

TABLE 28-1: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name
CWG pin	PPS selection
Comparator C1	C1_OUT_sync
Comparator C2	C2_OUT_sync
CCP1	CCP1_out
CCP2	CCP2_out
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
CLC4	LC4_out
PWM3	PWM3_out
PWM4	PWM4_out

The input sources are selected using the CWGxISM register.

28.4 Output Control

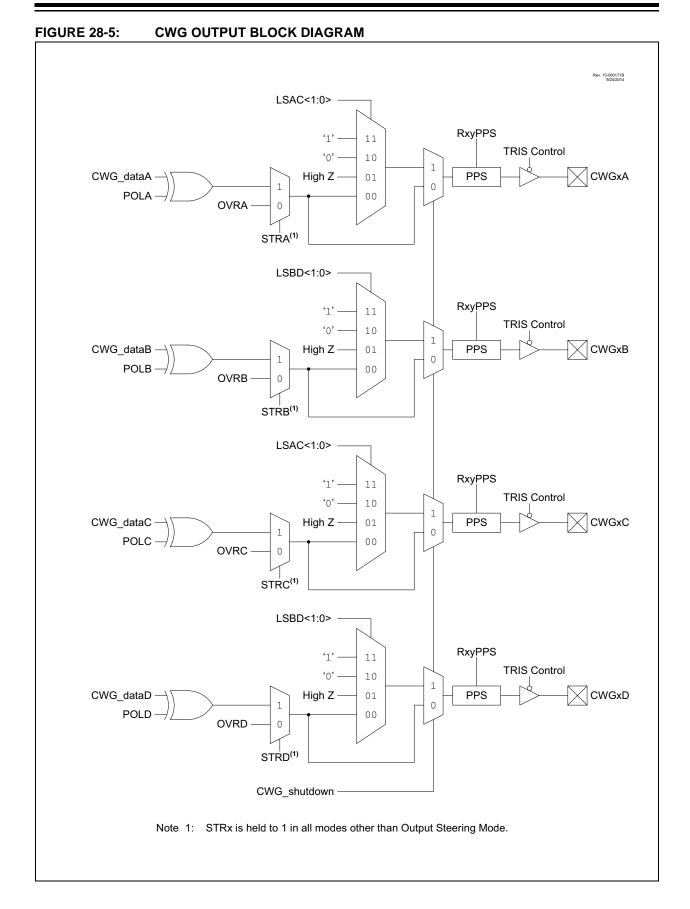
28.4.1 OUTPUT ENABLES

Each CWG output pin has individual output enable control. Output enables are selected with the Gx1OEx <3:0> bits. When an output enable control is cleared, the module asserts no control over the pin. When an output enable is set, the override value or active PWM waveform is applied to the pin per the port priority selection. The output pin enables are dependent on the module enable bit, EN of the CWGxCON0 register. When EN is cleared, CWG output enables and CWG drive levels have no effect.

28.4.2 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the POLx bits of the CWGxCON1. Auto-shutdown and steering options are unaffected by polarity.

PIC16(L)F1615/9



28.5 Dead-Band Control

The dead-band control provides non-overlapping PWM signals to prevent shoot-through current in PWM switches. Dead-band operation is employed for Half-Bridge and Full-Bridge modes. The CWG contains two 6-bit dead-band counters. One is used for the rising edge of the input source control in Half-Bridge mode or for reverse dead-band Full-Bridge mode. The other is used for the falling edge of the input source control in Half-Bridge mode or for forward dead band in Full-Bridge mode.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling deadband counter registers. See CWGxDBR and CWGxDBF registers, respectively.

28.5.1 DEAD-BAND FUNCTIONALITY IN HALF-BRIDGE MODE

In Half-Bridge mode, the dead-band counters dictate the delay between the falling edge of the normal output and the rising edge of the inverted output. This can be seen in Figure 28-9.

28.5.2 DEAD-BAND FUNCTIONALITY IN FULL-BRIDGE MODE

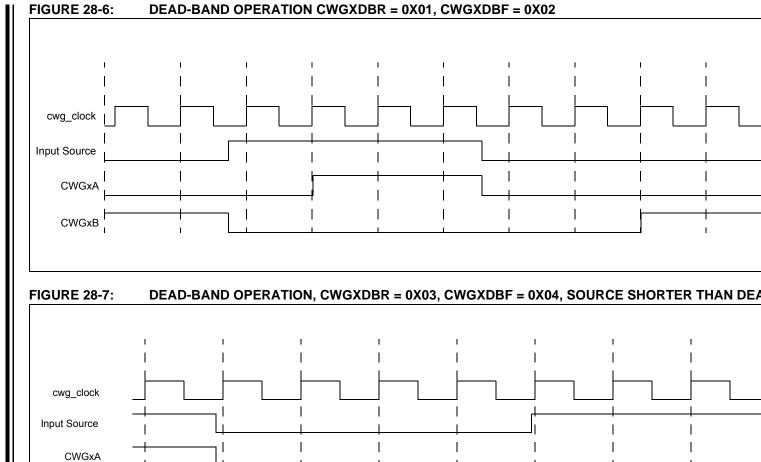
In Full-Bridge mode, the dead-band counters are used when undergoing a direction change. The MODE<0> bit of the CWGxCON0 register can be set or cleared while the CWG is running, allowing for changes from Forward to Reverse mode. The CWGxA and CWGxC signals will change immediately upon the first rising input edge following a direction change, but the modulated signals (CWGxB or CWGxD, depending on the direction of the change) will experience a delay dictated by the dead-band counters. This is demonstrated in Figure 28-3.

28.6 Rising Edge and Reverse Dead Band

CWGxDBR controls the rising edge dead-band time at the leading edge of CWGxA (Half-Bridge mode) or the leading edge of CWGxB (Full-Bridge mode). The CWGxDBR value is double-buffered. When EN = 0, the CWGxDBR register is loaded immediately when CWGxDBR is written. When EN = 1, then software must set the LD bit of the CWGxCON0 register, and the buffer will be loaded at the next falling edge of the CWG input signal. If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

28.7 Falling Edge and Forward Dead Band

CWGxDBF controls the dead-band time at the leading edge of CWGxB (Half-Bridge mode) or the leading edge of CWGxD (Full-Bridge mode). The CWGxDBF value is double-buffered. When EN = 0, the CWGxDBF register is loaded immediately when CWGxDBF is written. When EN = 1 then software must set the LD bit of the CWGxCON0 register, and the buffer will be loaded at the next falling edge of the CWG input signal. If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output. Refer to Figure 28.6 and Figure 28-7 for examples.



I

I

L

source shorter than dead band

T

L

I

L

CWGxB

28.8 Dead-Band Uncertainty

When the rising and falling edges of the input source are asynchronous to the CWG clock, it creates uncertainty in the dead-band time delay. The maximum uncertainty is equal to one CWG clock period. Refer to Equation 28-1 for more details.

EQUATION 28-1: DEAD-BAND UNCERTAINTY

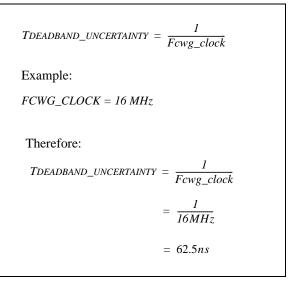
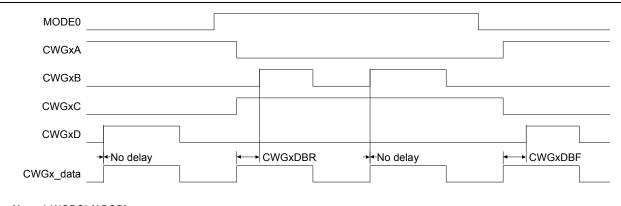


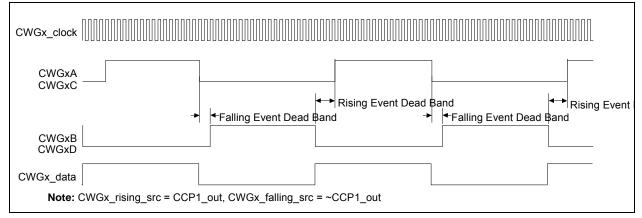
FIGURE 28-8: EXAMPLE OF PWM DIRECTION CHANGE



Note 1:WGPOL{ABCD} = 0

- 2: The direction bit MODE<0> (Register 28-1) can be written any time during the PWM cycle, and takes effect at the next rising CWGx_data.
- 3: When changing directions, CWGxA and CWGxC switch at rising CWGx_data; modulated CWGxB and CWGxD are held inactive for the dead band duration shown; dead band affects only the first pulse after the direction change.

FIGURE 28-9: CWG HALF-BRIDGE MODE OPERATION



28.9 CWG Steering Mode

In Steering mode (MODE = 00x), the CWG allows any combination of the CWGxx pins to be the modulated signal. The same signal can be simultaneously available on multiple pins, or a fixed-value output can be presented.

When the respective STRx bit of CWGxOCON0 is '0', the corresponding pin is held at the level defined. When the respective STRx bit of CWGxOCON0 is '1', the pin is driven by the input data signal. The user can assign the input data signal to one, two, three, or all four output pins.

The POLx bits of the CWGxCON1 register control the signal polarity only when STRx = 1.

The CWG auto-shutdown operation also applies in Steering modes as described in Section 28.10 "Auto-Shutdown". An auto-shutdown event will only affect pins that have STRx = 1.

28.9.1 STEERING SYNCHRONIZATION

Changing the MODE bits allows for two modes of steering, synchronous and asynchronous.

When MODE = 000, the steering event is asynchronous and will happen at the end of the instruction that writes to STRx (that is, immediately). In this case, the output signal at the output pin may be an incomplete waveform. This can be useful for immediately removing a signal from the pin.

When MODE = 001, the steering update is synchronous and occurs at the beginning of the next rising edge of the input data signal. In this case, steering the output on/off will always produce a complete waveform.

Figure 28-10 and Figure 28-11 illustrate the timing of asynchronous and synchronous steering, respectively.



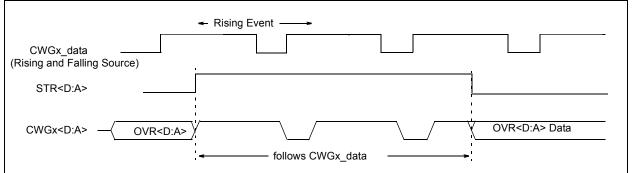
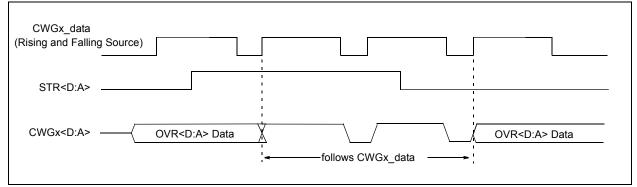


FIGURE 28-11: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (MODE<2:0> = 001)



28.10 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in Figure 28-12.

28.10.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

28.10.1.1 Software Generated Shutdown

Setting the SHUTDOWN bit of the CWGxAS0 register will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event.

28.10.2 EXTERNAL INPUT SOURCE

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the selected override levels without software delay. Several input sources can be selected to cause a shutdown condition. All input sources are active-low. The sources are:

- Comparator C1_OUT_sync
- Comparator C2_OUT_sync
- Timer2 TMR2_postscaled
- Timer4 TMR4 postscaled
- Timer6 TMR6 postscaled
- CWGxIN input pin

Shutdown inputs are selected using the CWGxAS1 register (Register 28-6).

Note:	Shutdown inputs are level sensitive, not
	edge sensitive. The shutdown state can-
	not be cleared, except by disabling auto-
	shutdown, as long as the shutdown input
	level persists.

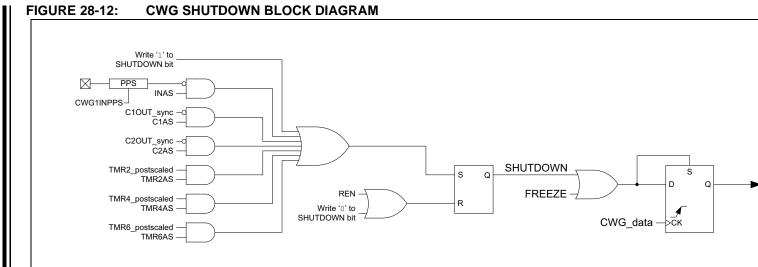
28.11 Operation During Sleep

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

The HFINTOSC remains active during Sleep when all the following conditions are met:

- · CWG module is enabled
- Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.



© 2014-2017 Microchip Technology Inc.

28.12 Configuring the CWG

The following steps illustrate how to properly configure the CWG.

- 1. Ensure that the TRIS control bits corresponding to the desired CWG pins for your application are set so that the pins are configured as inputs.
- 2. Clear the EN bit, if not already cleared.
- 3. Set desired mode of operation with the MODE bits.
- Set desired dead-band times, if applicable to mode, with the CWGxDBR and CWGxDBF registers.
- 5. Setup the following controls in the CWGxAS0 and CWGxAS1 registers.
 - a. Select the desired shutdown source.
 - b. Select both output overrides to the desired levels (this is necessary even if not using autoshutdown because start-up will be from a shutdown state).
 - c. Set which pins will be affected by auto-shutdown with the CWGxAS1 register.
 - d. Set the SHUTDOWN bit and clear the REN bit.
- 6. Select the desired input source using the CWGxISM register.
- 7. Configure the following controls.
 - a. Select desired clock source using the CWGxCLKCON register.
 - b. Select the desired output polarities using the CWGxCON1 register.
 - c. Set the output enables for the desired outputs.
- 8. Set the EN bit.
- 9. Clear TRIS control bits corresponding to the desired output pins to configure these pins as outputs.
- If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit to start the CWG.

28.12.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the LSBD and LSAC bits of the CWGxAS0 register. LSBD<1:0> controls the CWGxB and D override levels and LSAC<1:0> controls the CWGxA and C override levels. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not affect the override level.

28.12.2 AUTO-SHUTDOWN RESTART

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

- Software controlled
- Auto-restart

The restart method is selected with the REN bit of the CWGxAS0 register. Waveforms of software controlled and automatic restarts are shown in Figure 28-13 and Figure 28-14.

28.12.2.1 Software Controlled Restart

When the REN bit of the CWGxAS0 register is cleared, the CWG must be restarted after an auto-shutdown event by software. Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the SHUTDOWN bit will remain set. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

28.12.2.2 Auto-Restart

When the REN bit of the CWGxAS0 register is set, the CWG will restart from the auto-shutdown state automatically. The SHUTDOWN bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

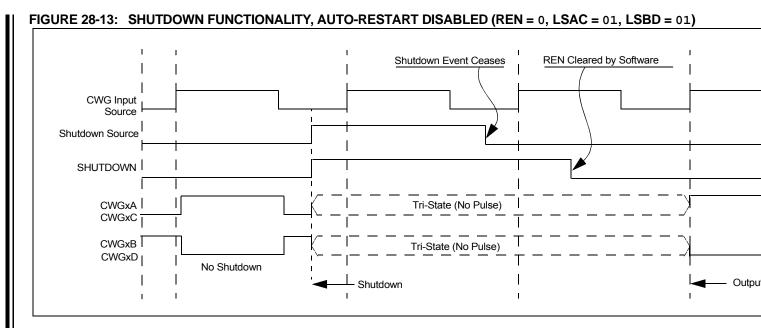
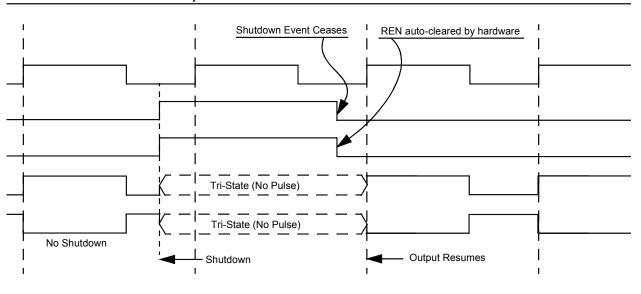


FIGURE 28-14: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (REN = 1, LSAC = 01, LSBD = 01)



28.13 Register Definitions: CWG Control

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	
EN	LD ⁽¹⁾	_	_	_		MODE<2:0>		
bit 7	•				•		bit C	
Legend:								
HC = Bit is cle	eared by hardwa	re		HS = Bit is set by hardware				
R = Readable	e bit	W = Writable I	bit	U = Unimplemented bit, read as '0'				
u = Bit is unch	nanged	x = Bit is unkn	nown	-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	EN: CWGx Enable bit 1 = Module is enabled 0 = Module is disabled
bit 6	 LD: CWGx Load Buffer bits⁽¹⁾ 1 = Buffers to be loaded on the next rising/falling event 0 = Buffers not loaded
bit 5-3	Unimplemented: Read as '0'
bit 2-0	MODE<2:0>: CWGx Mode bits 111 = Reserved 110 = Reserved 101 = CWG outputs operate in Push-Pull mode 100 = CWG outputs operate in Half-Bridge mode 011 = CWG outputs operate in Reverse Full-Bridge mode 010 = CWG outputs operate in Forward Full-Bridge mode 001 = CWG outputs operate in Synchronous Steering mode 000 = CWG outputs operate in Steering mode

Note 1: This bit can only be set after EN = 1 and cannot be set in the same instruction that EN is set.

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	_	POLD	POLC	POLB	POLA
bit 7							bit 0

Legend:						
R = Readable bit W = Writable		W = Writable bit	U = Unimplemented bit, read as '0'			
u = Bit is unchanged x = Bit is unknown		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 7-6	Unimplemer	nted: Read as '0'				
bit 5	IN: CWG Inp	ut Value				
bit 4	Unimplemer	nted: Read as '0'				
bit 3	POLD: CWG	SxD Output Polarity bit				
	1 = Signal o	utput is inverted polarity				
	0 = Signal o	utput is normal polarity				
bit 2	POLC: CWG	SxC Output Polarity bit				
	•	utput is inverted polarity				
	0 = Signal o	utput is normal polarity				
bit 1	POLB: CWG	SxB Output Polarity bit				
	•	utput is inverted polarity				
	0 = Signal o	utput is normal polarity				
bit 0	POLA: CWG	SxA Output Polarity bit				
	•	utput is inverted polarity				
	0 = Signal o	utput is normal polarity				

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				
—	_		DBR<5:0>								
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 Re							ther Resets				

q = Value depends on condition

bit 7-6	Unimplemented: Read as '0'

1' = Bit is set

bit 5-0 DBR<5:0>: Rising Event Dead-Band Value for Counter bits

'0' = Bit is cleared

REGISTER 28-4: CWGxDBF: CWGx FALLING DEAD-BAND COUNTER 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
—	—			DBF	<5:0>		
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	q = Value depends on condition

bit 7-6 Unimplemented: Read as '0'

bit 5-0 DBF<5:0>: Falling Event Dead-Band Value for Counter bits

R/W/HS-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0		
SHUTDOWN ^(1, 2) REN			LSBD<1:0> LSAC<1:0>						
bit 7							bit (
Legend:									
HC = Bit is cleared	HS = Bit is se	et by hardware	9						
R = Readable bit		W = Writable	bit	U = Unimple	mented bit, rea	ad as '0'			
u = Bit is unchang	ed	x = Bit is unk	nown	-n/n = Value	at POR and B	OR/Value at all	other Reset		
'1' = Bit is set		'0' = Bit is cle	ared	q = Value de	pends on conc	lition			
			_	(1 2)					
bit 7	SHUTDOWN: Auto-Shutdown Event Status bit ^(1, 2)								
	 1 = An Auto-Shutdown state is in effect 0 = No Auto-shutdown event has occurred 								
bit 6									
	REN: Auto-Restart Enable bit 1 = Auto-restart enabled								
	1 = Auto-restart enabled 0 = Auto-restart disabled								
bit 5-4	LSBD<1:0>: CWGxB and CWGxD Auto-Shutdown State Control bits								
	$11 = A \log (1)^{1}$ is placed on CWGxB/D when an auto-shutdown event is present								
	10 = A logic '0' is placed on CWGxB/D when an auto-shutdown event is present								
	01 = Pin is tri-stated on CWGxB/D when an auto-shutdown event is present 00 = The inactive state of the pin, including polarity, is placed on CWGxB/D after the required								
		active state of and interval	the pin, inclu	ding polarity, i	s placed on C	WGxB/D after	the require		
bit 3-2	LSAC<1:0>: CWGxA and CWGxC Auto-Shutdown State Control bits								
	11 = A logic '1' is placed on CWGxA/C when an auto-shutdown event is present								
	10 = A logic '0' is placed on CWGxA/C when an auto-shutdown event is present								
	01 = Pin is tri-stated on CWGxA/C when an auto-shutdown event is present 00 = The inactive state of the pin, including polarity, is placed on CWGxA/C after the required								
		and interval	the pin, inclu	ung polanty, r			the require		
bit 1-0	Unimplemer	nted: Read as	ʻ0'						
Note 1: This bi	-			IN register) to r	lace the outpu	its into the shut	down confi		
uration	•						down oonny		

REGISTER 28-5: CWGxAS0: CWGx AUTO-SHUTDOWN CONTROL REGISTER 0

2: The outputs will remain in auto-shutdown state until the next rising edge of the input signal after this bit is cleared.

U-1	R/W-0/0	R/W-0/0	R/W-0/0	U-1	R/W-0/0	R/W-0/0	R/W-0/0				
—	TMR6AS	TMR4AS	TMR2AS	—	C2AS ⁽¹⁾	C1AS	INAS				
bit 7							bit 0				
Logondi											
Legend: R = Readable bit		W = Writable bit		II = II nimplemented hit read as '0'							
		x = Bit is unknown		U = Unimplemented bit, read as '0' -n/n = Value at POR and BOR/Value at all other Resets							
u = Bit is unchanged		'0' = Bit is cleared		 n/n = value at POR and BOR/value at all other Resets q = Value depends on condition 							
'1' = Bit is se	l	0 = Bit is cle	areo	q = value de	penas on conait	ION					
bit 7	Unimplemented: Read as '1'										
bit 6	TMR6AS: TMR6 Postscale Output bit										
	 1 = TMR6 postscale shut-down is enabled 0 = TMR6 postscale shut-down is disabled 										
bit 5	TMR4AS: TMR4 Postscale Output bit										
	 1 = TMR4 postscale shut-down is enabled 0 = TMR4 postscale shut-down is disabled 										
bit 4	TMR2AS: TMR2 Postscale Output bit 1 = TMR2 postscale shut-down is enabled 0 = TMR2 postscale shut-down is disabled										
bit 3	•	Unimplemented: Read as '1'									
bit 2	C2AS: Comparator C2 Output bit										
	1 = C2 output shut-down is enabled 0 = C2 output shut-down is disabled										
bit 1	C1AS: Comparator C1 Output bit 1 = C1 output shut-down is enabled 0 = C1 output shut-down is disabled										
bit 0	INAS: CWGx Input Pin bit 1 = CWGxIN input pin shut-down is enabled 0 = CWGxIN input pin shut-down is disabled										

REGISTER 28-6: CWGxAS1: CWGx AUTO-SHUTDOWN CONTROL REGISTER 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	R/W-0/0				
OVRD	OVRC	OVRB	OVRA	STRD ⁽²⁾	STRC ⁽²⁾	STRB ⁽²⁾	STRA ⁽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		q = Value depends on condition							
bit 7	OVRD: Steer	OVRD: Steering Data D bit									
bit 6	OVRC: Steer	OVRC: Steering Data C bit									
bit 5	OVRB: Steer	OVRB: Steering Data B bit									
bit 4	OVRA: Steer	OVRA: Steering Data A bit									
bit 3	STRD: Steering Enable D bit ⁽²⁾										
	 1 = CWGxD output has the CWGx_data waveform with polarity control from POLD bit 0 = CWGxD output is assigned the value of OVRD bit 										
bit 2 bit 1	STRC: Steering Enable C bit ⁽²⁾										
	1 = CWGxC output has the CWGx_data waveform with polarity control from POLC bit										
		 0 = CWGxC output is assigned the value of OVRC bit STRB: Steering Enable B bit⁽²⁾ 									
		1 = CWGxB output has the CWGx data waveform with polarity control from POLB bit									
		0 = CWGxB output has the CWGx_data wavelorm with polarity control norm OLB bit									
bit 0	STRA: Steering Enable A bit ⁽²⁾										
	 1 = CWGxA output has the CWGx_data waveform with polarity control from POLA bit 0 = CWGxA output is assigned the value of OVRA bit 										

REGISTER 28-7: CWGxOCON0: CWGx STEERING CONTROL REGISTER 0⁽¹⁾

Note 1: The bits in this register apply only when MODE<2:0> = 00x.
2: This bit is effectively double-buffered when MODE<2:0> = 001.

REGISTER 28-8: CWGxCLKCON: CWGx CLOCK SELECTION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	CS
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 0

bit 3-0

CS: CWGx Clock Selection bit

1 = HFINTOSC 16 MHz is selected

0 = Fosc is selected

REGISTER 28-9: CWGxISM: CWGx INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	_		IS<3	3:0>	
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	q = Value depends on condition

bit 7-4 Unimplemented: Read as '0'

IS<3:0>: CWGx Input Selection bits

1111 = Reserved. No channel connected.

- •
- •
- 1011 = Reserved. No channel connected.
- 1010 = PWM4_out
- 1001 = PWM3_out
- 1000 = LC4_out
- 0111 = LC3_out
- 0110 = LC2_out
- 0101 = LC1_out
- 0100 = CCP2_out
- 0011 = CCP1_out
- 0010 = C2_OUT_sync
- 0001 = C1_OUT_sync
- 0000 = CWGxIN pin

TABLE 28-2:	SUMMARY OF REGISTERS ASSOCIATED WITH CWG
--------------------	--

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWG1AS0	SHUTDOWN	REN	LSBD	<1:0>	LSAC	<1:0>	—	_	386
CWG1AS1	_	TMR6AS	TMR4AS	TMR2AS	_	C2AS	C1AS	INAS	387
CWG1CLKCON	_	_	_	_	_	_	_	CS	389
CWG1CON0	EN	LD	—	— — — MODE<2:0>			388		
CWG1CON1	_	_	IN	_	POLD	POLC	POLB	POLA	384
CWG1DBF	_	_		DBF<5:0>				385	
CWG1DBR	_	_		DBR<5:0>					385
CWG1ISM	_	_	_	IS<3:0>			389		
CWG10C0N0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	388

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

29.0 SIGNAL MEASUREMENT TIMER (SMT)

The SMT is a 24-bit counter with advanced clock and gating logic, which can be configured for measuring a variety of digital signal parameters such as pulse width, frequency and duty cycle, and the time difference between edges on two signals.

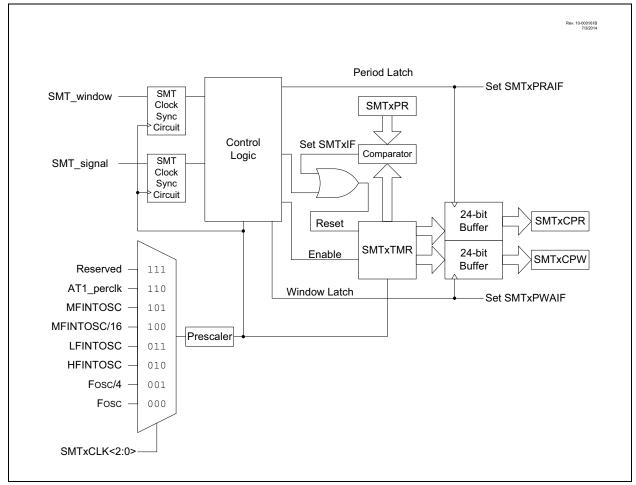
Features of the SMT include:

- · 24-bit timer/counter
 - Four 8-bit registers (SMTxTMRL/H/U)
 - Readable and writable
 - Optional 16-bit operating mode
- Two 24-bit measurement capture registers
- One 24-bit period match register
- Multi-mode operation, including relative timing measurement
- Interrupt on period match
- Multiple clock, gate and signal sources
- Interrupt on acquisition complete
- · Ability to read current input values

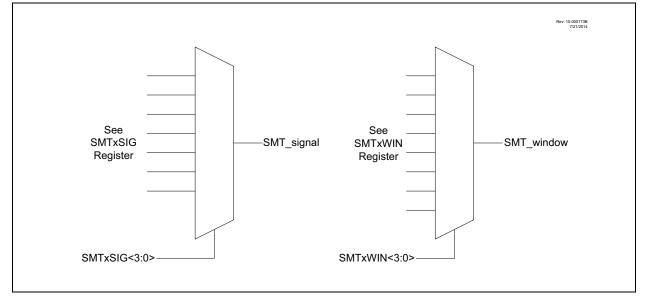
Note: These devices implement two SMT modules. All references to SMTx apply to SMT1 and SMT2.

PIC16(L)F1615/9









29.1 SMT Operation

The core of the module is the 24-bit counter, SMTxTMR combined with a complex data acquisition front-end. Depending on the mode of operation selected, the SMT can perform a variety of measurements summarized in Table 29-1.

29.1.1 CLOCK SOURCES

Clock sources available to the SMT include:

- Fosc
- Fosc/4
- HFINTOSC 16 MHz
- LFINTOSC
- MFINTOSC 31.25 kHz

The SMT clock source is selected by configuring the CSEL<2:0> bits in the SMTxCLK register. The clock source can also be prescaled using the PS<1:0> bits of the SMTxCON0 register. The prescaled clock source is used to clock both the counter and any synchronization logic used by the module.

29.1.2 PERIOD MATCH INTERRUPT

Similar to other timers, the SMT triggers an interrupt when SMTxTMR rolls over to '0'. This happens when SMTxTMR = SMTxPR, regardless of mode. Hence, in any mode that relies on an external signal or a window to reset the timer, proper operation requires that SMTxPR be set to a period larger than that of the expected signal or window.

29.2 Basic Timer Function Registers

The SMTxTMR time base and the SMTxCPW/SMTxPR/SMTxCPR buffer registers serve several functions and can be manually updated using software.

29.2.1 TIME BASE

The SMTxTMR is the 24-bit counter that is the center of the SMT. It is used as the basic counter/timer for measurement in each of the modes of the SMT. It can be reset to a value of 24'h00_0000 by setting the RST bit of the SMTxSTAT register. It can be written to and read from software, but it is not guarded for atomic access, therefore reads and writes to the SMTxTMR should only be made when the GO = 0, or the software should have other measures to ensure integrity of SMTxTMR reads/writes.

29.2.2 PULSE WIDTH LATCH REGISTERS

The SMTxCPW registers are the 24-bit SMT pulse width latch. They are used to latch in the value of the SMTxTMR when triggered by various signals, which are determined by the mode the SMT is currently in. The SMTxCPW registers can also be updated with the current value of the SMTxTMR value by setting the CPWUP bit of the SMTxSTAT register.

29.2.3 PERIOD LATCH REGISTERS

The SMTxCPR registers are the 24-bit SMT period latch. They are used to latch in other values of the SMTxTMR when triggered by various other signals, which are determined by the mode the SMT is currently in.

The SMTxCPR registers can also be updated with the current value of the SMTxTMR value by setting the CPRUP bit in the SMTxSTAT register.

29.3 Halt Operation

The counter can be prevented from rolling-over using the STP bit in the SMTxCON0 register. When halting is enabled, the period match interrupt persists until the SMTxTMR is reset (either by a manual reset, **Section29.2.1 "Time Base**") or by clearing the SMTxGO bit of the SMTxCON1 register and writing the SMTxTMR values in software.

29.4 Polarity Control

The three input signals for the SMT have polarity control to determine whether or not they are active high/positive edge or active low/negative edge signals.

The following bits apply to Polarity Control:

- WSEL bit (Window Polarity)
- SSEL bit (Signal Polarity)
- CSEL bit (Clock Polarity)

These bits are located in the SMTxCON0 register.

29.5 Status Information

The SMT provides input status information for the user without requiring the need to deal with the polarity of the incoming signals.

29.5.1 WINDOW STATUS

Window status is determined by the WS bit of the SMTxSTAT register. This bit is only used in Windowed Measure, Gated Counter and Gated Window Measure modes, and is only valid when TS = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

29.5.2 SIGNAL STATUS

Signal status is determined by the AS bit of the SMTxSTAT register. This bit is used in all modes except Window Measure, Time of Flight and Capture modes, and is only valid when TS = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

29.5.3 GO STATUS

Timer run status is determined by the TS bit of the SMTxSTAT register, and will be delayed in time by synchronizer delays in non-Counter modes.

29.6 Modes of Operation

The modes of operation are summarized in Table 29-1. The following sections provide detailed descriptions, examples of how the modes can be used. Note that all waveforms assume WPOL/SPOL/CPOL = 0. When WPOL/SPOL/CPOL = 1, all SMTSIGx, SMTWINx and SMT clock signals will have a polarity opposite to that indicated. For all modes, the REPEAT bit controls whether the acquisition is repeated or single. When REPEAT = 0 (Single Acquisition mode), the timer will stop incrementing and the SMTxGO bit will be reset upon the completion of an acquisition. Otherwise, the timer will continue and allow for continued acquisitions to overwrite the previous ones until the timer is stopped in software.

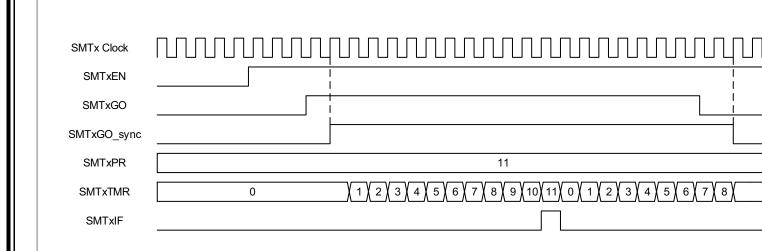
29.6.1 TIMER MODE

Timer mode is the simplest mode of operation where the SMTxTMR is used as a 16/24-bit timer. No data acquisition takes place in this mode. The timer increments as long as the SMTxGO bit has been set by software. No SMT window or SMT signal events affect the SMTxGO bit. Everything is synchronized to the SMT clock source. When the timer experiences a period match (SMTxTMR = SMTxPR), SMTxTMR is reset and the period match interrupt trips. See Figure 29-3.

MODE	Mode of Operation	Synchronous Operation	Reference
0000	Timer	Yes	Section29.6.1 "Timer Mode"
0001	Gated Timer	Yes	Section29.6.2 "Gated Timer Mode"
0010	0010 Period and Duty Cycle Acquisition		Section29.6.3 "Period and Duty-Cycle Mode"
0011	0011 High and Low Time Measurement		Section29.6.4 "High and Low Measure Mode"
0100	Windowed Measurement	Yes	Section29.6.5 "Windowed Measure Mode"
0101	Gated Windowed Measurement	Yes	Section29.6.6 "Gated Window Measure Mode"
0110	110 Time of Flight		Section29.6.7 "Time of Flight Measure Mode"
0111	Capture	Yes	Section29.6.8 "Capture Mode"
1000	Counter	No	Section29.6.9 "Counter Mode"
1001	Gated Counter	No	Section29.6.10 "Gated Counter Mode"
1010	Windowed Counter	No	Section29.6.11 "Windowed Counter Mode"
1011 - 1111	Reserved	—	_

TABLE 29-1: MODES OF OPERATION





29.6.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMTxTMR will increment. Upon a falling edge of the external signal, the SMTxCPW register will update to the current value of the SMTxTMR. Example waveforms for both repeated and single acquisitions are provided in Figure 29-4 and Figure 29-5.

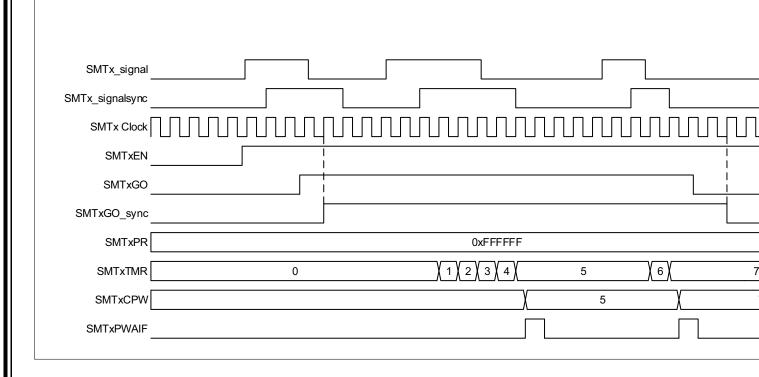
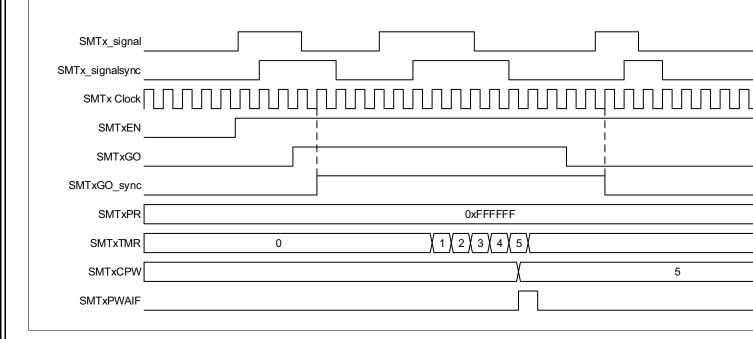


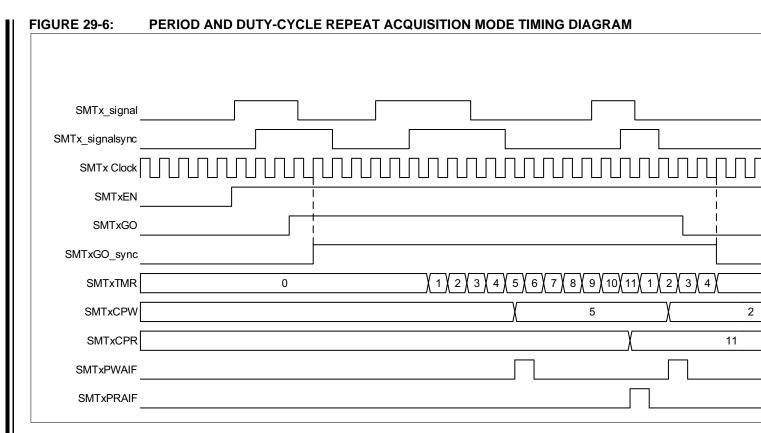
FIGURE 29-4: GATED TIMER MODE REPEAT ACQUISITION TIMING DIAGRAM





29.6.3 PERIOD AND DUTY-CYCLE MODE

In Duty-Cycle mode, either the duty cycle or period (depending on polarity) of the SMTx_signal can be acquired relative to the SMT clock. The CPW register is updated on a falling edge of the signal, and the CPR register is updated on a rising edge of the signal, along with the SMTxTMR resetting to 0x0001. In addition, the SMTxGO bit is reset on a rising edge when the SMT is in Single Acquisition mode. See Figure 29-6 and Figure 29-7.



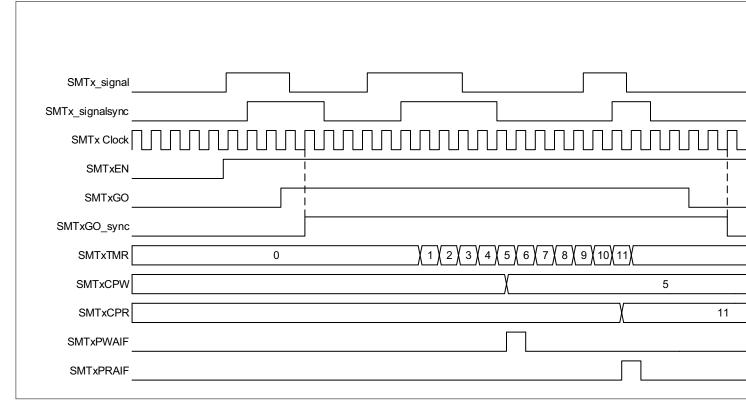


FIGURE 29-7: PERIOD AND DUTY-CYCLE SINGLE ACQUISITION TIMING DIAGRAM

© 2014-2017 Microchip Technology Inc.

29.6.4 HIGH AND LOW MEASURE MODE

This mode measures the high and low pulse time of the SMTSIGx relative to the SMT clock. It begins incrementing the SMTxTMR on a rising edge on the SMTSIGx input, then updates the SMTxCPW register with the value and resets the SMTxTMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMTxCPR register with its current value and once again resets the SMTxTMR value and begins incrementing again. See Figure 29-8 and Figure 29-9.

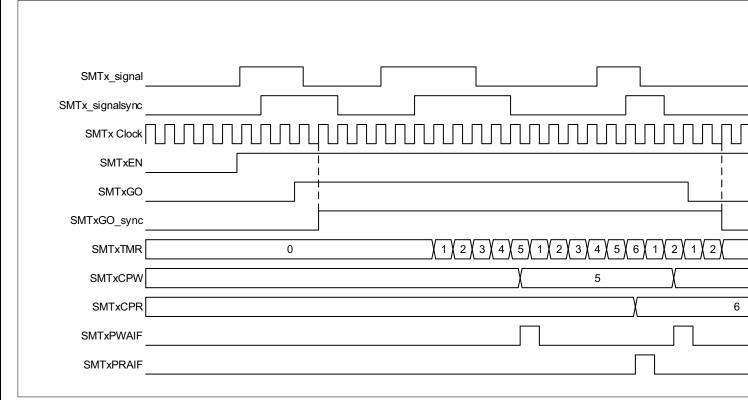
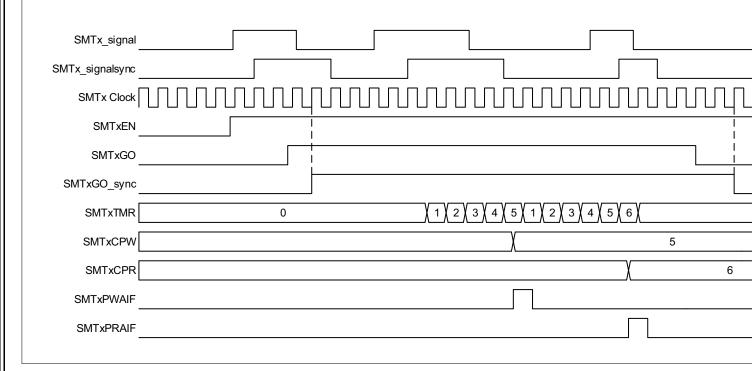


FIGURE 29-8: HIGH AND LOW MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

FIGURE 29-9: HIGH AND LOW MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM



29.6.5 WINDOWED MEASURE MODE

This mode measures the window duration of the SMTWINx input of the SMT. It begins incrementing the timer on a rising edge of the SMTWINx input and updates the SMTxCPR register with the value of the timer and resets the timer on a second rising edge. See Figure 29-10 and Figure 29-11.

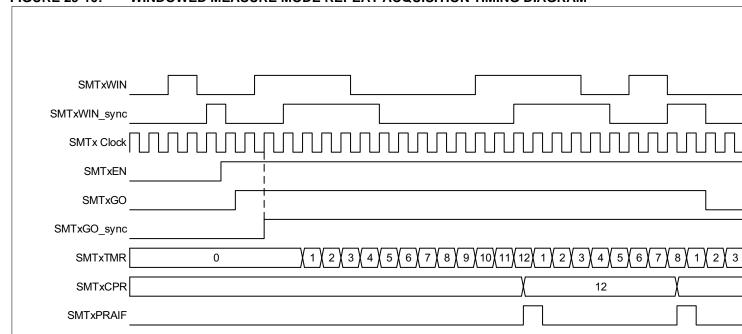
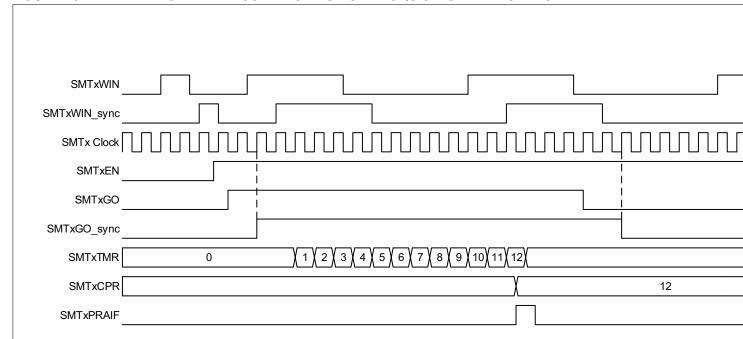


FIGURE 29-10: WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



| FIGURE 29-11: WINDOWED MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM

29.6.6 GATED WINDOW MEASURE MODE

This mode measures the duty cycle of the SMTx_signal input over a known input window. It does so by incrementing the timer on each pulse of the clock signal while the SMTx_signal input is high, updating the SMTxCPR register and resetting the timer on every rising edge of the SMTWINx input after the first. See Figure 29-12 and Figure 29-13.

SMTxWIN SMTxWIN_sync SMTx_signal SMTx_signalsync SMTx Clock SMTxEN SMTxGO SMTxGO_sync 1 2 0 (1) 2 3 4 \ 5 \ SMTxTMR 6 0 3 SMTxCPR 6 SMTxPRAIF

FIGURE 29-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

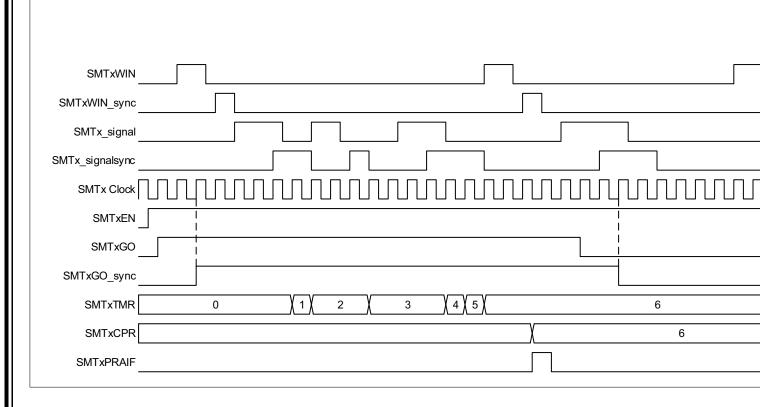


FIGURE 29-13: GATED WINDOWED MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAMS

29.6.7 TIME OF FLIGHT MEASURE MODE

This mode measures the time interval between a rising edge on the SMTWINx input and a rising edge on the SMTx_signal input, beginning to increment the timer upon observing a rising edge on the SMTWINx input, while updating the SMTxCPR register and resetting the timer upon observing a rising edge on the SMTx_signal input. In the event of two SMTWINx rising edges without an SMTx_signal rising edge, it will update the SMTxCPW register with the current value of the timer and reset the timer value. See Figure 29-14 and Figure 29-15.

SMTxWIN SMTxWIN_sync SMTx_signal SMTx_signalsync SMTx Clock SMTxEN SMTxGO SMTxGO_sync 1 2 3 4 $1 \left(2 \left(3 \right) \left(4 \right) \left(5 \right) \left(6 \right) \left(7 \right) \left(8 \right) \left(9 \right) \left(10 \right) \left(11 \right) \left(12 \right) \left$ SMTxTMR 0 5 SMTxCPW SMTxCPR 4 SMTxPWAIF

FIGURE 29-14: TIME OF FLIGHT MODE REPEAT ACQUISITION TIMING DIAGRAM

SMTxPRAIF

SMTxWIN SMTxWIN_sync SMTx_signal SMTx_signalsync SMTx Clock SMTxEN SMTxGO SMTxGO_sync 1 2 3 4 SMTxTMR 0 5 SMTxCPW SMTxCPR 4 SMTxPWAIF SMTxPRAIF

FIGURE 29-15: TIME OF FLIGHT MODE SINGLE ACQUISITION TIMING DIAGRAM

29.6.8 CAPTURE MODE

This mode captures the Timer value based on a rising or falling edge on the SMTWINx input and triggers an interrupt. This mimics the capture feature of a CCP module. The timer begins incrementing upon the SMTxGO bit being set, and updates the value of the SMTxCPR register on each rising edge of SMTWINx, and updates the value of the CPW register on each falling edge of the SMTWINx. The timer is not reset by any hardware conditions in this mode and must be reset by software, if desired. See Figure 29-16 and Figure 29-17.

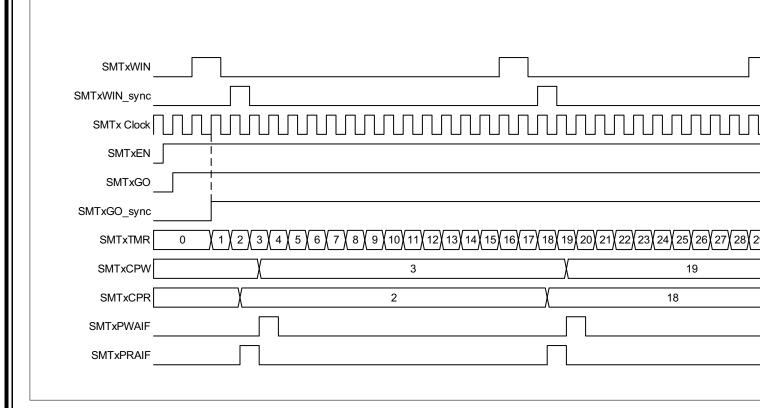


FIGURE 29-16: CAPTURE MODE REPEAT ACQUISITION TIMING DIAGRAM

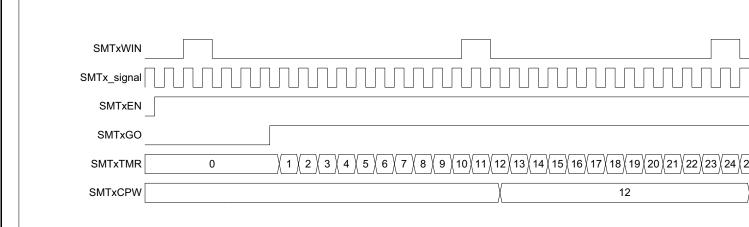
SMTxWIN SMTxWIN_sync SMTx Clock SMTxEN SMTxGO SMTxGO_sync 1 2 0 3 SMTxTMR SMTxCPW 3 2 SMTxCPR SMTxPWAIF SMTxPRAIF

FIGURE 29-17: CAPTURE MODE SINGLE ACQUISITION TIMING DIAGRAM

29.6.9 COUNTER MODE

This mode increments the timer on each pulse of the SMTx_signal input. This mode is asynchronous to the SMT clock and uses the SMTx_signal as a time source. The SMTxCPW register will be updated with the current SMTxTMR value on the falling edge of the SMTxWIN input. See Figure 29-18.

FIGURE 29-18: COUNTER MODE TIMING DIAGRAM



© 2014-2017 Microchip Technology Inc.

29.6.10 GATED COUNTER MODE

This mode counts pulses on the SMTx_signal input, gated by the SMTxWIN input. It begins incrementing the timer upon seeing a rising edge of the SMTxWIN input and updates the SMTxCPW register upon a falling edge on the SMTxWIN input. See Figure 29-19 and Figure 29-20.

FIGURE 29-19: GATED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM

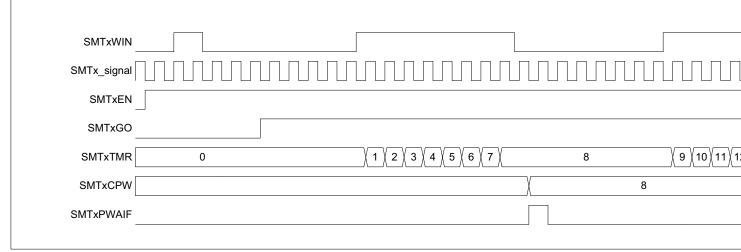
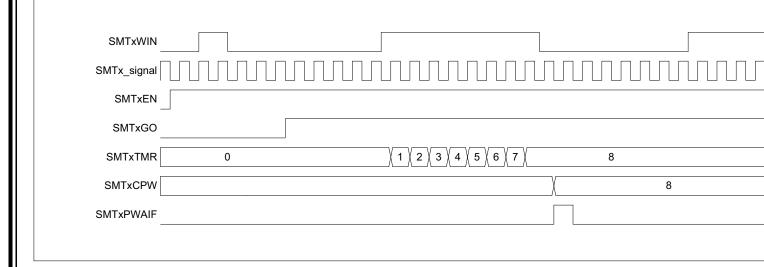


FIGURE 29-20: GATED COUNTER MODE SINGLE ACQUISITION TIMING DIAGRAM



29.6.11 WINDOWED COUNTER MODE

This mode counts pulses on the SMTx_signal input, within a window dictated by the SMTxWIN input. It begins counting upon seeing a rising edge of the SMTxWIN input, updates the SMTxCPW register on a falling edge of the SMTxWIN input, and updates the SMTxCPR register on each rising edge of the SMTxWIN input beyond the first. See Figure 29-21 and Figure 29-22.

FIGURE 29-21: WINDOWED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM

SMTxWIN				
SMTx_signal				
SMTxEN				
SMTxGO				
SMTxTMR	0	1 2 3 4	5 \ 6 \ 7 \ 8 \ 9 \10 \11	
SMTxCPW			χ	9
SMTxCPR				χ
SMTxPWAIF				
SMTxPRAIF				
FIGURE 29-22: W	INDOWED COUNTER MC	DDE SINGLE ACQUISIT	ION TIMING DIAGRA	М
CNAT- JA/INI				
SMTxWIN				
SMTxWIN SMTx_signal				
SMTx_signal				
SMTx_signal] 	∑5 \ 6 \ 7 \ 8 \ 9 \10\(1	 1\12\13\14\(15\(16\
SMTx_signal SMTxEN SMTxGO] 	↓ ↓ 5 \ 6 \ 7 \ 8 \ 9 \10\1 \	 1 <u>\12 \13 \14 \15 \16 \</u> 9
SMTx_signal SMTxEN SMTxGO SMTxTMR		 	<u>χ 5 χ 6 χ 7 χ 8 χ 9 χ 10 χ 1</u>	
SMTx_signal SMTxEN SMTxGO SMTxTMR SMTxCPW] 	∑5 ∑ 6 ∑ 7 ∑ 8 ∑ 9 ∑10 ∑1 ∑ 5 ∑ 6 ∑ 7 ∑ 8 ∑ 9 ∑10 ∑1	
SMTx_signal SMTxEN SMTxGO SMTxTMR SMTxCPW SMTxCPR			<u> </u>	

29.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

29.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMTxCPW and SMTxCPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMTxCPW interrupt is controlled by SMTxPWAIF and SMTxPWAIE bits in registers PIR4 and PIE4, respectively. The SMTxCPR interrupt is controlled by the SMTxPRAIF and SMTxPRAIF and SMTxPRAIF and PIE4, respectively.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMTxCLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

29.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in Section 29.1.2 "Period Match interrupt", the SMT will also interrupt upon SMTxTMR, matching SMTxPR with its period match limit functionality described in Section29.3 "Halt Operation". The period match interrupt is controlled by SMTxIF and SMTxIE, located in registers PIR4 and PIE4, respectively.

29.8 Register Definitions: SMT Control

Long bit name prefixes for the Signal Measurement Timer peripherals are shown in Table 29-2. Refer to Section 1.1 "Register and Bit Naming Conventions" for more information.

TABLE 29-2:

Peripheral	Bit Name Prefix	
SMT1	SMT1	
SMT2	SMT2	

REGISTER 29-1: SMTxCON0: SMT CONTROL REGISTER 0

R/W-0/0	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
EN ⁽¹⁾	—	STP	WPOL	SPOL	CPOL	SMTxP	S<1:0>
bit 7							bit 0

Legend:					
R = Readable bit		W = Writable bit	U = Unimplemented bit, read as '0'		
u = Bit is unchanged '1' = Bit is set		x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets		
		'0' = Bit is cleared			
bit 7	1 = SMT is		e reset, clock requests are disabled		
bit 6	Unimplem	ented: Read as '0'			
bit 5	5 STP: SMT Counter Halt Enable bit When SMTxTMR = SMTxPR:				
 1 = Counter remains SMTxPR; period match interrupt occurs when clocked 0 = Counter resets to 24'h000000; period match interrupt occurs when clocked 					
bit 4	1 = SMTxV	/TxWIN Input Polarity Contr VIN signal is active-low/fallin VIN signal is active-high/risi	ng edge enabled		
bit 3	SPOL: SMTxSIG Input Polarity Control bit 1 = SMTx_signal is active-low/falling edge enabled 0 = SMTx_signal is active-high/rising edge enabled				
bit 2	CPOL: SMT Clock Input Polarity Control bit 1 = SMTxTMR increments on the falling edge of the selected clock signal 0 = SMTxTMR increments on the rising edge of the selected clock signal				
bit 1-0	SMTxPS<1:0>: SMT Prescale Select bits 11 = Prescaler = 1:8 10 = Prescaler = 1:4 01 = Prescaler = 1:2 00 = Prescaler = 1:1				

Note 1: Setting EN to '0' does not affect the register contents.

REGISTER 29-2: SMTxCON1: SMT CONTROL REGISTER 1

R/W/HC-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	
SMTxGO	REPEAT		_		MODE	=<3:0>		
bit 7							bit (
Legend:								
HC = Bit is cle	ared by hardw	/are		HS = Bit is se	et by hardware			
R = Readable bit W = Writable bit			U = Unimpler	nented bit, rea	d as '0'			
u = Bit is unchanged x = Bit is unknown			-n/n = Value a	at POR and BC	OR/Value at all	other Resets		
'1' = Bit is set		'0' = Bit is clea	ared	q = Value dep	pends on condi	tion		
bit 6	0 = Incremer	nting, acquiring nting, acquiring /IT Repeat Acqu	data is disable	ed				
bit o	1 = Repeat D	Data Acquisition	mode is enab					
bit 5-4	Unimpleme	nted: Read as '	0'					
bit 3-0	MODE<3:0> SMT Operation Mode Select bits 1111 = Reserved							
	1001 = Gate 1000 = Cour 0111 = Capt 0110 = Time 0101 = Gate 0100 = Winc 0011 = High	dowed counter ed counter nter e of flight ed windowed me dowed measure and low time m od and Duty-Cyc ed Timer	easurement	1				

R/W/HC-0/0	R/W/HC-0/0	R/W/HC-0/0	U-0	U-0	R-0/0	R-0/0	R-0/0
CPRUP	CPWUP	RST			TS	WS	AS
bit 7							bit 0
Legend:							
HC = Bit is clea	-	are			et by hardware		
R = Readable	bit	W = Writable	bit	•	mented bit, read		
u = Bit is unch	anged	x = Bit is unkr	iown	-n/n = Value a	at POR and BC	R/Value at all	other Resets
'1' = Bit is set		'0' = Bit is clea	ared	q = Value dep	pends on condi	tion	
bit 7	1 = Request u 0 = SMTxCPF	Manual Perio update to SMT Rx registers up	CPRx register	rs te			
bit 6	1 = Request ເ	T Manual Pulse update to SMT N registers upo	CPW register	S			
bit 5	1 = Request F	anual Timer Re Reset to SMTx R registers upd	TMR registers	e			
bit 4-3	Unimplemen	ted: Read as '	כ'				
bit 2	1 = SMT time	Value Status b r is incrementir r is not increme	ng				
bit 1	WS: SMTxWI 1 = SMT wind 0 = SMT wind	•	bit				
bit 0	1 = SMT acqu	nal Value Statu uisition is in pro uisition is not in	gress				

REGISTER 29-3: SMTxSTAT: SMT STATUS REGISTER

Г

REGISTER 29-4:	SMTxCLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
0-0	0-0	0-0	0-0	0-0	R/W-0/U		R/W-0/0
	_		_		CSEL<2:0>		
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		U = Unimpler	nented bit, rea	d as '0'			
u = Bit is unchanged x = Bit is unknown			nown	-n/n = Value a	at POR and BC	R/Value at all	other Resets
'1' = Bit is se	1' = Bit is set '0' = Bit is cleared		ared	q = Value dep	ends on condi	tion	
bit 7-3	Unimplemen	ted: Read as '	0'				
bit 2-0	CSEL<2:0>:	SMT Clock Se	lection bits				
	111 = Reserv	ved					
	110 = AT1_p	erclk					
	101 = MFINT	OSC					
	100 = MFINT	OSC/16					
	011 = LFINT	OSC					
	010 = HFINT	OSC 16 MHz					
	001 = Fosc/4	1					

000 = Fosc

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0				
	_				WSEL<4:0>						
bit 7							bit				
Legend:											
R = Readal		W = Writab			mented bit, read						
u = Bit is unchanged x = Bit is unknown				at POR and BO		other Resets					
'1' = Bit is set '0' = Bit is cleared		cleared	q = Value de	pends on condi	tion						
bit 7-5	Unimpleme	nted: Read a	s '0'								
bit 4-0	-	/SEL<4:0>: SMT1 Window Selection bits									
	111111 = Re			5							
	•										
	•										
	• 11000 = Re	served									
		FINTOSC/16									
	10110 = AT										
	10101 = LF										
	10100 = PV										
	10011 = PV 10010 = SN										
	10010 – Si	_									
		1R0_overflow									
		1R5_overflow									
		IR3_overflow									
		IR1_overflow									
		01100 = LC4_out 01011 = LC3_out									
	01010 = LC										
		01001 = LC1_out									
		01000 = TMR6_postscaled									
		00111 = TMR4_postscaled									
		00110 = TMR2_postscaled 00101 = ZCD1 out									
	00100 = CC										
	00011 = CC										
	00010 = C2										
	00001 = C1										
	00000 = SN	nia XNIN II									

REGISTER 29-5: SMT1WIN: SMT1 WINDOW INPUT SELECT REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0				
_	—	_			WSEL<4:0>						
oit 7							bit				
Legend:	-1:-					L = = (O)					
R = Readal		W = Writable			mented bit, read						
u = Bit is unchanged x = Bit is unknown				at POR and BO		other Resets					
'1' = Bit is s	et	'0' = Bit is cle	eared	q = Value de	pends on condit	ion					
bit 7-5	Unimpleme	nted [.] Read as	ʻ∩'								
bit 4-0	-	implemented: Read as '0' SEL<4:0>: SMT2 Window Selection bits									
Dit 4 -0	11111 = Res		W Selection bits	2							
	•										
	•										
	•										
	11000 = Res 10111 = MF										
	$10110 = AT^{2}$										
	10101 = LFI										
	10100 = PW	_									
	10011 = PW										
	10010 = Res 10001 = SM										
		R0 overflow									
	01111 = TM										
	01110 = TM	MR3_overflow									
	01101 = TM	_									
	$01100 = LC_{4}$										
		01011 = LC3_out									
		01010 = LC2_out 01001 = LC1_out									
		01000 = TMR6_postscaled									
		00111 = TMR4_postscaled									
		00110 = TMR2_postscaled									
	00101 = ZC 00100 = CC										
	00100 = CC 00011 = CC										
	00011 = CC										
	00001 = C10	OUT_sync									
	00000 = SM										

REGISTER 29-6: SMT2WIN: SMT2 WINDOW INPUT SELECT REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	_	_			SSEL<4:0>		
bit 7							bit
Legend:							
R = Readal	ole bit	W = Writable	e bit	U = Unimpler	nented bit, read	d as '0'	
u = Bit is unchanged x = Bit is unl		nown	-n/n = Value a	at POR and BC	R/Value at all	other Resets	
'1' = Bit is s	et	'0' = Bit is cle	eared	q = Value dep	pends on condi	tion	
bit 7-5	Unimpleme	nted: Read as	' ∩'				
bit 4-0	-	: SMT1 Signal					
	11111 = Re						
	•						
	•						
	10101 = R e	served					
	10100 = PV						
	10011 = PV						
	10010 = CC 10001 = CC						
		1R0 overflow					
	01111 = SN						
	01110 = R e						
		IR5_overflow					
		1R3_overflow					
	01011 - 10 01010 = LC	IR1_overflow					
	01001 = LC						
	01000 = LC						
	00111 = LC						
		IR6_postscaled					
		1R4_postscaled					
	00100 = TK 00011 = ZC	IR2_postscaled	1				
	00011 = 20 00010 = C2						
	00001 = C1	OUT_sync					
	00000 = SN						

REGISTER 29-7: SMT1SIG: SMT1 SIGNAL INPUT SELECT REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
_	_	_			SSEL<4:0>		
bit 7		·					bit
Legend:							
R = Readable bit W = Writable bit		bit	U = Unimpler	mented bit, read	d as '0'		
u = Bit is unchanged		x = Bit is unk	nown	-n/n = Value a	at POR and BC	R/Value at all	other Resets
'1' = Bit is s	et	'0' = Bit is cle	ared	q = Value dep	pends on condi	tion	
bit 7-5	Unimpleme	nted: Read as	0'				
bit 4-0	-	SMT2 Signal S					
	11111 = Re	-					
	•						
	•						
	10101 = Re	served					
	10100 = PW	_					
	10011 = PW						
	10010 = CC 10001 = CC						
		IR0 overflow					
	01111 = Re						
	01110 = SN						
		IR5_overflow IR3_overflow					
		IR1 overflow					
	01010 = LC						
	01001 = LC						
	01000 = LC						
	00111 = LC 00110 = TM	I_out IR6_postscaled					
		IR4_postscaled					
	00100 = TM	IR2_postscaled					
	00011 = ZC						
	00010 = C2 00001 = C1						
	000001 = CT	COI_Sync					

REGISTER 29-8: SMT2SIG: SMT2 SIGNAL INPUT SELECT 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	
			SMTxT	MR<7:0>				
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 unknown		-n/n = Value at POR and BOR/Value at all other Res			other Resets	
'1' = Bit is set		'0' = Bit is clea	ared					

REGISTER 29-9: SMTxTMRL: SMT TIMER REGISTER – LOW BYTE

bit 7-0 SMTxTMR<7:0>: Significant bits of the SMT Counter – Low Byte

REGISTER 29-10: SMTxTMRH: SMT TIMER REGISTER - HIGH BYTE

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		
SMTxTMR<15:8>									
bit 7 bit 0									

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

bit 7-0 SMTxTMR<15:8>: Significant bits of the SMT Counter – High Byte

REGISTER 29-11: SMTxTMRU: SMT TIMER REGISTER - UPPER BYTE

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
			SMTxTM	R<23:16>			
bit 7							bit 0
Legend:							
R = Readable	hit	W = Writable bi	ŧ	= Inimpler	nented hit read	1 as '0'	

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 SMTxTMR<23:16>: Significant bits of the SMT Counter – Upper Byte

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x		
			SMTx0	CPR<7:0>					
bit 7							bit 0		
Legend:									
R = Readable b	oit	W = Writable bit		U = Unimplemented bit, read as '0'					
u = Bit is uncha	anged	x = Bit is unknow	'n	-n/n = Value at POR and BOR/Value at all other Re			other Resets		
'1' = Bit is set		'0' = Bit is cleare	d						

REGISTER 29-12: SMTxCPRL: SMT CAPTURED PERIOD REGISTER – LOW BYTE

bit 7-0 SMTxCPR<7:0>: Significant bits of the SMT Period Latch – Low Byte

REGISTER 29-13: SMTxCPRH: SMT CAPTURED PERIOD REGISTER - HIGH BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	
SMTxCPR<15:8>								
bit 7							bit 0	

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

bit 7-0 SMTxCPR<15:8>: Significant bits of the SMT Period Latch – High Byte

REGISTER 29-14: SMTxCPRU: SMT CAPTURED PERIOD REGISTER – UPPER BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
			SMTxCPF	R<23:16>			
bit 7							bit 0
Legend.							

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 SMTxCPR<23:16>: Significant bits of the SMT Period Latch – Upper Byte

REGISTER 29-15: SMTxCPWL: SMT CAPTURED PULSE WIDTH REGISTER – LOW BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	
			SMTxC	PW<7:0>				
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable bit	t	U = Unimplemented bit, read as '0'				
u = Bit is uncha	anged	x = Bit is unknow	wn	-n/n = Value a	at POR and BC	R/Value at all o	other Resets	
'1' = Bit is set		'0' = Bit is cleare	ed					

bit 7-0 SMTxCPW<7:0>: Significant bits of the SMT PW Latch – Low Byte

REGISTER 29-16: SMTxCPWH: SMT CAPTURED PULSE WIDTH REGISTER - HIGH BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	
SMTxCPW<15:8>								
bit 7							bit 0	

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

bit 7-0 SMTxCPW<15:8>: Significant bits of the SMT PW Latch – High Byte

REGISTER 29-17: SMTxCPWU: SMT CAPTURED PULSE WIDTH REGISTER - UPPER BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
			SMTxCPV	V<23:16>			
bit 7							bit 0
Lawrende							

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 SMTxCPW<23:16>: Significant bits of the SMT PW Latch – Upper Byte

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
			SMTxF	PR<7:0>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	it	U = Unimpler	nented bit, read	d as '0'	
u = Bit is unch	anged	x = Bit is unkno	own	-n/n = Value at POR and BOR/Value at all other Resets			
'1' = Bit is set		'0' = Bit is clear	red				

REGISTER 29-18: SMTxPRL: SMT PERIOD REGISTER - LOW BYTE

bit 7-0 SMTxPR<7:0>: Significant bits of the SMT Timer Value for Period Match – Low Byte

REGISTER 29-19: SMTxPRH: SMT PERIOD REGISTER - HIGH BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<15:8>							
bit 7 bit 0							

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

bit 7-0 SMTxPR<15:8>: Significant bits of the SMT Timer Value for Period Match – High Byte

REGISTER 29-20: SMTxPRU: SMT PERIOD REGISTER – UPPER BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
			SMTxPF	R<23:16>			
bit 7							bit 0
Legend:							
R = Readable b	it	W = Writable bit		U = Unimpler	mented bit. read	d as '0'	

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 SMTxPR<23:16>: Significant bits of the SMT Timer Value for Period Match – Upper Byte

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	101
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	106
SMT1CON0	EN	_	STP	WPOL	SPOL	CPOL	SMT1P	S<1:0>	424
SMT1CON1	SMT1GO	REPEAT	_	_		MODE	<3:0>		425
SMT1CPRH				SMT1CP	R<15:8>				433
SMT1CPRL				SMT1CF	PR<7:0>				433
SMT1CPRU		SMT1CPR<23:16>							433
SMT1CPWH				SMT1CP	W<15:8>				434
SMT1CPWL				SMT1CF	PW<7:0>				434
SMT1CPWU				SMT1CP\	V<23:16>				434
SMT1PRH				SMT1PF	R<15:8>				435
SMT1PRL				SMT1P	R<7:0>				435
SMT1PRU				SMT1PF	<23:16>				435
SMT1SIG	_	_	_			SSEL<4:0>			430
SMT1STAT	CPRUP	CPWUP	RST	_	_	TS	WS	AS	426
SMT1TMRH	SMT1TMR<15:8>							432	
SMT1TMRL	SMT1TMR<7:0>						432		
SMT1TMRU	SMT1TMR<23:16>						432		
SMT1WIN	_	_	_			WSEL<4:0>			428
SMT2CLK	_	_	_	_	_		CSEL<2:0>		427
SMT2CON0	EN	_	STP	WPOL	SPOL	CPOL	SMT2P	'S<1:0>	424
SMT2CON1	SMT2GO	REPEAT	_	_		MODE	<3:0>		425
SMT2CPRH				SMT2CP	R<15:8>				433
SMT2CPRL	SMT2CPR<7:0>						433		
SMT2CPRU				SMT2CPI	R<23:16>				433
SMT2CPWH				SMT2CP	W<15:8>				434
SMT2CPWL				SMT2CF	PW<7:0>				434
SMT2CPWU				SMT2CP\	V<23:16>				434
SMT2PRH				SMT2PF	R<15:8>				435
SMT2PRL				SMT2P	R<7:0>				435
SMT2PRU				SMT2PF	<23:16>				435
SMT2SIG	SSEL<2:0>					430			
SMT2STAT	CPRUP	CPWUP	RST	_	_	TS	WS	AS	426
SMT2TMRH			•	SMT2TM	R<15:8>				432
SMT2TMRL				SMT2TN	/R<7:0>				432
SMT2TMRU				SMT2TM	R<23:16>				432
SMT2WIN	_	_				WSEL<4:0>			427

TABLE 29-3: SUMMARY OF REGISTERS ASSOCIATED WITH SMTx

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

30.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLCx) provides programmable logic that operates outside the speed limitations of software execution. The logic cell takes up to 16 input signals, and through the use of configurable gates, reduces the 16 inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

- · I/O pins
- Internal clocks
- · Peripherals
- · Register bits

The output can be directed internally to peripherals and to an output pin.

Refer to Figure 30-1 for a simplified diagram showing signal flow through the CLCx.

Possible configurations include:

- Combinatorial Logic
 - AND
 - NAND
 - AND-OR
 - AND-OR-INVERT
 - OR-XOR
 - OR-XNOR
- Latches
 - S-R
 - Clocked D with Set and Reset
 - Transparent D with Set and Reset
 - Clocked J-K with Reset

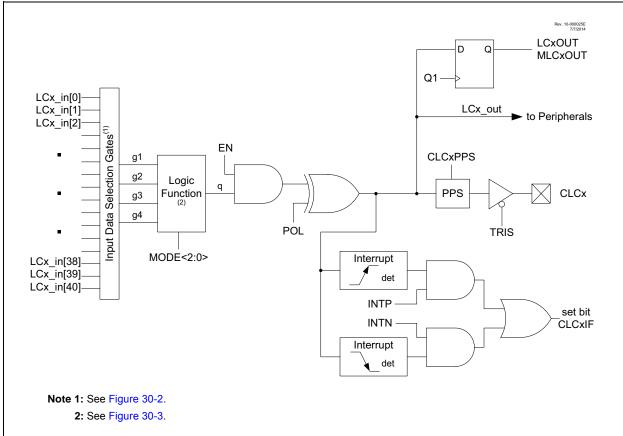


FIGURE 30-1: CONFIGURABLE LOGIC CELL BLOCK DIAGRAM

CLCx DATA INPUT

TABLE 30-1:

30.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- · Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

30.1.1 DATA SELECTION

There are 41 signals available as inputs to the configurable logic. Four 41 input multiplexers are used to select the inputs to pass on to the next stage. This allows for any of the possible input signals to be used as any of the four inputs to the CLC module.

Data selection is through four multiplexers as indicated on the left side of Figure 30-2. Data inputs in the figure are identified by a generic numbered input name.

Table 30-1 correlates the generic input name to the actual signal for each CLC module. The column labeled CLCxSELy refers to the value of any of the four registers associated with the four multiplexers, CLCxSEL0 through CLCxSEL3.

Data inputs for each multiplexer are selected with their respective CLCxSELy registers.

TABLE 30-1:	CLCX DATA INPUT SELECTION				
Data Input	CLCxSELy	CLC Input Signal			
LCx_in[0]	000000	CLCIN0			
LCx_in[1]	000001	CLCIN1			
LCx_in[2]	000010	CLCIN2			
LCx_in[3]	000011	CLCIN3			
LCx_in[4]	000100	LC1_out			
LCx_in[5]	000101	LC2_out			
LCx_in[6]	000110	LC3_out			
LCx_in[7]	000111	LC4_out			
LCx_in[8]	001000	C1OUT_sync			
LCx_in[9]	001001	C2OUT_sync			
LCx_in[10]	001010	CWGOUTA			
LCx_in[11]	001011	CWGOUTB			
LCx_in[12]	001100	CCP1_out			
LCx_in[13]	001101	CCP2_out			
LCx_in[14]	001110	PWM3_out			
LCx_in[15]	001111	PWM4_out			
LCx_in[16]	010000	AT1_cmp1			
LCx_in[17]	010001	AT1_cmp2			
LCx_in[18]	010010	AT1_cmp3			
LCx_in[19]	010011	SMT1_match			
LCx_in[20]	010100	SMT2_match			
LCx_in[21]	010101	ZCD1_output			
LCx_in[22]	010110	TMR0_overflow			
LCx_in[23]	010111	TMR1_overflow			
LCx_in[24]	011000	TMR2_postscaled			
LCx_in[25]	011001	TMR3_overflow			
LCx_in[26]	011010	TMR4_postscaled			
LCx_in[27]	011011	TMR5_overflow			
LCx_in[28]	011100	TMR6_postscaled			
LCx_in[29]	011101	IOC_interrupt			
LCx_in[30]	011110	ADC_rc			
LCx_in[31]	011111	LFINTOSC			
LCx_in[32]	100000	HFINTOSC			
LCx_in[33]	100001	FOSC			
LCx_in[34]	100010	AT1_missedpulse			
LCx_in[35]	100011	AT1_perclk			
LCx_in[36]	100100	AT1_phsclk			
LCx_in[37]	100101	TX			
LCx_in[38]	100110	RX			
LCx_in[39]	100111	SCK			
LCx_in[40]	101000	SDO			

30.2 Data Gating

Outputs from the input multiplexers are directed to the desired logic function input through the data gating stage. Each data gate can direct any combination of the four selected inputs.

Note: Data gating is undefined at power-up.

The gate stage is more than just signal direction. The gate can be configured to direct each input signal as inverted or non-inverted data. Directed signals are ANDed together in each gate. The output of each gate can be inverted before going on to the logic function stage.

The gating is in essence a 1-to-4 input AND/NAND/OR/ NOR gate. When every input is inverted and the output is inverted, the gate is an OR of all enabled data inputs. When the inputs and output are not inverted, the gate is an AND or all enabled inputs.

Table 30-2 summarizes the basic logic that can be obtained in gate 1 by using the gate logic select bits. The table shows the logic of four input variables, but each gate can be configured to use less than four. If no inputs are selected, the output will be zero or one, depending on the gate output polarity bit.

TABLE 30-2: DATA GATING LOGIC

CLCxGLS0	LCxG1POL	Gate Logic
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	OR
0x00	0	Logic 0
0x00	1	Logic 1

It is possible (but not recommended) to select both the true and negated values of an input. When this is done, the gate output is zero, regardless of the other inputs, but may emit logic glitches (transient-induced pulses). If the output of the channel must be zero or one, the recommended method is to set all gate bits to zero and use the gate polarity bit to set the desired level.

Data gating is configured with the logic gate select registers as follows:

- Gate 1: CLCxGLS0 (Register 30-6)
- Gate 2: CLCxGLS1 (Register 30-7)
- Gate 3: CLCxGLS2 (Register 30-8)
- Gate 4: CLCxGLS3 (Register 30-9)

Register number suffixes are different than the gate numbers because other variations of this module have multiple gate selections in the same register. Data gating is indicated in the right side of Figure 30-2. Only one gate is shown in detail. The remaining three gates are configured identically with the exception that the data enables correspond to the enables for that gate.

30.2.1 LOGIC FUNCTION

There are eight available logic functions including:

- AND-OR
- OR-XOR
- AND
- S-R Latch
- D Flip-Flop with Set and Reset
- D Flip-Flop with Reset
- J-K Flip-Flop with Reset
- · Transparent Latch with Set and Reset

Logic functions are shown in Figure 30-3. Each logic function has four inputs and one output. The four inputs are the four data gate outputs of the previous stage. The output is fed to the inversion stage and from there to other peripherals, an output pin, and back to the CLCx itself.

30.2.2 OUTPUT POLARITY

The last stage in the configurable logic cell is the output polarity. Setting the LCxPOL bit of the CLCxCON register inverts the output signal from the logic stage. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

30.2.3 CLCx SETUP STEPS

The following steps should be followed when setting up the CLCx:

- Disable CLCx by clearing the LCxEN bit.
- Select desired inputs using CLCxSEL0, CLCxSEL1, CLCxSEL2 and CLCxSEL3 registers (See Table 30-1).
- · Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- · Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the LCxPOLy bits of the CLCxPOL register.
- Select the desired logic function with the LCxMODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the LCxPOL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving a device, set the desired pin PPS control register and also clear the TRIS bit corresponding to that output.
- If interrupts are desired, configure the following bits:
 - Set the LCxINTP bit in the CLCxCON register for rising event.
 - Set the LCxINTN bit in the CLCxCON register or falling event.
 - Set the CLCxIE bit of the associated PIE registers.
 - Set the GIE and PEIE bits of the INTCON register.
- Enable the CLCx by setting the LCxEN bit of the CLCxCON register.

30.3 CLCx Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR registers will be set when either edge detector is triggered and its associated enable bit is set. The LCxINTP enables rising edge interrupts and the LCxINTN bit enables falling edge interrupts. Both are located in the CLCxCON register.

To fully enable the interrupt, set the following bits:

- LCxON bit of the CLCxCON register
- · CLCxIE bit of the associated PIE registers
- LCxINTP bit of the CLCxCON register (for a rising edge detection)
- LCxINTN bit of the CLCxCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The CLCxIF bit of the associated PIR registers, must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

30.4 Output Mirror Copies

Mirror copies of all LCxCON output bits are contained in the CLCxDATA register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the CLCxOUT bits in the individual CLCxCON registers.

30.5 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

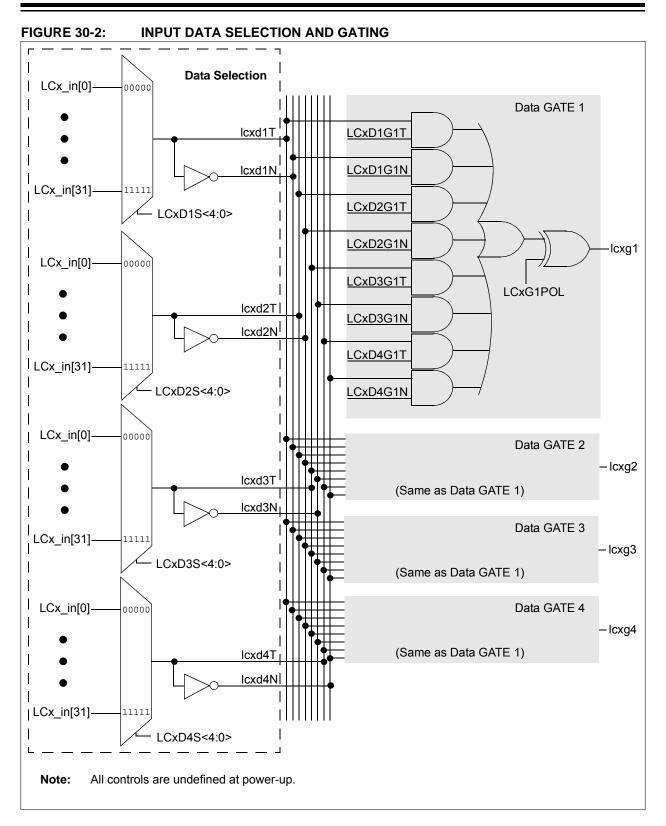
30.6 Operation During Sleep

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

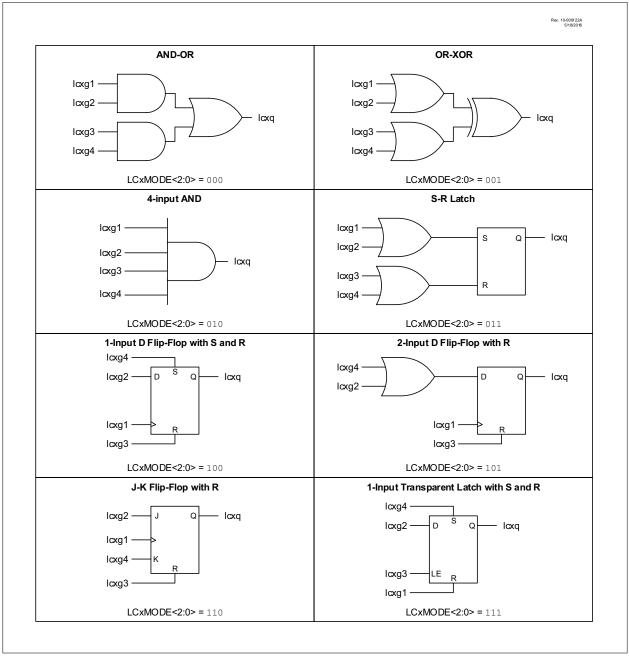
The HFINTOSC remains active during Sleep when the CLC module is enabled and the HFINTOSC is selected as an input source, regardless of the system clock source selected.

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

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







30.7 Register Definitions: CLC Control

R/W-0/0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN		LCxOUT	LCxINTP	LCxINTN	L	CxMODE<2:0>	•
bit 7							bit (
Legend:							
R = Readab	le bit	W = Writable	bit	U = Unimpler	nented bit, read	d as '0'	
u = Bit is und	changed	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all o	ther Resets
'1' = Bit is se	et	'0' = Bit is clea	ared				
bit 7		igurable Logic					
		able logic cell is					
bit 6	-	able logic cell is t ed: Read as '		Thas logic zero	σοιραί		
bit 5	•			staut bit			
		nfigurable Logi aic cell output (•	from lcx out v	vire	
bit 4	-	•		•	nterrupt Enable		
		will be set wher		• •	•		
	0 = CLCxIF v		. a nonig ouge		_001		
bit 3	LCxINTN: Co	onfigurable Log	ic Cell Negativ	ve Edge Going	Interrupt Enab	le bit	
		will be set wher	n a falling edge	e occurs on lcx	_out		
	0 = CLCxIF v						
bit 2-0		:0>: Configura	-		de bits		
		1-input transpa J-K flip-flop wi		h S and R			
		2-input D flip-f					
		1-input D flip-f		IR			
	011 = Cell is	• •	•				
	010 = Cell is	4-input AND					
	001 = Cell is						
	000 = Cell is	and-or					

REGISTER 30-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL	
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable	bit	U = Unimpler	mented 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 o	other Resets	
'1' = Bit is set		'0' = Bit is clea	ared					
bit 7	LCxPOL: LCO	OUT Polarity C	ontrol bit					
	•	ut of the logic o						
		ut of the logic c		erted				
bit 6-4	Unimplemen	ted: Read as 'o)'					
bit 3	LCxG4POL:	Gate 4 Output I	Polarity Cont	rol bit				
		0		n applied to the	logic cell			
		ut of gate 4 is r						
bit 2		Gate 3 Output I						
		•		n applied to the	logic cell			
bit 1	0 = The output of gate 3 is not inverted							
DIL	LCxG2POL: Gate 2 Output Polarity Control bit 1 = The output of gate 2 is inverted when applied to the logic cell							
	0 = The output of gate 2 is not inverted							
bit 0	LCxG1POL: Gate 1 Output Polarity Control bit							
		1 = The output of gate 1 is inverted when applied to the logic cell						
		= The output of gate 1 is not inverted						

REGISTER 30-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

REGISTER 30-3: CLCxSEL0: MULTIPLEXER DATA 0 SELECT REGISTERS

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
—	_			LCxD ²	1S<5:0>		
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
 Unimplemented: Read as '0'

 bit 5-0
 LCxD1S<5:0>: Input Data 1 Selection Control bits

 See Table 30-1 for signal names associated with inputs.

REGISTER 30-4: CLCxSEL1: MULTIPLEXER DATA 1 SELECT REGISTERS

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		
_	—	LCxD2S<5:0>							
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 Unimplemented: Read as '0'

bit 5-0 LCxD2S<5:0>: Input Data 2 Selection Control bits

See Table 30-1 for signal names associated with inputs.

REGISTER 30-5: CLCxSEL2: MULTIPLEXER DATA 2 SELECT REGISTERS

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		
—	—	LCxD3S<5:0>							
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	

0'
(

bit 5-0 LCxD3S<5:0>: Input Data 3 Selection Control bits See Table 30-1 for signal names associated with inputs.

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			
—	—		LCxD4S<5:0>							
bit 7	•						bit 0			
Legend:										
R = Readable b	R = Readable bit W = Writable bit			U = Unimplemented bit, read as '0'						
u = Bit is uncha	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 7-6	Unimplemented: Read as '0'
bit 5-0	LCxD4S<5:0>: Input Data 4 Selection Control bits
	See Table 30-1 for signal names associated with inputs.

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
LCxG1D4T	LCxG1D4N	LCxG1D3T	LCxG1D3N	LCxG1D2T	LCxG1D2N	LCxG1D1T	LCxG1D1N
bit 7							bit 0
Legend:							
R = Readable		W = Writable	bit	•	mented bit, read		
u = Bit is uncha	anged	x = Bit is unkr		-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		Gate 1 Data 4 T		rted) bit			
		not gated into					
bit 6	LCxG1D4N:	Gate 1 Data 4 I	Negated (inver	rted) bit			
		gated into Icxo not gated into					
bit 5	LCxG1D3T: 0	Gate 1 Data 3 T	rue (non-inve	rted) bit			
		gated into lcxg not gated into					
bit 4	1 = Icxd3N is	Gate 1 Data 3 I gated into Icxo not gated into	g1	rted) bit			
bit 3	LCxG1D2T: 0 1 = lcxd2T is	Gate 1 Data 2 T gated into lcxg	rue (non-inve 1	rted) bit			
bit 2		not gated into Gate 1 Data 2 I	•	tod) bit			
DIL Z	1 = Icxd2N is	gated into lcxg	g1	ited) bit			
bit 1	1 = Icxd1T is	Bate 1 Data 1 T gated into lcxg not gated into	j1	rted) bit			
bit 0		Gate 1 Data 1 I	•	rted) bit			
	1 = Icxd1N is	gated into lcxg not gated into	g1				

REGISTER 30-7: CLCxGLS0: GATE 1 LOGIC SELECT 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
LCxG2D4T	LCxG2D4N	LCxG2D3T	LCxG2D3N	LCxG2D2T	LCxG2D2N	LCxG2D1T	LCxG2D1N
bit 7							bit (
Legend:							
R = Readable		W = Writable		•	nented bit, read		
u = Bit is unch	anged	x = Bit is unkr		-n/n = Value a	at POR and BO	R/Value at all o	ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				
hit 7		Note 2 Data 4 7	True (non inve	rtad) bit			
bit 7		Gate 2 Data 4 7 gated into lcxc		ited) bit			
		not gated into					
bit 6	LCxG2D4N:	Gate 2 Data 4 I	Negated (inve	rted) bit			
		gated into Icx					
		not gated into	•				
bit 5		Gate 2 Data 3 1	,	rted) bit			
		gated into lcxg not gated into					
bit 4		Gate 2 Data 3	•	rted) bit			
		gated into lcx	•				
		not gated into					
bit 3	LCxG2D2T: (Gate 2 Data 2 1	rue (non-inve	rted) bit			
		gated into lcxg					
bit 2		not gated into Gate 2 Data 2 I	•	rtad) bit			
		gated into lcx	•	(led) bit			
		not gated into					
bit 1	LCxG2D1T: (Gate 2 Data 1 1	rue (non-inve	rted) bit			
		gated into lcxg					
		not gated into	•				
bit 0		Gate 2 Data 1 I	÷ .	rted) bit			
		gated into lcxg not gated into					

REGISTER 30-8: CLCxGLS1: GATE 2 LOGIC SELECT 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
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit		nented bit, read		
u = Bit is uncha	anged	x = Bit is unkr		-n/n = Value a	at POR and BO	R/Value at all c	ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7		Gate 3 Data 4 1	Truc (non invo	rtod) bit			
		gated into lcxc	•	iteu) bit			
		not gated into					
bit 6	LCxG3D4N:	Gate 3 Data 4	Negated (inver	rted) bit			
		gated into Icx					
		not gated into	•				
bit 5		Gate 3 Data 3 1	•	rted) bit			
		gated into lcxg not gated into					
bit 4		Gate 3 Data 3 I	•	rted) bit			
		gated into lcxg	•	···· , · ·			
	0 = Icxd3N is	not gated into	lcxg3				
bit 3		Gate 3 Data 2 1		rted) bit			
		gated into lcxg not gated into					
bit 2		Gate 3 Data 2	•	rted) hit			
5112		gated into Icx	•				
		not gated into					
bit 1	LCxG3D1T: (Gate 3 Data 1 1	rue (non-invei	rted) bit			
		gated into lcxg					
		not gated into	•				
bit 0		Gate 3 Data 1	•	rted) bit			
		gated into lcxo not gated into					
			- 3-				

REGISTER 30-9: CLCxGLS2: GATE 3 LOGIC SELECT 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
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 Rese '1' = Bit is set '0' = Bit is cleared bit 7 LCxG4D4T: Gate 4 Data 4 True (non-inverted) bit 1 = lcxd4T is gated into lcxg4 0 = lcxd4T is not gated into lcxg4 0 = lcxd4T is not gated into lcxg4 0 = lcxd4T is not gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into	LCxG4D4T	LCxG4D4N	LCxG4D3T	LCxG4D3N	LCxG4D2T	LCxG4D2N	LCxG4D1T	LCxG4D1N
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 Rese '1' = Bit is set '0' = Bit is cleared bit 7 LCxG4D4T: Gate 4 Data 4 True (non-inverted) bit 1 = loxd4T is gated into loxg4 0 = loxd4T is not gated into loxg4 0 = loxd4T is not gated into loxg4 0 = loxd4T is gated into loxg4 0 = loxd4N is gated into loxg4 0 = loxd4N is not gated into loxg4 0 = loxd4T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is n	bit 7							bit C
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 Rese '1' = Bit is set '0' = Bit is cleared bit 7 LCxG4D4T: Gate 4 Data 4 True (non-inverted) bit 1 = loxd4T is gated into loxg4 0 = loxd4T is not gated into loxg4 0 = loxd4T is not gated into loxg4 0 = loxd4T is gated into loxg4 0 = loxd4N is gated into loxg4 0 = loxd4N is not gated into loxg4 0 = loxd4T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is gated into loxg4 0 = loxd3T is not gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is gated into loxg4 0 = loxd2T is n								
u = Bit is unchanged x = Bit is unknown -n/n = Value at POR and BOR/Value at all other Reservers '1' = Bit is set '0' = Bit is cleared bit 7 LCxG4D4T: Gate 4 Data 4 True (non-inverted) bit 1 = lcxd4T is gated into lcxg4 0 = lcxd4T is not gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is ong gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is on gated into lcxg4 0 = lcxd2T is on gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0	-							
'1' = Bit is set '0' = Bit is cleared bit 7 LCxG4D4T: Gate 4 Data 4 True (non-inverted) bit 1 = lcxd4T is gated into lcxg4 0 = lcxd4T is not gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd4N is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is on gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4	R = Readable	bit			•			
bit 7 LCxG4D4T: Gate 4 Data 4 True (non-inverted) bit 1 = lcxd4T is gated into lcxg4 0 = lcxd4T is not gated into lcxg4 bit 6 LCxG4D4N: Gate 4 Data 4 Negated (inverted) bit 1 = lcxd4N is gated into lcxg4 0 = lcxd4N is not gated into lcxg4 bit 5 LCxG4D3T: Gate 4 Data 3 True (non-inverted) bit 1 = lcxd3T is gated into lcxg4 0 = lcxd3T is not gated into lcxg4 bit 4 LCxG4D3N: Gate 4 Data 3 Negated (inverted) bit 1 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 bit 3 LCxG4D2T: Gate 4 Data 2 True (non-inverted) bit 1 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is not gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4		anged			-n/n = Value a	at POR and BO	R/Value at all c	ther Resets
1 = lcxd4T is gated into lcxg4 0 = lcxd4T is not gated into lcxg4 bit 6 LCxG4D4N: Gate 4 Data 4 Negated (inverted) bit 1 = lcxd4N is gated into lcxg4 0 = lcxd4N is not gated into lcxg4 bit 5 LCxG4D3T: Gate 4 Data 3 True (non-inverted) bit 1 = lcxd3T is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2N is not gated into lcxg4<	'1' = Bit is set		'0' = Bit is cle	ared				
0 = lcxd4T is not gated into lcxg4 bit 6 LCxG4D4N: Gate 4 Data 4 Negated (inverted) bit 1 = lcxd4N is gated into lcxg4 0 = lcxd4N is not gated into lcxg4 bit 5 LCxG4D3T: Gate 4 Data 3 True (non-inverted) bit 1 = lcxd3T is gated into lcxg4 0 = lcxd3T is ont gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3T is not gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 <t< td=""><td>bit 7</td><td>LCxG4D4T: G</td><td>Gate 4 Data 4 1</td><td>Frue (non-inve</td><td>rted) bit</td><td></td><td></td><td></td></t<>	bit 7	LCxG4D4T: G	Gate 4 Data 4 1	Frue (non-inve	rted) bit			
1 = lcxd4N is gated into lcxg4 0 = lcxd4N is not gated into lcxg4 bit 5 LCxG4D3T: Gate 4 Data 3 True (non-inverted) bit 1 = lcxd3T is gated into lcxg4 0 = lcxd3T is not gated into lcxg4 bit 4 LCxG4D3N: Gate 4 Data 3 Negated (inverted) bit 1 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd3N is not gated into lcxg4 bit 3 LCxG4D2T: Gate 4 Data 2 True (non-inverted) bit 1 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4								
0 = lcxd4N is not gated into lcxg4 bit 5 LCxG4D3T: Gate 4 Data 3 True (non-inverted) bit 1 = lcxd3T is gated into lcxg4 0 = lcxd3T is not gated into lcxg4 bit 4 LCxG4D3N: Gate 4 Data 3 Negated (inverted) bit 1 = lcxd3N is gated into lcxg4 0 = lcxd3N is gated into lcxg4 bit 4 LCxG4D3N: Gate 4 Data 3 Negated (inverted) bit 1 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd3N is not gated into lcxg4 0 = lcxd2T: Gate 4 Data 2 True (non-inverted) bit 1 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4	bit 6	LCxG4D4N:	Gate 4 Data 4	Negated (inver	rted) bit			
1 = lcxd3T is gated into lcxg40 = lcxd3T is not gated into lcxg4bit 4LCxG4D3N: Gate 4 Data 3 Negated (inverted) bit1 = lcxd3N is gated into lcxg40 = lcxd3N is not gated into lcxg4bit 3LCxG4D2T: Gate 4 Data 2 True (non-inverted) bit1 = lcxd2T is gated into lcxg40 = lcxd2T is not gated into lcxg40 = lcxd2T is not gated into lcxg4bit 2LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit1 = lcxd2T is not gated into lcxg40 = lcxd2N is gated into lcxg40 = lcxd2N is gated into lcxg40 = lcxd2N is not gated into lcxg40 = lcxd1T: Gate 4 Data 1 True (non-inverted) bit1 = lcxd1T is gated into lcxg40 = lcxd1T is not gated into lcxg4bit 0LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit			•	•				
0 = lcxd3T is not gated into lcxg4 bit 4 LCxG4D3N: Gate 4 Data 3 Negated (inverted) bit 1 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 bit 3 LCxG4D2T: Gate 4 Data 2 True (non-inverted) bit 1 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4	bit 5	LCxG4D3T: @	Gate 4 Data 3 1	True (non-inve	rted) bit			
1 = lcxd3N is gated into lcxg4 0 = lcxd3N is not gated into lcxg4 bit 3 LCxG4D2T: Gate 4 Data 2 True (non-inverted) bit 1 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2T is not gated into lcxg4 0 = lcxd2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4								
0 = lcxd3N is not gated into lcxg4 bit 3 LCxG4D2T: Gate 4 Data 2 True (non-inverted) bit 1 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 0 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit	bit 4	LCxG4D3N:	Gate 4 Data 3	Negated (inve	rted) bit			
1 = lcxd2T is gated into lcxg4 0 = lcxd2T is not gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit								
0 = lcxd2T is not gated into lcxg4 bit 2 LCxG4D2N: Gate 4 Data 2 Negated (inverted) bit 1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit	bit 3	LCxG4D2T:	Gate 4 Data 2 1	True (non-inve	rted) bit			
1 = lcxd2N is gated into lcxg4 0 = lcxd2N is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit								
0 = lcxd2N is not gated into lcxg4 bit 1 LCxG4D1T: Gate 4 Data 1 True (non-inverted) bit 1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit	bit 2	LCxG4D2N:	Gate 4 Data 2	Negated (inver	rted) bit			
1 = lcxd1T is gated into lcxg4 0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit								
0 = lcxd1T is not gated into lcxg4 bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit	bit 1	LCxG4D1T: 0	Gate 4 Data 1 1	True (non-inve	rted) bit			
bit 0 LCxG4D1N: Gate 4 Data 1 Negated (inverted) bit			•	•				
			•	•				
1 = ICX01N is gated into ICXQ4	bit 0			•	rted) bit			
0 = lcxd1N is not gated into lcxg4			0					

REGISTER 30-10: CLCxGLS3: GATE 4 LOGIC SELECT REGISTER

REGISTER 30-11: CLCDATA: CLC DATA OUTPUT

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	_	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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	MLC4OUT: Mirror copy of LC4OUT bit
bit 2	MLC3OUT: Mirror copy of LC3OUT bit
bit 1	MLC2OUT: Mirror copy of LC2OUT bit
bit 0	MLC1OUT: Mirror copy of LC1OUT bit

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit7	Bit6	Bit5	Bit4	Blt3	Bit2	Bit1	Bit0	Register on Page
ANSELA	—	_	_	ANSA4	_	ANSA2	ANSA1	ANSA0	152
ANSELB	_	_	ANSB5	ANSB4	—	—	_	—	159
ANSELC	ANSC7(2)	ANSC6(2)	_	—	ANSC3	ANSC2	ANSC1	ANSC0	166
CLC1CON	LC1EN	_	LC10UT	LC1INTP	LC1INTN	l	C1MODE<2:0	>	443
CLCDATA	_			_	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	450
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	446
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	447
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	448
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	449
CLC1POL	LC1POL	_	_	_	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	444
CLC1SEL0	_				LC1D1	S<5:0>			445
CLC1SEL1	_				LC1D2	S<5:0>			445
CLC1SEL2	_				LC1D3	S<5:0>			453
CLC1SEL3	_				LC1D4	S<5:0>			454
CLC2CON	LC2EN		LC2OUT	LC2INTP	LC2INTN	l	C2MODE<2:0	>	443
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	446
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	447
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	448
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	449
CLC2POL	LC2POL	_	-	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	444
CLC2SEL0	—	_			LC2D1	S<5:0>			445
CLC2SEL1	—	-			LC2D2	S<5:0>			445
CLC2SEL2	—	_			LC2D3	S<5:0>			453
CLC2SEL3	—	_			LC2D4	S<5:0>			454
CLC3CON	LC3EN	-	LC3OUT	LC3INTP	LC3INTN	l	C3MODE<2:0	>	451
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	455

— = unimplemented read as '0',. Shaded cells are not used for CLC module. Unimplemented, read as '1'. Legend: Note 1:

2: PIC16(L)F1619 only.

Name	Bit7	Bit6	Bit5	Bit4	Blt3	Bit2	Bit1	Bit0	Register on Page
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	456
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	457
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	458
CLC3POL	LC3POL	_	_	_	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	452
CLC3SEL0	_	—		•	LC3D1	S<5:0>	•		453
CLC3SEL1	_	—			LC3D2	S<5:0>			453
CLC3SEL2	_	_			LC3D3	S<5:0>			453
CLC3SEL3	_	—			LC3D4	S<5:0>			454
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	l	C4MODE<2:0	>	451
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	455
CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	456
CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	457
CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	458
CLC4POL	LC4POL	_	_	_	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	452
CLC4SEL0	_	—			LC4D1	S<5:0>	•		453
CLC4SEL1	_	—	LC4D2S<5:0>				453		
CLC4SEL2	_	—			LC4D3	S<5:0>			453
CLC4SEL3	_	—			LC4D4	S<5:0>			454
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE3	_	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	108
PIR3	—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	113
TRISA	—	—	TRISA5	TRISA4	(1)	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	158
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx (continue

— = unimplemented read as '0',. Shaded cells are not used for CLC module. Unimplemented, read as '1'. Legend: Note 1:

2: PIC16(L)F1619 only.

31.0 ANGULAR TIMER (AT) MODULE

The Angular Timer (AT) module subdivides periodic signals into smaller equally spaced intervals, the number of which remain constant as the periodic signal frequency changes. A counter tracks the intervals starting at zero at each period event. The counter can be compared to user defined values to cause events, or the counter value can be captured by events external to the module. This allows for a variety of applications, such as measuring of A/C mains, stall detection for motors, commutation for brushless motors, and TDC detection for internal combustion engines. A second counter tracks the period time. This can be used to measure the error of the period based on a pre-programmed set point, as well as detect missing pulses in the signal. The angular timer includes the following features:

- · Two operating modes
 - Single-pulse per period
- Multiple-pulses per period
- Two missing pulse modes
 - Adaptive
 - Fixed
- Multiple selectable clock sources
- Phase clock output with polarity control
- · Period clock output with polarity control
- Missing pulse output with polarity control
- Interrupts for phase and period clock generation, as well as for missing pulse detect
- Period set point and error register
- Compare-pulse outputs
 - Independent interrupts
- · Capture inputs
 - Input polarity control
 - Independent interrupts

31.1 Principle of Operation

Consider the statements in Equation 31-1:

EQUATION 31-1:

If: And: Then:

$$P = \frac{F}{R}$$
 $A = \frac{F}{P}$ $A = R$

In these three equations:

- P represents the period count ATxPER
- · A represents the angle or phase count ATxPHS
- · R represents the desired resolution ATxRES
- · F represents some arbitrary scaler value

Notice that the phase count equals the desired resolution regardless of what F is. If we let F equal the ratio of a system clock to the input signal frequency then that means the phase count is a constant equaling the desired resolution regardless of the input frequency. This has many extraordinary uses including:

- Use phase compare feature to create an event at a fixed phase angle in the period
- Use capture feature to capture the phase angle at which an event occurs
- Use error feature to monitor deviations from a user specified period time

The details of these features, and more, are described in the following sections of this chapter.

31.2 Angular Timer Operating Modes

The AT module operates in two basic modes:

- · Single-Pulse mode
- Multi-Pulse mode

Both modes function on the same principle: Dividing a periodic input signal into intervals, and allowing events to trigger off of these smaller intervals. The primary difference between these two modes is how the period is determined. The Single-Pulse mode determines the period as the time between every pulse in the input pulse stream. The Multi-Pulse mode determines the period as the time between missing pulses in the input pulse stream.

The primary parameter for both modes is the ATxRES register pair. This value is used to determine the granularity of the phase counter and the frequency of the phase clock output of the module.

31.2.1 SINGLE-PULSE MODE

The operation of Single-Pulse mode is illustrated in Figure 31-1. The calculations on the input signal are done in a few distinct steps. First, there is a divider that divides the module clock by the ATxRES register pair and uses the resulting signal to increment a period counter. This operation is expressed by Equation 31-2. This equation differs slightly from that of Equation 31-1 because the counters include the count of zero. To compensate for this, the number written to the resolution register, ATxRES, must be one less than the desired resolution.

EQUATION 31-2:

$$ATxPER = \frac{\frac{F(ATxclk)}{F(ATxsig)}}{(ATxRES+1)}$$

Variables in Equation 31-2 are as follows:

- ATxPER is the value of the period counter latched by the input signal.
- ATxRES is the user-specified resolution. The phase counter will count up to this value.
- F(ATxclk) is the ATx clock frequency.
- F(ATxsig) is the input signal frequency.

The second step in the angular timer's operation is the creation of the phase clock, which is also illustrated in Figure 31-1. The input clock is divided by the ATxPER value, latched-in during the previous step, and the resulting signal is used to increment the phase counter. This signal also is used as the phase clock output, and for setting the PHSIF interrupt flag bit of the ATxIR0 register. The result is that the phase counter counts from zero to a final value expressed in Equation 31-3, outputting a pulse each time the counter increments. The value of the phase counter can be accessed by software by reading the ATxPHS register pair. However, because of the synchronization required, in order for reads of this register pair to be accurate, the instruction clock (Fosc/4) needs to be at least 3x the ATx_phsclk output frequency.

EQUATION 31-3:

$$ATxPHS(final) = \frac{\left(\frac{F(ATxclk)}{F(ATxsig)}\right)}{(ATxPER+1)}$$

The variables in Equation 31-3 are as follows:

- ATxPHS(final) is the maximum value that the phase counter will reach before being reset by the input signal. As noted in Equation 31-1, this will equal ATxRES.
- ATxPER is the maximum value of the period counter.
- F(ATxclk) is the ATx clock frequency.
- F(ATxsig) is the input signal frequency.

Notice that the division is ATxPER + 1. Ideally, this would be just ATxPER but the divider includes zero in the count. In most applications, ATxPER is a large number so the error introduced by adding one is negligible.

ATxPHS counting from 0 to ATxRES is useful when the input signal represents a rotation (for example, a motor or A/C mains). In this case, the input signal is understood to provide a period pulse every 360 degrees. Since the phase clock equally divides the signal period into a number of intervals determined by the ATxRES register pair, each pulse on the phase clock output marks a fixed phase angle in that rotation, as expressed by Equation 31-4.

EQUATION 31-4:

$$Angle Resolution = \frac{360 degrees}{AT x RES + 1}$$

ATxRES can then be used with the instantaneous value of the ATxPHS register pair to get the instantaneous angle of the rotation using Equation 31-5.

EQUATION 31-5:

$$Angle = 360 degrees \bullet \frac{ATxPHS}{ATxRES + 1}$$

31.2.2 MULTI-PULSE MODE

The operation of Multi-Pulse mode is illustrated in Figure 31-3. The calculations on the input signal are similar to those in Single-Pulse mode, with the primary difference relating to when the ATxPHS register pair is reset.

The period counter is latched into the ATxPER register pair and reset on every input pulse except the pulse immediately following a missing pulse. The first active pulse following a missing pulse triggers all of the following:

- Period clock output
- PERIF interrupt
- Phase counter reset

The result is a period clock output that has a period length equal to the time between missing pulses (e.g., a missing tooth in a gear). This leads to a significantly different relation between ATxRES and the maximum phase count, ATxPHS, as shown in Equation 31-6.

EQUATION 31-6:

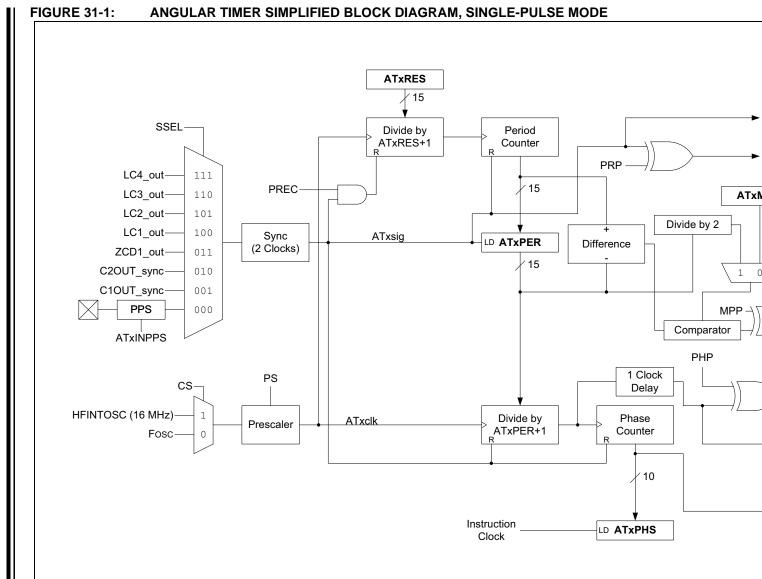
$$ATxPHS(final) = ATxRES\left(\frac{MissP}{PulseP}\right)$$

The variables in Equation 31-6 are as follows:

- MissP is the period between missing pulses
- PulseP is the period between input pulses
- ATxPHS(final) is the maximum value of the phase counter

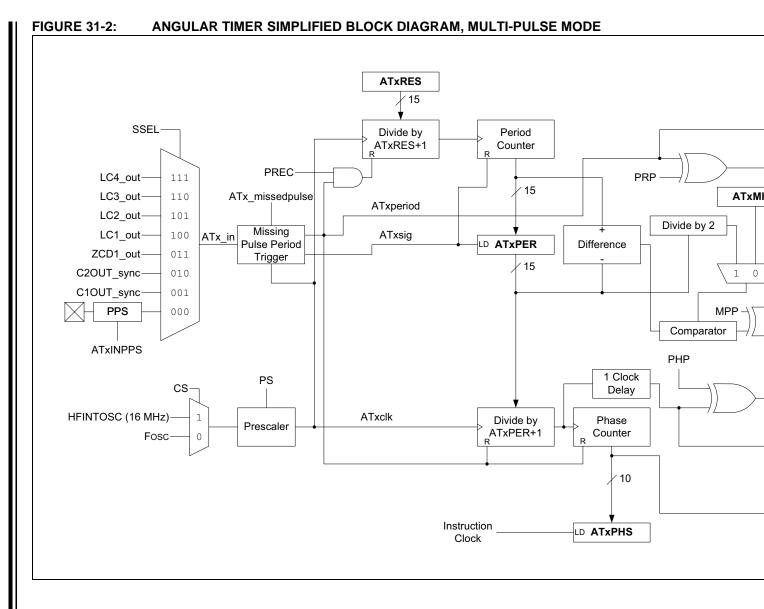
This results in a phase clock output that pulses ATxRES+1 times every input pulse, and a phase counter that increments from 0 to ATxPHS(final) over the entire time between the missing pulses.

Similar to Single-Pulse mode, this allows for triggered events to occur at fixed phase angles in the signal's period where the period is defined as the time between missing pulses. An example of multi-pulse operation is illustrated in the timing diagram of Figure 31-5, which also demonstrates what happens as a result of variations in the input signal period.



© 2014-2017 Microchip Technology Inc.

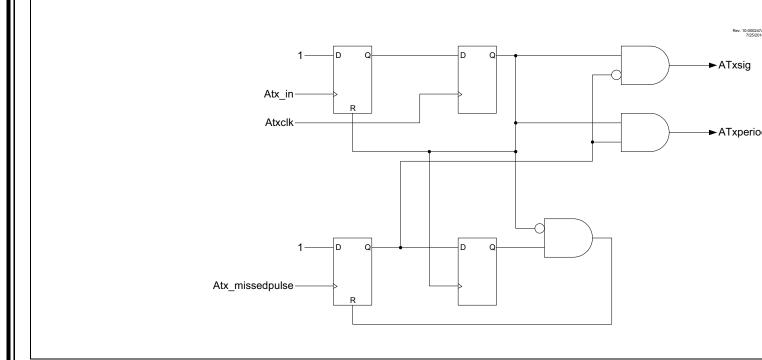
DS40001770D-page 454



DS40001770D-page 455

© 2014-2017 Microchip Technology Inc.

FIGURE 31-3: ANGULAR TIMER SIMPLIFIED MULTI-PULSE PERIOD TRIGGER BLOCK DIAGRAM



31.2.3 MISSING PULSE DETECTION

In both Single-Pulse and Multi-Pulse modes, the AT module monitors for missing pulses in the following manner. The latched value of the ATxPER register pair is continuously subtracted from the value of the period counter as it counts up. The result of this subtraction is compared to a third value and a missing pulse event is generated when the comparison is equal.

The third value is either the ATxMISS register pair or the ATxPER register pair divided by two. The APMOD bit of ATxCON0 register (Register 31-1) selects which of these two values is used.

In Single-Pulse mode, a missing pulse event generates the missing pulse output of the module as well as triggering the MISSIF interrupt.

In Multi-Pulse mode, a missing pulse event generates the output and interrupt, and is also used to determine the period signal timing.

31.2.4 MISSING PULSE MODES

Missing pulse detection has two modes of operation selected with the APMOD bit of the ATxCON0 register:

- Adaptive
- Fixed

31.2.4.1 Adaptive Missing Pulse Mode

When APMOD = 1, the missing pulse detection is in the Adaptive mode. In Adaptive mode, the difference between the period counter and the latched ATxPER value is compared to the latched ATxPER value divided by two. A missing pulse event will occur when an input signal pulse is not detected within 1.5 times the previous time between pulses. If the signal input period changes, the missing pulse comparison adapts to the change to maintain the relative time to the missing pulse event at 1.5 times the previous pulse interval.

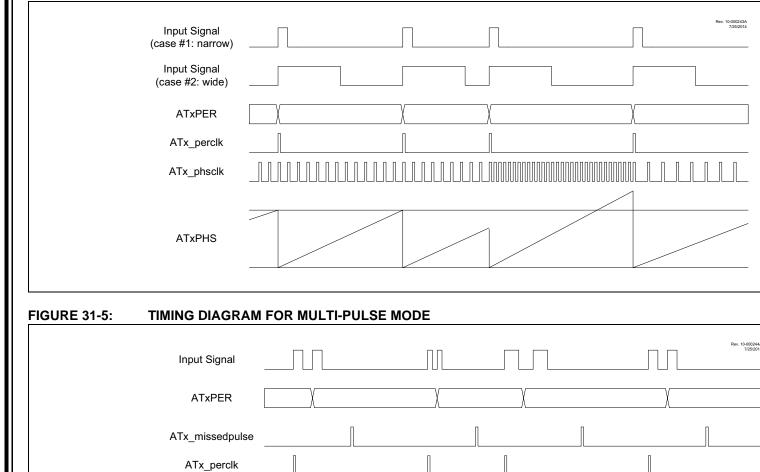
31.2.4.2 Fixed Missing Pulse Mode

When APMOD = 0, the missing pulse detection is in the Fixed mode. In Fixed mode, the difference between the period counter and the latched ATxPER value is compared to the value in the ATxMISS register pair. This gives the user absolute control over when the missing pulse will be detected, with the trade-off of not being adaptive to changes in the period.

FIGURE 31-4: TIMING DIAGRAM FOR SINGLE PULSE MODE

ATx_phsclk

ATxPHS



31.2.5 VALID BIT

Several values used by the AT module must be calculated from external signals. As such, these values may be inaccurate for a period of time after the angular timer starts up. Because of this, the module will not output signals or trigger interrupts for a period of time after the module is enabled, or under certain other conditions that might jeopardize accurate output values. This output inhibition is indicated by the read-only VALID bit of the ATxCON1 being clear.

The following cases will clear the VALID bit in hardware:

- · Any write to ATxRES register pair
- Phase counter overflow (ATxPHS register pair) clocked beyond 0x3FF)
- · In-Circuit Debugger halt
- EN = 0
- ATxPER register pair = 0
- Device Reset

As long as the VALID bit is cleared, the following occurs:

- Period clock is not output and associated interrupts do not trigger.
- Missed pulse is not output and associated interrupts do not trigger.
- Phase clock is not output and associated interrupts do not trigger.
- Phase counter does not increment.
- Capture logic does not function and associated interrupts do not trigger.
- Compare logic does not function and associated interrupts do not trigger.
- Every ATxsig edge latches the period counter into the ATxPER register pair, regardless of mode.

In single-pulse modes, the VALID bit becomes set upon the 3rd active input edge of the signal that latches the ATxPER register pair. In multi-pulse modes, a missing pulse trigger is also required, ensuring that at least one full revolution of the input has occurred.

An example of the VALID bit in Single-Pulse mode is shown in Figure 31-6.

31.2.6 DETERMINING ACCURACY

The ATxRES register pair determines the resolution of the period measurement and, by extension, the maximum value that the phase counter reaches at the end of each input signal period. The interim value, ATxPER, used to derive the phase counter is, by nature of the circuitry, an integer. The ratio of the integer value obtained by the circuit and the calculated floating point value is the inherent error of the measurement. When ATxRES is small then integer rounding results in large errors. Factors that contribute to large errors include:

- Large values for ATxRES
- Relatively low ATxclk frequency
- Relatively high ATxsig input frequency

The actual error can be determined with Equation 31-7.

EQUATION 31-7:

$$period = \frac{F(ATxclk)}{F(ATxsig) \bullet (ATxRES + 1)}$$
$$error\% = 100 \bullet \left(\frac{period - int(period + 1)}{period}\right)$$

31.3 Input and Clock Selection

The input clock for the AT module can come from either the Fosc system clock or the 16 MHz HFINTOSC, and is chosen by the CS0 bit of the ATxCLK register. In addition, the clock is run through a prescaler that can be /1, /2, /4, or /8, which is configured by the PS<1:0> bits of the ATxCON0 register. This prescaled clock is then used for all clock operations of the Angular Timer, and as such, should be used for all of the equations demonstrated above determining the Angular Timer's behavior.

The input signal for the AT module can come from a variety of sources. The source is selected by the SSEL bits of the ATxSIG register (Register 31-4).

31.4 Module Outputs

31.4.1 ANGLE/PHASE CLOCK OUTPUT

The angle/phase clock signal (ATx_phsclk) can be used by the CLC as an input signal to combinational logic. The polarity of this signal is configured by the PHP bit of the ATxCON1 register.

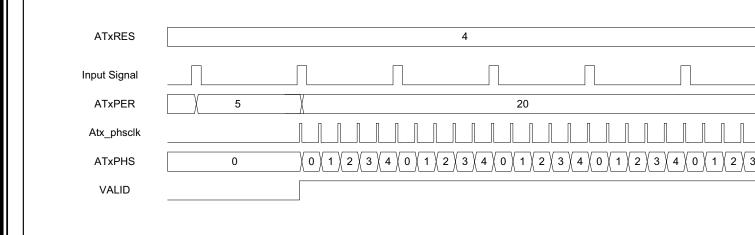
31.4.2 PERIOD CLOCK OUTPUT

The period clock signal (ATx_perclk) can be used as an input clock for the Timer2/4/6 and Signal Measurement module, as well as an input signal to the CLC for combinational logic. The polarity of this signal is configured by the PRP bit of the ATxCON1 register (Register 31-2).

31.4.3 MISSED PULSE OUTPUT

The missed pulse signal (ATx_missedpulse) can be used by the CLC as an input signal to combinational logic. The polarity of this signal is configured by the MPP bit of the ATxCON1 register.

FIGURE 31-6: EXAMPLE OPERATION (ATxRES = 4)

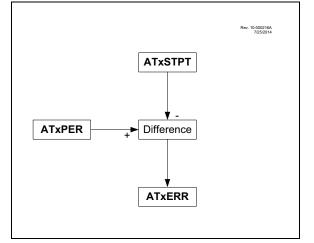


31.5 Period Set Point and Error Measurement

The ATxSTPT register pair controls the period set point of the AT module. The signal period captured in the ATxPER register pair at every signal input pulse. The unsigned 15-bit ATxSTPT value is subtracted from the unsigned 15-bit ATxPER value and the signed 16-bit result is placed in the ATxERR register pair.

The ATxSTPT value is double buffered requiring an ATxSTPTL value write for the ATxSTPTH value to take effect. This is done so that all 16 bits update at the same time, thereby avoiding a miscalculation of the error.

FIGURE 31-7: ANGULAR TIMER SET POINT CALCULATION BLOCK DIAGRAM



31.6 Capture and Compare Functions

The angular timer contains multiple built-in capture/compare modules. These are controlled by their respective ATxCCONy registers where "x" refers to the AT instance and "y' refers to the Capture/Compare instance within that AT module.

This particular device contains three capture/compare modules within the AT module. The CCyMODE bit of the ATxCCONy register controls whether each particular module is in Capture or Compare mode. The polarity of each module's respective output signal is controlled by the CCyPOL bit of the ATxCCONy register (Register 31-21). Both the Capture and Compare modes use an edge detect that runs off of the ATxclk signal.

31.6.1 CAPTURE MODE

Capture mode is selected when the CCyMODE bit (of the ATxCCONy register) = 1. Refer to Figure 31-8.

In Capture mode, the value of the phase counter is written to the respective ATxCCy registers on the rising edge of the capture input signal.

The capture event also generates a pulse that can be used for the following:

- Trigger an ADC reading
- CLC logic input
- · Set the CCyIF bit

See **Section 31.7 "Interrupts**" for more details on the interrupts triggered by the AT module.

The capture input signal source is selected by the capture/compare's respective ATxCSELy register (Register 31-22), and its polarity is selected by the ATxCAPyP bit of the ATxCCONy register (Register 31-21). Note that when in Capture mode, the ATxCCy register pair is read-only.

31.6.2 COMPARE MODE

Compare mode is selected when the CCyMODE bit (of the ATxCCONy register) = 0. Refer to Figure 31-9.

In Compare mode, the module compares the current value in the ATxCCy register pair to the phase counter value. When the two values are equal then a compare event is generated and output to the following:

- Trigger an ADC reading
- CLC logic input
- · Set the CCyIF bit

See **Section 31.7 "Interrupts**" for more details on the interrupts triggered by the AT module.

FIGURE 31-8: ANGULAR TIMER CAPTURE/COMPARE UNIT BLOCK DIAGRAM: CAPTURE MODE

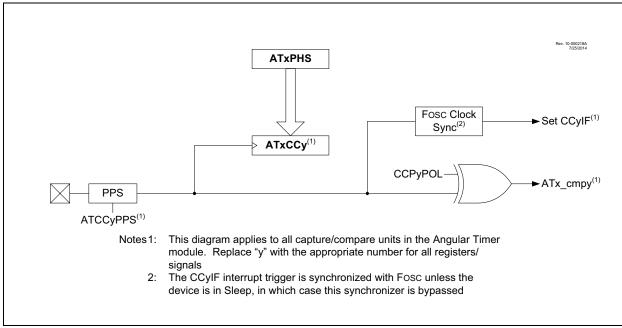
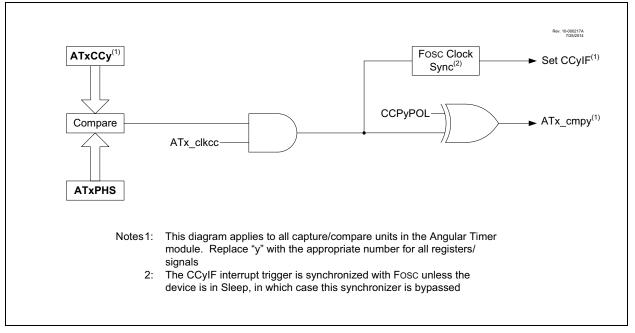


FIGURE 31-9: ANGULAR TIMER CAPTURE/COMPARE UNIT BLOCK DIAGRAM: COMPARE MODE



31.7 Interrupts

The angular timer and its capture/compare features can generate multiple interrupt conditions. To accommodate all of these interrupt sources, the module is provided with its own interrupt logic structure, similar to that of the micro controller. Angular timer interrupts are enabled by the ATxIE0 register (Register 31-13) and their respective flags are located in the ATxIR0 register (Register 31-14). The capture/compare interrupts are enabled by the ATxIE1 register (Register 31-15) with flags in the ATxIR1 register (Register 31-16). All sources are funneled into a single Angular Timer Interrupt Flag bit, ATxIF of the PIR5 register (Register 7-11). This means that upon a triggered interrupt, the ATxIR0 and ATxIR1 register bits will indicate the source of the triggered interrupt. It also means that in order for specific interrupts to generate a microcontroller interrupt, both the ATxIE bit of the PIE register and the desired enable bit in either ATxIE0 or ATxIE1 must be set.

Note:	Due to the nature of the angular timer
	interrupts, the ATxIF flag bit of the PIR5
	register is read-only.

31.7.1 ANGULAR TIMER PERIOD INTERRUPT

This interrupt is triggered when the AT module detects a period event. In Single-Pulse mode, a period event occurs on every input signal edge. In Multi-Pulse mode, a period event occurs on the input signal edge following a missed pulse. The period interrupt generation matches with the pulses on the period clock output of the timer. It is enabled by the ATPERIE bit of the ATxIE0 register and the status is indicated by the PERIF bit of the ATxIR0 register.

31.7.2 ANGULAR TIMER PHASE CLOCK INTERRUPT

This interrupt is triggered on each pulse of the phase clock output of the timer. It is enabled by the ATPHIE bit of the ATxIE0 register and the status is indicated by the PHSIF bit of the ATxIR0 register.

31.7.3 ANGULAR TIMER MISSING PULSE INTERRUPT

This interrupt is triggered upon the output of a missing pulse detection signal. Refer to **Section 31.2.3 "Missing Pulse Detection**" for more information. This interrupt is enabled by the ATMISSIE bit of the ATXIE0 register and its status is indicated by the ATMISSIF bit of the ATXIR0 register.

31.7.4 ANGULAR TIMER CAPTURE/COMPARE INTERRUPTS

Capture and compare interrupts are triggered by the capture/compare functions of the module. If configured for Capture mode, the interrupt will trigger after the capture signal has successfully latched the value of the phase counter into the capture registers. If configured for Compare mode, the interrupt will trigger when a match is detected between the value placed in the compare register and the value of the phase counter. These interrupts are controlled by the CC1IE, CC2IE, and CC3IE bits of the ATxIE1 register, respectively, and are similarly indicated by the CC1IF, CC2IF, and CC3IF bits of the ATxIR1 register.

31.8 Angular Timer Control Registers

Long bit name prefixes for the angular timer peripherals are shown in Table 31-1. Refer to **Section 1.1 "Register and Bit Naming Conventions**" for more information.

TABLE 31-1:

Peripheral	Bit Name Prefix
AT1	AT1

REGISTER 31-1: ATxCON0: ANGULAR TIMER CONTROL 0 REGISTER

R/W-0/0	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		
EN	PREC	PS<		POL	_	APMOD	MODE		
bit 7			-				bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	d as '0'			
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all o	ther Resets		
'1' = Bit is set		'0' = Bit is cle	ared	q = Value dep	ends on condi	tion			
bit 7	-	Timer Enable b							
		timer is enabled timer is disable		cks are active					
bit 6	PREC: Perio	d Precision Cor	ntrol bit						
		rescaler is rese							
	0 = Period pi suremen		reset at the st	art of every per	riod; fraction pe	eriod affects nex	t period mea-		
bit 5-4		ock Prescaler C	ontrol hits						
bit o T				cale logic is clo	cked by ATxCL	.K/8			
				cale logic is clo					
	 01 = Resolution and phase counter prescale logic is clocked by ATxCLK/2 00 = Resolution and phase counter prescale logic is clocked by ATxCLK 								
bit 3		Active Edge (P	-	-		.r.			
DIL 3	•	dge of ATxsig i	• /						
		dge of ATxsig is							
bit 2	Unimplemen	ted: Read as '	0'						
bit 1	APMOD: Ada	aptive Missing F	Pulse Mode S	elect bit					
	•	•	e mode. Miss	sing pulse is c	letected when	no pulse is de	etected within		
		SATXPER	DO ATVMISS	register pair d	atarminas miss	ing pulse event			
bit 0		lar Timer Mode		register pair u			•		
Sit U	•			(period of input	signal defined	by missing puls	ses)		
	•				•	d by input pulse	,		
	0 = Angular	timer is in Singl	e-Pulse mode	e (period of inp	ut signal define	d by input pulse	es)		

U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R-0/0	R-0/0
_	PHP	—	PRP		MPP	ACCS	VALID
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 BO	R/Value at all c	other Resets
'1' = Bit is se	et	'0' = Bit is clea	ared	q = Value de	pends on conditi	on	
bit 7	Unimplemen	ted: Read as '	0'				
bit 6	PHP: Phase	Clock Output P	olarity bit				
	1 = Phase clock output is active-low						
		ock output is a	•				
bit 5	Unimplemen	ted: Read as '	0'				
bit 4		Clock Output P	2				
		ock output is a					
h:1 0		ock output is a	•				
bit 3	•	ted: Read as '					
bit 2		g Pulse Output	,				
	01	pulse output is pulse output is p					
bit 1	•	eration Sign bit	•				
		e currently in A		s than the prev	ious value		
	0 = The valu	e currently in A	TxPER is gre	ater than or eq	ual to the previo	ous value	
bit 0	VALID: Valid	Measurement	bit				
					ER and ATxPHS		
	0 = The valu	es in ATxPER	and ATxPHS	are not valid; n	ot enough input	cycles have o	ccurred

REGISTER 31-2: ATxCON1: ANGULAR TIMER CONTROL 1 REGISTER

REGISTER 31-3: ATxCLK: ANGULAR TIMER CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/x
—	—	—	—	—	—	—	CS0
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 7-1 Unimplemented: Read as '0' bit 0 CS0: Angular Timer Clock Selection bit 1 = HFINTOSC 16 MHz 0 = Fosc

REGISTER 31-4: ATxSIG: ANGULAR TIMER INPUT SIGNAL SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	—	—		SSEL<2:0>	
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	q = Value depends on condition

bit 7-3 Unimplemented: Read as '0'

bit 2-0

SSEL<2:0>: Angular Input Signal Selection bit 111 = LC4_out 110 = LC3_out 101 = LC2_out 100 = LC1_out 011 = ZCD1_out 010 = cmp2_sync 001 = cmp1_sync 000 = ATxINPPS

REGISTER 31-5: ATXRESH: ANGULAR TIMER RESOLUTION HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	
_	—	—	-	—	—	RES<9:8>		
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	q = Value depends on condition

bit 1-0 **RES<9:8>:** ATxRES Most Significant bits, the Phase Counter Resolution

Note 1: Writing to this register resets VALID bit of the ATxCON1 (Register 31-2); output signals are inhibited for at least two input cycles.

2: This register is not guarded for atomic access, and should only be accessed while the timer is not running.

REGISTER 31-6: ATXRESL: ANGULAR TIMER RESOLUTION LOW 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
			RES	<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	q = Value depends on condition

bit 7-0 **RES<7:0>:** ATxRES Least Significant bits, the Phase Counter Resolution

- Note 1: Writing to this register resets VALID bit of the ATxCON1 (Register 31-2); output signals are inhibited for at least two input cycles.
 - 2: This register is not guarded for atomic access, and should only be accessed while the timer is not running.

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			
	MISS<15:8> ⁽¹⁾									
bit 7							bit 0			
Legend:										
R = Readable	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 clea	ared	q = Value dep	pends on condit	ion				

REGISTER 31-7: ATXMISSH: ANGULAR TIMER MISSING PULSE DELAY HIGH REGISTER

- bit 7-0 **MISS<15:8>⁽¹⁾:** Most Significant bits (2's complement) of ATxMISS. ATxMISS defines the period counter value at which the missing pulse output becomes valid, based on the difference between the current counter value and the latched-in value of ATxPER.
- **Note 1:** ATxMISSH is held until ATxMISSL is written. Proper writes of ATxMISS should write to ATxMISSH first, then ATxMISSL to ensure the value is properly written.

REGISTER 31-8: ATXMISSL: ANGULAR TIMER MISSING PULSE DELAY LOW 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
			MISS	<7:0>			
bit 7							bit 0
Legend:							
P - Roadable b	.it	M = Mritable bit	ł		nonted hit read	as 'O'	

'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
R = Readable bit	vv = vvritable bit	O = Onimplemented bit, read as 'O

bit 7-0 **MISS<7:0>:** Least Significant bits (2's complement) of ATxMISS. ATxMISS defines the period counter value at which the missing pulse output becomes valid, based on the difference between the current counter value and the latched-in value of ATxPER.

REGISTER 31-9: ATxPERH: ANGULAR TIMER MEASURED PERIOD HIGH REGISTER

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x		
POV				PER<14:8>					
bit 7							bit 0		
Legend:									
R = Readabl	e bit	W = Writable	bit	U = Unimplem	nented bit, read	l as '0'			
u = Bit is und	changed	x = Bit is unkr	nown	-n/n = Value at POR and BOR/Value at all other Resets					
'1' = Bit is se	t	'0' = Bit is cle	ared	q = Value depends on condition					
bit 7	POV: Period	d Counter Overfl	ow bit						
	 1 = Counter rolled over one or more times during measurement 0 = Value shown by ATxPER is valid 								

bit 6-0 **PER<14:8>:** Most Significant bits of ATxPER. ATxPER is the measured period value from the period counter.

REGISTER 31-10: ATxPERL: ANGULAR TIMER MEASURED PERIOD LOW REGISTER

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
			PER	<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	q = Value depends on condition

bit 7-0 **PER<7:0>:** Least Significant bits of ATxPER. ATxPER is the measured period value from the period counter.

REGISTER 31-11: ATxPHSH: ANGULAR TIMER PHASE COUNTER HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-x/x	R-x/x	
	_	—	—	—	—	PHS	<9:8>	
bit 7							bit 0	
Legend:								
R = Readable bit W = Writable bit		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				

bit 7-2 Unimplemented: Read as '0'

1' = Bit is set

bit 1-0 PHS<9:8>: Most Significant bits of ATxPHS. ATxPHS is the instantaneous value of the phase counter.

q = Value depends on condition

REGISTER 31-12: ATxPHSL: ANGULAR TIMER PHASE COUNTER LOW REGISTER

'0' = Bit is cleared

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	
PHS<7:0>								
bit 7							bit 0	
L								

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 7-0 **PHS<7:0>:** Least Significant bits of ATxPHS. ATxPHS is the instantaneous value of the phase counter.

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	
—	—	—	—	_	PHSIE	MISSIE	PERIE	
bit 7							bit 0	
Legend:								
R = Readab	le bit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'		
u = Bit is un	changed	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all o	ther Resets	
'1' = Bit is se	et	'0' = Bit is clea	ared	q = Value dep	pends on conditi	on		
bit 7-3	Unimplemen	ted: Read as '	כי					
bit 2	PHSIE: Phas	e Interrupt Ena	ble bit					
		se interrupt is e						
	0 = The phas	se interrupt is d	isabled					
bit 1	MISSIE: Miss	ed Pulse Interr	upt Enable bit					
	1 = The missed pulse interrupt is enabled							
	0 = The missed pulse interrupt is disabled							
bit 0	PERIE: Perio	d Interrupt Ena	ble bit					
		od interrupt is e						
	0 = The period	od interrupt is d	isabled					

REGISTER 31-13: ATxIE0: ANGULAR TIMER ENABLE 0 REGISTER

REGISTER 31-14: ATxIR0: ANGULAR TIMER INTERRUPT FLAG 0 REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	PHSIF	MISSIF	PERIF
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 7-3	Unimplemented: Read as '0'
bit 2	PHSIF: Phase Interrupt Flag bit
	1 = The phase interrupt has occurred
	0 = The phase interrupt has not occurred, or has been cleared
bit 1	MISSIF: Missed Pulse Interrupt Flag bit
	1 = The missed pulse interrupt has occurred
	0 = The missed pulse interrupt has not occurred, or has been cleared
bit 0	PERIF: Period Interrupt Flag bit
	1 = The period interrupt has occurred
	0 = The period interrupt has not occurred, or has been cleared

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
	—	—	—	—	CC3IE	CC2IE	CC1IE
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplei	mented bit, read	as '0'	
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value	at POR and BO	R/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is clea	ared	q = Value de	pends on conditi	ion	
bit 7-3	Unimplemen	ted: Read as '	כ'				
bit 2 bit 1	If CC3MODE 1 = Capture i 0 = Capture i If CC3MODE 1 = Compare 0 = Compare CC2IE: Capture I = Capture i 0 = Capture i 1 = Capture i 1 = Compare	ure/Compare In = 1 (Capture) interrupt 3 is er interrupt 3 is di = 0 (Compare) e interrupt 3 is e interrupt 3 is e interrupt 3 is c ure/Compare In = 1 (Capture) interrupt 2 is er interrupt 2 is er e interrupt 2 is er interrupt 2 is er interrupt 2 is er interrupt 2 is er interrupt 2 is er	nabled sabled lisabled terrupt 2 Ena nabled sabled				
bit 0	CC1IE : Captu If CC1MODE 1 = Capture i 0 = Capture i If CC1MODE	ure/Compare In = 1 (Capture) interrupt 1 is er interrupt 1 is di = 0 (Compare) interrupt 1 is e	terrupt 1 Ena nabled sabled	ble bit			

REGISTER 31-15: ATxIE1: ANGULAR TIMER ENABLE 1 REGISTER

0 = Compare interrupt 1 is disabled

REGISTER 31-16: ATXIR1: ANGULAR TIMER INTERRUPT FLAG 1 REGISTER

0 = Compare interrupt 2 has not occurred, or has been cleared

0 = Capture interrupt 1 has not occurred, or has been cleared

0 = Compare interrupt 1 has not occurred, or has been cleared

1 = Capture interrupt 1 has occurred; captured phase value is in ATxCC1

CC1IF: Capture/Compare Interrupt 1 Flag bit

If CC1MODE = 1 (Capture)

If CC1MODE = 0 (Compare)

1 = Compare interrupt 1 has occurred

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	_		—	_	CC3IF	CC2IF	CC1IF
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	lown	-n/n = Value	at POR and BO	R/Value at all c	ther Resets
'1' = Bit is se	et	'0' = Bit is clea	ared	q = Value de	pends on condit	ion	
bit 2 bit 1	If CC3MODE 1 = Capture 0 = Capture If CC3MODE 1 = Compare 0 = Compare	ure/Compare In = <u>1 (Capture)</u> interrupt 3 has interrupt 3 has = <u>0 (Compare)</u> e interrupt 3 has e interrupt 3 has ure/Compare In	occurred; cap not occurred, occurred not occurred	tured phase va or has been cl I, or has been o		23	
	If CC2MODE 1 = Capture 0 = Capture If CC2MODE	= 1 (Capture)	occurred; cap not occurred,	tured phase va	alue is in ATxCC leared	2	

bit 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	R/W-x/u
—				STPT<14:8>			
bit 7							bit 0
Legend:							
R = Readable I	oit	W = Writable	bit	U = Unimplen	nented bit, read	l as '0'	
u = Bit is uncha	anged	x = Bit is unknown -n/n = Value at POR and BOR/Value at all other R				ther Resets	
'1' = Bit is set		'0' = Bit is clea	ared	a = Value dep	ends on condit	ion	

REGISTER 31-17: ATxSTPTH: ANGULAR TIMER SET POINT HIGH REGISTER ⁽¹⁾

bit 7 Unimplemented: Read as '0'

'1' = Bit is set

bit 6-0 **STPT<14:8>:** Set Point Most Significant bits. ATxSTPT determines the threshold setting that the period is compared against for error calculation.

REGISTER 31-18: ATxSTPTL: ANGULAR TIMER SET POINT LOW REGISTER

'0' = Bit is cleared

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
			STP	<7:0>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	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 o	ther Resets

q = Value depends on condition

bit 7-0 **STPT<7:0>:** Set Point Least Significant bits. ATxSTPT determines the threshold setting that the period is compared against for error calculation.

Note 1: Writes to ATxSTPTH are double buffered. The value written to this register is held until a write to ATxSTPTL occurs, at which point the value will be latched into the register

REGISTER 31-19: ATXERRH: ANGULAR TIMER SET POINT ERROR VALUE HIGH REGISTER

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
			ERR	<15:8>			
bit 7							bit 0
Legend:							
R = Readable b	oit	W = Writable bit		U = Unimpler	nented bit, read	l as '0'	
u = Bit is uncha	nged	x = Bit is unknow	/n	-n/n = Value a	at POR and BO	R/Value at all o	ther Resets
'1' = Bit is set		'0' = Bit is cleare	d	q = Value der	pends on condit	ion	

bit 7-0 **ERR<15:8>:** Most Significant bits of ATxERR. ATxERR is the error of the measured period value compared to the threshold setting, defined as ATxPER-ATxSTPTP.

REGISTER 31-20: ATxERRL: ANGULAR TIMER SET POINT ERROR VALUE LOW REGISTER

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
			ERR	<7:0>			
bit 7							

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 7-0 **ERR<7:0>:** Least Significant bits of ATxERR. ATxERR is the error of the measured period value compared to the threshold setting, defined as ATxPER-ATxSTPTP.

REGISTER 31-21: ATxCCONy: ANGULAR TIMER CAPTURE/COMPARE CONTROL 1 REGISTER

		-					
R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
CCyEN	—	—	CCPyPOL	CAPyP	_	—	CCyMODE
bit 7							bit 0
Legend:							
R = Readable bitW = Writable bitU = Unimplemented bit, read as '0'							
u = Bit is unch	anged	x = Bit is unkr	nown		at POR and BO		other Resets
'1' = Bit is set		'0' = Bit is cle	ared	q = Value dep	pends on condit	ion	
bit 7	1 = Capture/	ture/Compare I Compare logic Compare logic	is enabled				
bit 6-5	Unimplemen	ted: Read as '	0'				
bit 4	CCyPOL: Ca	pture/Compare	Output Polar	ity bit			
	1 = ATxCCO	ode (CCyMOD UT1 is active le UT1 is active h	ow when ATx0	•			
	1 = ATxCCO	node (CCyMOI UT1 is active le UT1 is active h	ow when ATxI				
bit 3	CAPyP: Capt	ture Input Pola	rity bit				
	 CAPyP: Capture Input Polarity bit <u>In Capture mode (CCyMODE = 1):</u> 1 = At falling edge of the capture input (Selected by ATxCSELy) the value of the phase counter captured in ATxCC1 0 = At rising edge of the capture input (Selected by ATxCSELy) the value of the phase counter captured in ATxCC1 						
	In Compare n	node (CCyMOI	DE = 0 <u>):</u>				
	This bit is ign	ored.					
bit 2-1	Unimplemen	ted: Read as '	0'				
bit 0	1 = Capture/	Capture/Compa compare logic compare logic	is in Capture r	node			

REGISTER 31-22: ATxCSELy: ANGULAR TIMER CAPTURE INPUT SELECT y REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—		CPyS<2:0>	
bit 7				•			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	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	q = Value dep	pends on condi	tion	

Unimplemented: Read as '0'
CPyS<2:0>: Capture Input Source Select bits
111 = CWG_interrupt
110 = LC4_out
101 = LC3_out
100 = LC2_out
111 = LC1_out
010 = cmp2_sync
001 = cmp1_sync
000 = ATxCCy pin

REGISTER 31-23: ATxCCyH: ANGULAR TIMER CAPTURE/COMPARE y HIGH REGISTER

- - - - CCy<9:8> bit 7 bit bit	U-0	U-0	U-0	U-0	U-0	U-0	R/q-0/0	R/q-0/0
bit 7 bi	—	—	—	—	—	—	CCy	<9:8>
	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	q = Value depends on condition

bit 7-2	2 Unimplemented: Read as '0'
bit 1-(CCy<9:8>: ATxCCy Most Significant bits
	In Capture mode (CCyMODE = 1) (Read-only): ATxCCy is the captured value of ATxPHS when the capture input is signaled.
	In Compare mode (CCyMODE = 0): ATxCCy is the value that is compared to the current value of ATxPHS to trigger an interrupt/output pulse.
Note	1: Writes to ATxCCyH are double buffered. The value written to this register is held until a write to ATxCCyL

occurs, at which point the value will be latched into the register

REGISTER 31-24: ATxCCyL: ANGULAR TIMER CAPTURE/COMPARE y LOW REGISTER

R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0
			CCy	<7:0>			
bit 7							bit 0
Legend:							

Legena.		
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 7-0 CCy<7:0>: ATxCCy Least Significant bits

In Capture mode (CCyMODE = 1) (Read-only):

ATxCCy is the captured value of ATxPHS when the capture input is signaled.

In Compare mode (CCyMODE = 0):

ATxCCy is the value that is compared to the current value of ATxPHS to trigger an interrupt/output pulse.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
AT1CC1H	—	—	—	—	—	—	CC1	<9:8>	478
AT1CC1L		CC1<7:0>							
AT1CCON1	CC1EN	—	_	CC1POL	CAP1P	_	_	CC1MODE	476
AT1CCON2	CC2EN	_	—	CC2POL	CAP2P	_	—	CC2MODE	476
AT1CCON3	CC3EN	—	—	CC3POL	CAP3P	_	—	CC3MODE	476
AT1CLK	—	—	—	—	—	—	—	CS0	466
AT1CON0	EN	PREC	PS<	1:0>	POL	_	APMOD	MODE	464
AT1CON1	_	PHP	—	PRP	—	MPP	ACCS	VALID	465
AT1CSEL1	—	CP1S<2:0>						477	
AT1CSEL2	—	—	—	—	—		477		
AT1CSEL3	—	—	—	—	—	CP3S<2:0>			477
AT1ERRH		ERR<15:8>						475	
AT1ERRL		ERR<7:0>						475	
AT1IE0		—	_	—	—	PHSIE	MISSIE	PERIE	471
AT1IR0		—	—	—	—	PHSIF	MISSIF	PERIF	471
AT1IE1		—	_	—	—	CC3IE	CC2IE	CC1IE	472
AT1IR1		—	—	—	—	CC3IF	CC2IF	CC1IF	473
AT1MISSH	MISS<15:8>								468
AT1MISSL	MISS<7:0>								468
AT1PERH	POV	PER<14:8>						469	
AT1PERL		PER<7:0>						469	
AT1PHSH	—	—	—	—	—	— PHS<9:8>			470
AT1PHSL		PHS<7:0>						470	
AT1RESH	—	—	—	—	—	_	RES	<9:8>	467
AT1RESL		RES<7:0>							467
AT1SIG	—	SSEL<2:0>						466	
AT1STPTH	_				STPT<14:8>	>			474
AT1STPTL		1		STPT	۲<7:0>				474
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	_	AT1IE	PID1EIE	PID1DIE	102
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	_	AT1IF	PID1EIF	PID1DIF	107

TABLE 31-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE ANGULAR TIMER MODULE

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the AT module.

32.0 MATH ACCELERATOR WITH PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) MODULE

The math accelerator module is a mathematics module that can perform a variety of operations, most prominently acting as a PID (Proportional-Integral-Derivative) controller. A PID controller is an algorithm that uses the present error (proportional), the sum of the present and all previous errors (integral), and the difference between the present and previous change (derivative) to correct errors and provide stability in a system. It provides feedback to a system through a series of iterations, using the present error as well as previous errors to calculate a new input to the controller. The data flow for both PID modes is illustrated in Figure 32-1.

The module accomplishes the task of calculating the PID algorithm by utilizing user-provided coefficients along with a multiplier and accumulator. As such, this multiplier and accumulator can also be configured to quickly and efficiently perform signed and unsigned multiply-and-add calculations both with and without accumulation. The data flow for these modes is illustrated in Figure 32-2.

Features of this module include:

- · Signed multiplier
- 35-bit signed accumulator
- PID controller support with user inputs for K1, K2, K3, system error and desired set point
- · Completion and Error interrupts
- Multiple user modes allowing for PID with or without accumulation as well as several multiplication operations

32.1 PID Module Setup Summary

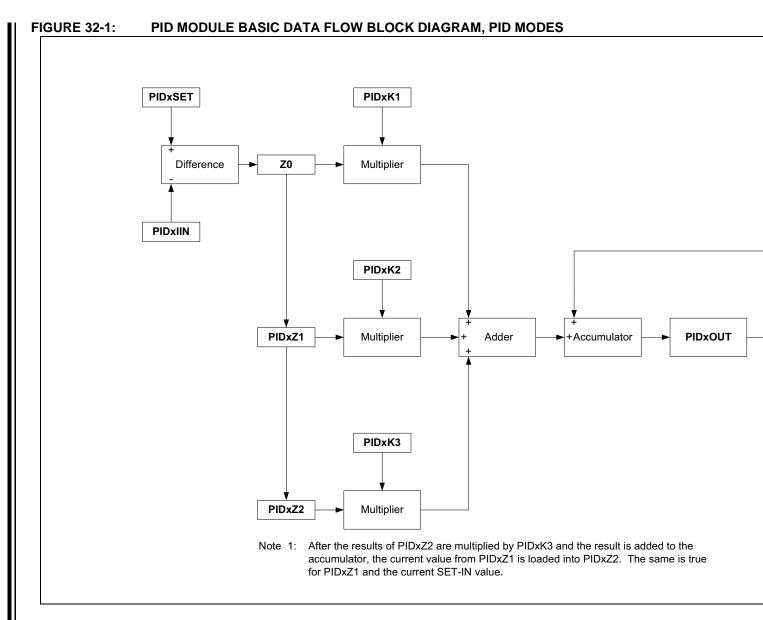
The PID module can be configured either as a PID controller or as a multiply and accumulate module. Multiply and accumulate can be performed in four modes:

- Unsigned multiply and add, without accumulation
- Unsigned multiply and accumulate
- · Signed multiply and add, without accumulation
- · Signed multiply and accumulate

All of the modes are selected by the MODE<2:0> bits of the PIDxCON register.

32.1.1 PID MODE SETUP AND OPERATION

When the MODE<2:0> bits of the PIDxCON register are equal to '101', the module is in PID controller mode. The operation of the module in PID controller mode is generally performed as a loop. The input from an external system is fed into the controller, and the controller's output is fed back into the external system. This will produce a new response from the system that is then looped back into the PID controller. The data flow for the PID operation is illustrated in Figure 32-1.



DS40001770D-page 481

© 2014-2017 Microchip Technology Inc.

Within the controller, the input is subtracted from a preprogrammed set point to get an error value. This error value, along with the previous two error values (if any), are multiplied by user-input coefficients and the results of these multiplications are added together to make up the output. If the MODE<2:0> bits of the PIDxCON register = 101, the PID output is equal to the current output added to any previous outputs.

The three user-input coefficients (K1, K2, and K3) are derived from the three classic PID coefficients Kp, Ki, and Kd, and must be calculated prior to using the PID module.

1. K1 is the coefficient that is multiplied with the current error (SET-IN). It is defined by the following equation:

EQUATION 32-1:

$$K1 = Kp + Ki \bullet T + \frac{Kd}{T}$$

Note: T is the sampling period.

 K2 is the coefficient that is multiplied with the previous iteration's error (Z1). Where T is the sampling period, it is defined by the following equation:

EQUATION 32-2:

$$K2 = -\left(Kp + \frac{2Kd}{T}\right)$$

Note: T is the sampling period.

3. K3 is the coefficient that is multiplied with the error that occurred two iterations previous to the current one (Z2). It is defined by the following equation:

EQUATION 32-3:

$$K3 = \frac{Kd}{T}$$

Note: T is the sampling period.

To operate the module in PID controller mode, perform the following steps:

- Set the MODE<2:0> bits of the PIDxCON register to '101', then set the EN bit of the PIDxCON register.
- 2. Write the previously calculated K1, K2, and K3 values to the PIDxK1, PIDxK2, and PIDxK3 registers, respectively.
- 3. Write the desired set point that the input will be compared against to the PIDxSET registers.
- Write the high byte of the value from the external system to PIDxINH. Then write the low byte of the value from the external system to PIDxINL. This will begin the calculation and set the BUSY bit of the PIDxCON register.
- 5. Either poll the BUSY bit of the PIDxCON register to check for it clearing or wait for the PIDxDIF interrupt to trigger, indicating that the operation has completed.
- Read the PIDxOUT registers for the output value. If the PID was in Accumulation mode, PIDxOUT will contain the accumulation of the output added to the previous outputs, otherwise, it will contain only the latest output.
- 7. For proper PID operation, this output needs to be applied to the external system before the next input to the PID is applied. This is to ensure that the system can adjust based on the PID controller's feedback before the next calculation is made.
- Note: The BUSY bit of the PIDxCON register goes high as soon as PIDxINL is written and remains high until all computation is complete. Until the BUSY bit goes low, the PIDxOUT values are not valid, and none of the registers associated with the PID module should be written to, as any such writes will corrupt the calculation.

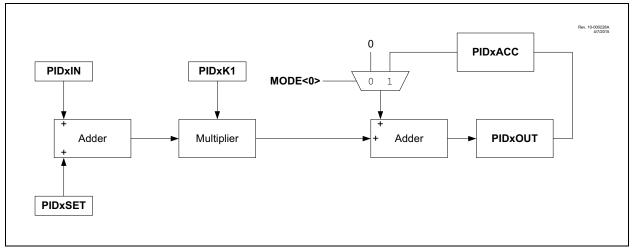
32.1.2 CONTEXT SAVING

It is possible to save the current state of the PID controller in software and restore it at a later time. In order to perform this, a calculation must not currently be active (BUSY = 0). Saving the PIDxOUT, PIDxZ1, and PIDxZ2 values elsewhere in memory will save the current state of the PID controller, although it may be desirable to also save PIDxK1, PIDxK2, PIDxK3, and/ or PIDxSET, depending on the application. At the desired later time, these values can be written back into their respective registers, writing PIDxINL last, and the PID will continue from its previous state.

32.2 Add and Multiply Mode Setup and Operation

The PID module can also be used to perform 16-bit Add and Multiply computations. When the MODE<2:0> bits of the PIDxCON register are equal to '000', '001', '010', or '011', the module is in Add and Multiply mode. The data flow for the multiply and add operation is illustrated in Figure 32-2.

FIGURE 32-2: PID MODULE BASIC DATA FLOW BLOCK DIAGRAM, ADD AND MULTIPLY MODES



All Add and Multiply modes perform operations of the following form.

EQUATION 32-4:

$$OUTPUT = (A+B) \cdot C$$

Note: A = PIDxIN, B = PICxSET, and C = PIDxK1.

The four different Add and Multiply modes are:

- MODE<2:0> = 000: Inputs are unsigned, and the output does not accumulate
- MODE<2:0> = 001: Inputs are unsigned, and the output accumulates with previous outputs
- MODE<2:0> = 010: Inputs are signed, and the output does not accumulate
- MODE<2:0> = 011: Inputs are signed, and the output accumulates with previous outputs

In order to perform an Add and Multiply operation, perform the following steps:

- Set the MODE<2:0> bits of the PIDxCON register to one of the four Add/Multiply modes, depending on which form of the calculation is desired, then set the EN bit of the PIDxCON register.
- Write the value of C to the PIDxK1H/L register pair and the value of B to the PIDxSETH/L register pair, as well as the high byte of A to the PIDxINH register.
- Finally, write the low byte of A to the PIDxINL register. This will begin the mathematical operation and set the BUSY bit of the PIDxCON register.
- Either poll the BUSY bit of the PIDxCON register to check for it clearing or wait for the PIDxDIF interrupt to trigger, indicating that the operation has completed.
- 5. Read the PIDxOUT registers for the result of the calculation. In accumulation modes, the PIDxOUT register will hold any previous values added to the current calculation's value. In non-accumulation modes, the PIDxOUT register will just hold the current calculation's value.

These modes can also be used to perform 16-bit addition (by setting the C term in the above equation to 1) or 16-bit multiplication (by setting A or B to 0).

32.3 Interrupts

The PID module has two interrupts, indicated by the interrupt flags PIDxDIF and PIDxEIF in the PIR5 register, and controlled by the interrupt control bits PIDxDIE and PIDxEIE, respectively, in the PIE5 register.

The PIDxDIF interrupt triggers at the successful completion of a calculation, when the BUSY bit of the PIDxCON register goes low.

The PIDxEIF interrupt triggers when there is an error in the PID or multiply and add calculation, specifically an overflow error on the output value.

32.4 Handling Error Overflow

If a calculation causes an overflow of the value in the OUT registers, the value in said registers will roll over and the PIDxEIF interrupt will trigger. In the case of a PID calculation, this indicates that the error has outpaced the PID's capability to correct for the error of the system. In this case, it is recommended to 'saturate' the OUT registers in software whenever the PIDxEIF interrupt is set as part of the Interrupt Service Routine (IRS), as shown in Example 32-1.

EXAMPLE 32-1: HANDLING PID OVERFLOWS

```
//Interrupt service routine
void interrupt ISR(void)
IF (PIR5BITS.PID1EIF==1&&PIE5BITS.PID1EIE==1)
{
    //saturate the PID10UT registers
    PID10UTHH=0xFF;
    PID10UTHL=0xFF;
    PID10UTHL=0xFF;
    PID10UTHH=0xFF;
    //clear the interrupt flag
    PIR5bits.PID1EIF=0;
}
```

32.5 PID Control Registers

Long bit name prefixes for the 16-bit PID peripherals are shown in Table 32-1. Refer to **Section 1.1 "Register and Bit Naming Conventions**" for more information

TABLE 32-1:

Peripheral	Bit Name Prefix
PID1	PID1

REGISTER 32-1: PIDxCON: PID CONFIGURATION REGISTER

R/W-0/0	R/HS/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	BUSY	_	_	_		MODE<2:0>	
bit 7							bit 0

Legend:		
HC = Bit is cleared by hard	dware	HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	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	EN: PID Module Enable b	oit
		<i>/</i>

- 1 = PID module is enabled
- 0 = PID module is disabled

bit 6 BUSY: PID module is currently calculating

- bit 5-3 Unimplemented: Read as '0'
- bit 2-0 MODE<2:0>: PID Mode Control bits
 - 11x = Reserved. Do not use.
 - 101 = PID output is the calculated output (current error plus accumulated previous errors) in 2's complement notation
 - 100 = Reserved. Do not use.
 - 011 = (IN<15:0>+SET<15:0>)*K1<15:0> 2's complement signed inputs, with accumulation
 - 010 = (IN<15:0>+SET<15:0>)*K1<15:0> 2's complement signed inputs, without accumulation
 - 001 = (IN<15:0>+SET<15:0>)*K1<15:0> unsigned inputs, with accumulation
 - 000 = (IN<15:0>+SET<15:0>)*K1<15:0> unsigned inputs, without accumulation

REGISTER 32-2: PIDxINH: PID INPUT HIGH 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
			IN<	15:8>			
bit 7							bit 0
Legend:							
R = Readable bit	= Readable bit W = Writable bit			U = Unimplemented bit, read as '0'			
u - Dit is unshang	ed	x = Bit is unknov	sunknown -n/n = Value at POR		POR and BOR/V	alue at all other F	Resets
u = Bit is unchang	ou						

bit 7-0

IN<15:8>: IN upper eight bits. IN is the 16-bit input from the control system to the PID module

REGISTER 32-3: PIDxINL: PID INPUT LOW 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
			IN<	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	q = Value depends on condition

bit 7-0 IN<7:0>: IN lower eight bits. IN is the 16-bit input from the control system to the PID module

REGISTER 32-4: PIDxSETH: PID SET POINT HIGH 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
	SET<15:8>						
bit 7	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	q = Value depends on condition

bit 7-0 SET<15:8>: SET upper eight bits. SET is the 16-bit user-controlled variable that the input from the control system is compared against to determine the error in the system

REGISTER 32-5: PIDxSETL: PID SET POINT LOW 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
	SET<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	q = Value depends on condition

bit 7-0 SET<7:0>: SET lower eight bits. SET is the 16-bit user-controlled variable that the input from the control system is compared against to determine the error in the system

REGISTER 32-6: PIDxK1H: PID K1 HIGH 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
			K1<	15:8>			
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		t	U = Unimplemented bit, read as '0'				
u = Bit is unchan	anged x = Bit is unknown -n/n = Value at POR and BOR/Value at all oth			alue at all other I	Resets		
'1' = Bit is set		'0' = Bit is cleared q = Value depends on condition					

bit 7-0

K1<15:8>: K1 upper eight bits. K1 is the 16-bit user-controlled coefficient calculated from Kp + Ki + Kd

REGISTER 32-7: PIDxK1L: PID K1 LOW 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
			K1<	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	q = Value depends on condition

bit 7-0 K1<7:0>: K1 lower eight bits. K1 is the 16-bit user-controlled coefficient calculated from Kp + Ki + Kd

REGISTER 32-8: PIDxK2H: PID K2 HIGH 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
	K2<15:8>						
bit 7	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 7-0 K2<15:8>: K2 upper eight bits. K2 is the 16-bit user-controlled coefficient calculated from -(Kp + 2Kd)

REGISTER 32-9: PIDxK2L: PID K2 LOW 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
			K2<	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	q = Value depends on condition

bit 7-0

K2<7:0>: K2 lower eight bits. K2 is the 16-bit user-controlled coefficient calculated from -(Kp + 2Kd)

REGISTER 32-10: PIDxK3H: PID K3 HIGH 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
			K3<	15:8>			
bit 7							bit 0
Legend:							
R = Readable bit	t	W = Writable bit	t	U = Unimpleme	ented bit, read as	'0'	
u = Bit is unchan	iged	x = Bit is unknow	wn	-n/n = Value at	POR and BOR/V	alue at all other I	Resets
'1' = Bit is set		'0' = Bit is cleare	ed	q = Value depe	nds on condition		

bit 7-0

K3<15:8>: K3 upper eight bits. K3 is the 16-bit user-controlled coefficient calculated from Kd

REGISTER 32-11: PIDxK3L: PID K3 LOW 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
			K3<	:7:0>			
bit 7							bit 0
Legend:							
R = Readable bit	:	W = Writable bi	t	U = Unimpleme	ented bit, read as	ʻ0'	
u = Bit is unchan	ged	x = Bit is unkno	wn	-n/n = Value at	POR and BOR/V	alue at all other f	Resets
'1' = Bit is set		'0' = Bit is clear	ed	q = Value depe	ends on condition		

bit 7-0 K3<7:0>: K3 lower eight bits. K3 is the 16-bit user-controlled coefficient calculated from Kd

REGISTER 32-12: PIDxOUTU: PID OUTPUT UPPER REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—		OUT<	35:32>	
bit 7							bit 0
Legend:							

Logona.		
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 7-4 Unimplemented: Read as '0'

bit 3-0 **OUT<35:32>:** Bits <35:32> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-13: PIDxOUTHH: PID OUTPUT HIGH HIGH 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
			OUT<	31:24>			
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 7-0 **OUT<31:24>:** Bits <31:24> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-14: PIDxOUTHL: PID OUTPUT HIGH LOW 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
			OUT<	<23:16>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	bit	U = Unimpler	nented bit, read	l as '0'	
u = Bit is unch	anged	x = Bit is unkno	own	-n/n = Value a	at POR and BO	R/Value at all c	ther Resets
'1' = Bit is set		ʻ0' = Bit is clea	red	q = Value dep	pends on condit	ion	

bit 7-0 **OUT<23:16>** of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-15: PIDxOUTLH: PID OUTPUT LOW HIGH 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
OUT<15:8>							
bit 7							bit 0

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

bit 7-0 **OUT<15:8>:** Bits <15:8> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-16: PIDxOUTLL: PID OUTPUT LOW LOW 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
			OUT	<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	q = Value depends on condition

bit 7-0 **OUT<7:0>:** Bits <7:0> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

Γ.

REGISTER 32-17: PIDxZ1U: PID Z1 UPPER REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	_	—	—	_	—	—	Z116
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 7-1	Unimplemented: Read as '0'
---------	----------------------------

bit 0 Z116: Bit 16 of Z1. In PID mode, Z1 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-18: PIDxZ1H: PID Z1 HIGH 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
Z1<15:8>							
bit 7							bit 0

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

bit 7-0 Z1<15:8>: Bits <15:8> of Z1. In PID mode, Z1 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-19: PIDxZ1L: PID Z1 LOW 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
Z1<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	q = Value depends on condition

bit 7-0 Z1<7:0>: Bits <7:0> of Z1. In PID mode, Z1 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-20: PIDxZ2U: PID Z2 UPPER REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	_		_			—	Z216
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 7-1	Unimplemented: Read as '0'
---------	----------------------------

bit 0 **Z216:** Bit 16 of Z2. In PID mode, Z2 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-21: PIDxZ2H: PID Z2 HIGH 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
Z2<15:8>							
bit 7							

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 7-0 **Z2<15:8>:** Bits <15:8> of Z2. In PID mode, Z2 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-22: PIDxZ2L: PID Z2 LOW 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
Z2<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	q = Value depends on condition

bit 7-0 **Z2<7:0>:** Bits <7:0> of Z2. In PID mode, Z2 is the value of the error (IN minus SET) from the previous iteration of the PID control loop.

REGISTER 32-23: PIDxACCU: PID ACCUMULATOR UPPER REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	-		ACC<34:32>	
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 7-3 Unimplemented: Read as '0'

.

bit 2-0 ACC<34:32>: Bits <34:32> of ACC. ACC is the accumulator register in which all of the multiplier results for the PID are accumulated before being written to the output.

REGISTER 32-24: PIDxACCHH: PID ACCUMULATOR HIGH HIGH 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
			ACC<	31:24>			
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 7-0 **ACC<31:24>:** Bits <31:24> of ACC. ACC is the accumulator register in which all of the multiplier results for the PID are accumulated before being written to the output.

REGISTER 32-25: PIDxACCHL: PID ACCUMULATOR HIGH LOW 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
ACC<23:16>							
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 7-0 ACC<23:16>: Bits <23:16> of ACC. ACC is the accumulator register in which all of the multiplier results for the PID are accumulated before being written to the output.

REGISTER 32-26: PIDxACCLH: PID ACCUMULATOR LOW HIGH 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
			ACC	<15:8>			
bit 7							bit 0
Legend:							
R = Readable	eadable bit W = Writable bit		bit	U = Unimpler	nented bit, read	l as '0'	
u = Bit is unch	anged	x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other I			ther Resets
'1' = Bit is set		'0' = Bit is cleared q = Value depends on condition				ion	

bit 7-0 **ACC<15:8>:** Bits <15:8> of ACC. ACC is the accumulator register in which all of the multiplier results for the PID are accumulated before being written to the output.

REGISTER 32-27: PIDxACCLL: PID ACCUMULATOR LOW LOW 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
	ACC<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	q = Value depends on condition

bit 7-0 ACC<7:0>: Bits <7:0> of ACC. ACC is the accumulator register in which all of the multiplier results for the PID are accumulated before being written to the output.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PID1ACCU	—		—				ACC<34:32>	>	493
PID1ACCHH		ACC<31:24>							493
PID1ACCHL				ACC<	23:16>				493
PID1ACCLH				ACC<	:15:8>				494
PID1ACCLL				ACC	<7:0>				494
PID1CON	EN	BUSY	—	—	—		MODE<2:0>	•	485
PID1INH				IN<1	5:8>				486
PID1INL				IN<	7:0>				486
PID1K1H				K1<′	15:8>				487
PID1K1L				K1<	7:0>				487
PID1K2H				K2<′	15:8>				487
PID1K2L				K2<	7:0>				487
PID1K3H				K3<′	15:8>				488
PID1K3L		K3<7:0>						488	
PID10UTU	—	-	-	_		OUT<	35:32>		489
PID10UTHH	OUT<31:24>						489		
PID10UTHL	OUT<23:16>						490		
PID10UTLH		OUT<15:8>						490	
PID10UTLL	OUT<7:0>						490		
PID1SETH				SET<	15:8>				486
PID1SETL				SET	<7:0>				486
PID1Z1U	_	—	—		—	_	—	Z116	491
PID1Z1H				Z1<′	5:8>				491
PID1Z1L				Z1<	7:0>				491
PID1Z2U	—	—	—	—	—	—	—	Z216	492
PID1Z2H				Z2<′	5:8>			•	492
PID1Z2L				Z2<	7:0>				492
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	102
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	107

TABLE 32-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE PID M
--

33.0 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 ICSPTM refer to the "*PIC12(L)F1612/PIC16(L)F161X Memory Programming Specification*" (DS40001720).

33.1 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.

33.2 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 ICSP Low-Voltage Programming Entry mode 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.

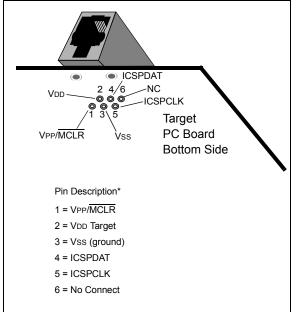
If low-voltage programming is enabled (LVP = 1), the $\overline{\text{MCLR}}$ Reset function is automatically enabled and cannot be disabled. See **Section6.5** "MCLR" for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

33.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSPTM header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6-connector) configuration. See Figure 33-1.

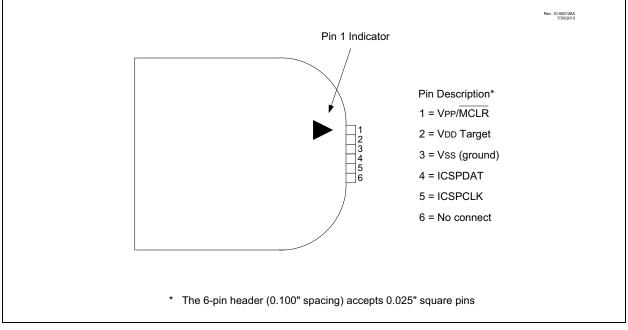
FIGURE 33-1: ICD RJ-11 STYLE CONNECTOR INTERFACE



Another connector often found in use with the PICkit[™] programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 33-2.

PIC16(L)F1615/9

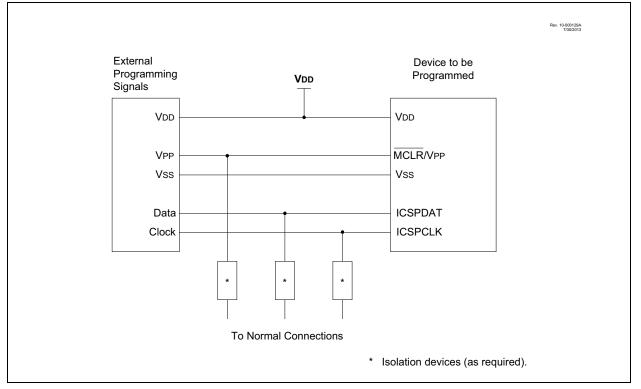
FIGURE 33-2: PICkit[™] PROGRAMMER STYLE CONNECTOR INTERFACE



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 33-3 for more information.

FIGURE 33-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



34.0 INSTRUCTION SET SUMMARY

Each 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 34-3 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)

TABLE 34-1: 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
x	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 34-2: 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

 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.

34.1 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.

FIGURE 34-1: GENERAL FORMAT FOR INSTRUCTIONS

13 8 OPCODE	3 7 d	6	f (FILE #)	0
d = 0 for destina	-		. (== ")	
d = 0 for destina d = 1 for destina f = 7-bit file regis	tion f	dres	S	
Bit-oriented file regis	ster op 0 9	erat 7	t ions 6	0
OPCODE	b (Bl	T #)	f (FILE #)
b = 3-bit bit addr f = 7-bit file regis		dres	S	
Literal and control o	peratio	ons		
General				
13	8	7	L. (1:t 1)	0
OPCODE			k (literal)	
k = 8-bit immedia	ate vali	ue		
CALL and GOTO instru	ctions	only		
<u>13 11 10</u>)			0
OPCODE		k ((literal)	
k = 11-bit immed	iate va	lue		
MOUT D instruction only				
MOVLP instruction only 13		7 (3	0
OPCODE			k (literal)	
k = 7-bit immedia	ate valu	Je		
k = 7-bit immedia	ate valı	le		
MOVLB instruction only		he		
MOVLB instruction only		Je	5 4	0
MOVLB instruction only 13 OPCODE	/		5 4 k (literal	
MOVLB instruction only	/			
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only	/ ate valu	Je		
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13	/	Je	k (literal)
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only	/ ate valu	Je)
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13	/ ate valu 9 8	Je	k (literal)
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE	/ ate valu 9 8	Je	k (literal)
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedia	/ ate valu 9 8	Je	k (literal)
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedia FSR Offset instruction	ate valu 9 8 ate val	ue	k (literal	0
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedia FSR Offset instruction 13	ate valu 9 8 ate valu s 7	ue 6 n	k (literal k (literal) 5	0
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedia FSR Offset instruction 13 OPCODE n = appropriate	ate valu 9 8 ate valu 1s 7 FSR ate val	ue 6 n	k (literal k (literal) 5	0
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedi FSR Offset instruction 13 OPCODE n = appropriate k = 6-bit immedi FSR Increment instruct	ate valu 9 8 ate valu 1s 7 FSR ate val	ue 6 n	k (literal) k (literal) 5 k (literal)) 0 0)) 0
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedia FSR Offset instruction 13 OPCODE n = appropriate k = 6-bit immedia FSR Increment instruct 13	ate valu 9 8 ate valu ate valu s FSR ate valu stions FSR	ue 6 n	k (literal) k (literal) 5 k (literal)) 0 0)) 0
MOVLB instruction only 13 OPCODE k = 5-bit immedia BRA instruction only 13 OPCODE k = 9-bit immedi FSR Offset instruction 13 OPCODE n = appropriate k = 6-bit immedi FSR Increment instruct 13 OPCODE n = appropriate n = appropriate	ate valu 9 8 ate valu ate valu s FSR ate valu stions FSR	ue 6 n	k (literal) k (literal) 5 k (literal)) 0 0)) 0

Mnemonic, Operands		-		14-Bit Opcode				Status	
		Description	Cycles	MSb			LSb	Affected	Notes
		BYTE-ORIENTED FILE	REGISTER OPE	RATIC	ONS				
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		0000		Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	z	2
DECF	f, d	Decrement f	1	00		dfff		z	2
INCF	f, d	Increment f	1	00		dfff			2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff			2
MOVF	f, d	Move f	1	00		dfff			2
MOVWF	f.	Move W to f	1	00		1fff		-	2
RLF	f, d	Rotate Left f through Carry	1	00		dfff		с	2
RRF	f, d	Rotate Right f through Carry	1	00		dfff		c	2
SUBWF	f, d	Subtract W from f	1	00		dfff		-	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11		dfff			2
	'		1					C, DC, Z	2
SWAPF XORWF	f, d f, d	Swap nibbles in f Exclusive OR W with f	1	00	1110	dfff		z	2
JURWE	1, U	BYTE ORIENTED			0110	dfff	LLLL	2	2
	fd		1		1011	dfff	ffff		1 0
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011				1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
		BIT-ORIENTED FILE R	EGISTER OPER	RATION	NS			-	
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	SKIP OPERATIO	NS				•	
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
		1	PERATIONS	1					
ADDLW	k	Add literal and W	1	11	1110	kkkk		C, DC, Z	
ANDLW	k	AND literal with W	1	11		kkkk		Z	
IORLW	k	Inclusive OR literal with W	1	11		kkkk		Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k			
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk			
MOVLW	k	Move literal to W	1	11	0000	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 34-3: ENHANCED MID-RANGE 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			14-Bit	Opcode	•	Status	Notes
				MSb			LSb	Affected	Notes
		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 OPERA	TIONS	•					
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 OPT	IMIZED						
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					kkkk		
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	1nmm	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec	1	00	0000	0001	kkkk		2, 3
		modifier, mm							
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk			2

TABLE 34-3: ENHANCED MID-RANGE 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.

34.2 Instruction Descriptions

ADDFSR	Add Literal to FSRn
Syntax:	[label] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0, 1]
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:	[label] 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

register 'f'.

►	register f	->	С	

ADD W and CARRY bit to f

Syntax:	[label] 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	
Syntax:	[<i>label</i>]BRA label [<i>label</i>]BRA \$+k	
Operands:	-256 \leq label - PC + 1 \leq 255 -256 \leq k \leq 255	
Operation:	$(PC) + 1 + k \rightarrow PC$	
Status Affected:	None	
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.

BTFSS	Bit Test f, Skip if Set
Syntax:	[label]BTFSS f,b
Operands:	$0 \le f \le 127$ $0 \le b < 7$
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:	[<i>label</i>] 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.

BSF	Bit Set f
Syntax:	[<i>label</i>]BSF f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	$1 \rightarrow (f \le b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.
Dperands: Dperation: Status Affected:	$0 \le f \le 127$ $0 \le b \le 7$ $1 \rightarrow (f \le b)$ None

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:	[<i>label</i>] COMF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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.

Clear W

Syntax:	[label] CLRW
Operands:	None
Operation:	$\begin{array}{l} \text{O0h} \rightarrow (\text{W}) \\ \text{l} \rightarrow \text{Z} \end{array}$
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[label] CLRWDT
Operands:	None
Operation:	$00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler,

	$1 \rightarrow \overline{\text{TO}}$
	$1 \rightarrow PD$
Status Affected:	TO, PD
Description:	CLRWDT instruction resets the Watch- dog Timer. It also resets the prescaler of the WDT. Status bits TO and PD are 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 regis- ter 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[label] DECFSZ 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 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:	$k \rightarrow PC<10:0>$ PCLATH<6:3> \rightarrow 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.

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'.

Increment f, Skip if 0

[label] INCFSZ f,d

(f) + 1 \rightarrow (destination), skip if result = 0

The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result

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

result is placed in the W register.

is placed back in register 'f'.

 $\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$

None

INCFSZ

Syntax:

Operands:

Operation:

Status Affected:

Description:

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

1	С	-	register f	← 0
				•

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



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

instruction.

MOVF	Move f	MOVIW	Move INDFn to	o W
Syntax: Operands: Operation:	$\begin{bmatrix} label \end{bmatrix} MOVF f,d$ $0 \le f \le 127$ $d \in [0,1]$ $(f) \rightarrow (dest)$	Syntax:	[label] MOVIW [label] MOVIW [label] MOVIW [label] MOVIW [label] MOVIW	FSRn FSRn++ FSRn
Status Affected: Description:	Z The contents of register f is moved to a destination dependent upon the	Operands:	$n \in [0,1]$ $mm \in [00,01, 10]$ $-32 \le k \le 31$), 11]
	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.	Operation:	INDFn → W Effective address • FSR + 1 (prein • FSR - 1 (preduced) • FSR + k (related)	ecrement)
Words:	1			ne FSR value will be
Cycles:	1		either: • FSR + 1 (all ir	crements)
Example:	MOVF FSR, 0		FSR - 1 (all decrements)	
	After Instruction		 Unchanged 	
	W = value in FSR register Z = 1	Status Affected:	Z	
		Mode	Syntax	mm
		Preincrement	++FSRn	00
		Predecrement	FSRn	01
		Postincrement	FSRn++	10
		Postdecrement	FSRn	11
		Description:	between W and registers (INDFn move, the pointe pre/post increme	s used to move data one of the indirect). Before/after this r (FSRn) is updated by enting/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.

MOVLB	Move literal to BSR
Syntax:	[<i>label</i>]MOVLB k
Operands:	$0 \leq k \leq 31$
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	ΜΟΥΨΙ	Move W to INDFn	
Syntax: Operands:	[<i>label</i>]]MOVLP_k 0 ≤ k ≤ 127	Syntax:	[<i>label</i>] MOVWI ++FSRn [<i>label</i>] MOVWIFSRn	
Operation:	$k \rightarrow PCLATH$		[label] MOVWI FSRn++	
Status Affected:	None		[<i>label</i>] MOVWI FSRn [<i>label</i>] MOVWI k[FSRn]	
Description:	The 7-bit literal 'k' is loaded into the PCLATH register.	Operands:	$n \in [0,1]$ $mm \in [00,01, 10, 11]$ $-32 \le k \le 31$	
MOVLW	Move literal to W	Operation:	$W \rightarrow \text{INDFn}$ Effective address is determined by	
Syntax:	[<i>label</i>] MOVLW k		 FSR + 1 (preincrement) FSR - 1 (predecrement) 	
Operands:	$0 \le k \le 255$		• FSR + k (relative offset)	
Operation:	$k \rightarrow (W)$		After the Move, the FSR value will be either: • FSR + 1 (all increments)	
Status Affected:	None			
Description:	The 8-bit literal 'k' is loaded into W reg- ister. The "don't cares" will assemble as		 FSR - 1 (all decrements) Unchanged 	
	ʻ0 's .	Status Affected:	None	
Words:	1			
Cycles:	1	Mode	Syntax mm	
Example:	MOVLW 0x5A	Preincrement	++FSRn 00	
	After Instruction W = 0x5A	Predecrement	FSRn 01	
	W = 0x5A	Postincrement	FSRn++ 10	
MOVWF	Move W to f	Postdecrement	FSRn 11	
Syntax:	[<i>label</i>] MOVWF f			
Operands:	$0 \le f \le 127$	Description:	This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this	
Operation:	$(W) \rightarrow (f)$			
Status Affected:	None		move, the pointer (FSRn) is updated by	
Description:	Move data from W register to register 'f'.		pre/post incrementing/decrementing it. Note: The INDFn registers are not	
Words:	1		physical registers. Any instruction that	
Cycles:	1		accesses an INDFn register actually accesses the register at the address	
Example:	MOVWF OPTION_REG		specified by the FSRn.	
	Before Instruction OPTION_REG = 0xFF W = 0x4F After Instruction OPTION_REG = 0x4F W = 0x4F		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

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-cycl instruction.Words:1Cycles:2Example:RETFIEAfter Interrupt PC = TOS GIE = 1RETLWReturn with literal in WSyntax:[label] RETLW kOperands: $0 \le k \le 255$ Operation: $k \to (W)$; TOS \to PCStatus Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE ; W contains tak ; offset value		Return from Interrupt
Operation:TOS \rightarrow PC, $1 \rightarrow$ GIEStatus Affected:NoneDescription:Return from Interrupt. Stack is POP 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-cycl instruction.Words:1Cycles:2Example:RETFTE After Interrupt PC = TOS GIE = 1RETLWReturn with literal in WSyntax:[label] RETLW kOperands:0 < k < 255	Syntax:	[label] RETFIE
$1 \rightarrow GIE$ Status Affected: None Description: Return from Interrupt. Stack is POP 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-cycl instruction. Words: 1 Cycles: 2 Example: RETFIE After Interrupt PC = TOS GIE = 1 $RETLW$ Return with literal in W Syntax: [<i>label</i>] RETLW k Operands: 0 ≤ k ≤ 255 Operation: k → (W); TOS → PC Status Affected: None Description: The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction. Words: 1 Cycles: 2 Example: CALL TABLE ; W contains tak ; offset value ; W now has table valu TABLE After PC ; W = offset RETLW k1 ; Begin table RETLW k2 ; .	Operands:	None
Description:Return from Interrupt. Stack is POP 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-cycl instruction.Words:1Cycles:2Example:RETFIEAfter Interrupt PC = TOS GIE = 1RETLWReturn with literal in WSyntax:[/abe/]RETLWReturn with literal in WSyntax:[/abe/]RETLWReturn with literal in WSyntax:[/abe/]RETLWReturn with literal in WSyntax:[/abe/]Return with literal in WSyntax:[/abe/]Coperands:0 $\leq k \leq 255$ Operands:0 $\leq k \leq 255$ Operands:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:(CALL TABLE ;W contains tak ;offset value <trr>.<</trr>	Operation:	
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:1Cycles:2Example:RETFIEAfter Interrupt PC = TOS GIE = 1RETLWReturn with literal in WSyntax:[/abe/] RETLW kOperands: $0 \le k \le 255$ Operation: $k \to (W)$; TOS \rightarrow PCStatus Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE ;W contains tak ;offset value··ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Status Affected:	None
Cycles:2Example:RETFIEAfter Interrupt PC = TOS GIE = 1RETLWReturn with literal in WSyntax: $[label]$ RETLW kOperands: $0 \le k \le 255$ Operation: $k \rightarrow (W);$ TOS \rightarrow PCStatus Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE ;W contains tak ;offset valueTABLE.ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Description:	setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle
Example:RETFIEAfter Interrupt $PC = TOS$ $GIE = 1$ RETLWReturn with literal in WSyntax:[label] RETLW kOperands: $0 \le k \le 255$ Operation: $k \rightarrow (W);$ TOS \rightarrow PCStatus Affected:NoneDescription:The W register is loaded with the 8- 	Words:	1
After Interrupt $PC = TOS$ $GIE = 1$ RETLWReturn with literal in W Syntax:[/abe/] RETLW kOperands: $0 \le k \le 255$ Operation: $k \to (W)$; $TOS \to PC$ Status Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE ;W contains tak ;offset valueTABLE.ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Cycles:	2
PC=TOS GIE RETLWReturn with literal in W Syntax: $[label]$ RETLW kOperands: $0 \le k \le 255$ Operation: $k \rightarrow (W)$; TOS \rightarrow PCStatus Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE;W contains tak ;offset valueTABLE.ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Example:	RETFIE
Syntax:[label] RETLW kOperands: $0 \le k \le 255$ Operation: $k \to (W);$ TOS $\to PC$ Status Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE;W contains tak ;offset valueTABLE.ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;		PC = TOS
Operands: $0 \le k \le 255$ Operation: $k \to (W);$ TOS \to PCStatus Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE;W contains tak ;offset valueTABLE.ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	RETLW	
Operation: $k \rightarrow (W);$ TOS \rightarrow PCStatus Affected:NoneDescription:The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.Words:1Cycles:2Example:CALL TABLE;W contains take ;offset valueTABLE.ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Syntax:	[<i>label</i>] RETLW k
$TOS \rightarrow PC$ Status Affected: None Description: The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction. Words: 1 Cycles: 2 Example: CALL TABLE;W contains tak ;offset value ;W now has table valu TABLE TABLE ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; .	Operands:	$0 \leq k \leq 255$
Description: The W register is loaded with the 8- literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction. Words: 1 Cycles: 2 Example: CALL TABLE;W contains take ;offset value ;W now has table value ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Operation:	. ,
<pre>literal 'k'. The program counter is loaded from the top of the stack (th return address). This is a 2-cycle instruction. Words: 1 Cycles: 2 Example: CALL TABLE;W contains tak ;offset value . ;W now has table value ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; .</pre>	Status Affected:	None
Cycles: 2 Example: CALL TABLE;W contains tak ;offset value · ;W now has table valu TABLE · ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; ·	Description:	loaded from the top of the stack (the return address). This is a 2-cycle
Example: CALL TABLE;W contains tak ;offset value ;W now has table valu TABLE ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Words:	1
;offset value ;W now has table valu ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;	Cycles:	2
• ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; •	<u>Example:</u>	CALL TABLE;W contains table ;offset value • ;W now has table value
• RETLW kn ; End of table	FABLE	RETLW k1 ;Begin table RETLW k2 ; • •
Before Instruction W = 0x07 After Instruction		
W = value of k8		W = 0x07

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.

Load OPTION_REG Register

RESET	Software Reset
Syntax:	[label] RESET
Operands:	None
Operation:	Execute a device Reset. Resets the \overline{RI} flag of the PCON register.
Status Affected:	None
Description:	This instruction provides a way to execute a hardware Reset by soft- ware.

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.

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

→ C →	Register f	-►

		SLEEP	Enter Sleep mode
RLF	Rotate Left f through Carry	Syntax:	[label] SLEEP
Syntax:	[label] RLF f,d	Operands:	None
Operands:	$\begin{array}{l} 0\leq f\leq 127\\ d\in [0,1] \end{array}$	Operation:	$00h \rightarrow WDT$, $0 \rightarrow WDT$ prescaler,
Operation:	See description below		$1 \rightarrow \overline{\overline{\text{TO}}},$
Status Affected:	С		$0 \rightarrow PD$
Description:	The contents of register 'f' are rotated	Status Affected:	TO, PD
	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 stored back in register 'f'.	Description:	The power-down Status bit, \overline{PD} is cleared. Time-out Status bit, \overline{TO} is set. Watchdog Timer and its pres- caler are cleared. The processor is put into Sleep mode with the oscillator stopped.
Words:	1		
Cycles:	1		
Example:	RLF REG1,0		
	Before Instruction REG1 = 1110 0110 C = 0 -<		

SUBLW	Subtract W	from literal
Syntax:	[label] SL	IBLW k
Operands:	$0 \leq k \leq 255$	
Operation:	$k -(W) \to (W$)
Status Affected:	C, DC, Z	
Description:	plement meth	er is subtracted (2's com- nod) from the 8-bit literal t is placed in the W regis-
	C = 0	W > k
	C = 1	$W \leq k$

DC = 0

DC = 1

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

C = 0	W > f
C = 1	$W \leq f$
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

W<3:0> > k<3:0>

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

SUBWFB	Subtract W from f with Borrow
Syntax:	SUBWFB f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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:	$0 \le f \le 127$ $d \in [0,1]$			
Operation:	$(f<3:0>) \rightarrow (destination<7:4>),$ $(f<7:4>) \rightarrow (destination<3:0>)$			
Status Affected:	None			
Description:	The upper and lower nibbles of regis- ter 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.			

TRIS	Load TRIS Register with W
Syntax:	[label] TRIS f
Operands:	$5 \leq f \leq 7$
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.

XORLW	Exclusive OR literal with W		
Syntax:	[label] 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.		

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'.			

35.0 ELECTRICAL SPECIFICATIONS

35.1 Absolute Maximum Ratings^(†)

Ambient temperature under bias
Storage temperature
Voltage on pins with respect to Vss
on Vod pin
PIC16F1615/9
PIC16LF1615/90.3V to +4.0V
on MCLR pin
on all other pins
Maximum current
on Vss pin ⁽¹⁾
-40°C \leq Ta \leq +85°C
+85°C \leq TA \leq +125°C
on VDD pin ⁽¹⁾
$-40^{\circ}C \le TA \le +85^{\circ}C$
+85°C \leq TA \leq +125°C
Sunk by any standard I/O pin 50 mA
Sourced by any standard I/O pin 50 mA
Sunk by any High Current I/O pin 100 mA
Sourced by any High Current I/O pin 100 mA
Clamp current, Iк (VPIN < 0 or VPIN > VDD)
Total power dissipation ⁽²⁾

Note 1: Maximum current rating requires even load distribution across I/O pins. Maximum current rating may be limited by the device package power dissipation characterizations, see Table 35-6: "Thermal Characteristics" to calculate device specifications.

2: Power dissipation is calculated as follows: PDIS = VDD x {IDD $-\Sigma$ IOH} + Σ {(VDD - VOH) x IOH} + Σ (VOI x IOL).

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause 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.

35.2 Standard Operating Conditions

The standard operating conditions for any device are defined as: $V \text{DDMIN} \leq V \text{DD} \leq V \text{DDMAX}$ Operating Voltage: Operating Temperature: TA MIN \leq TA \leq TA MAX VDD — Operating Supply Voltage⁽¹⁾ PIC16LF1615/9 PIC16F1615/9 TA — Operating Ambient Temperature Range Industrial Temperature TA MIN.....--40°C **Extended Temperature** Ta MIN.....--40°C

Note 1: See Parameter D001, DS Characteristics: Supply Voltage.

FIGURE 35-1: VOLTAGE FREQUENCY GRAPH, -40°C \leq Ta \leq +125°C, PIC16F1615/9 ONLY

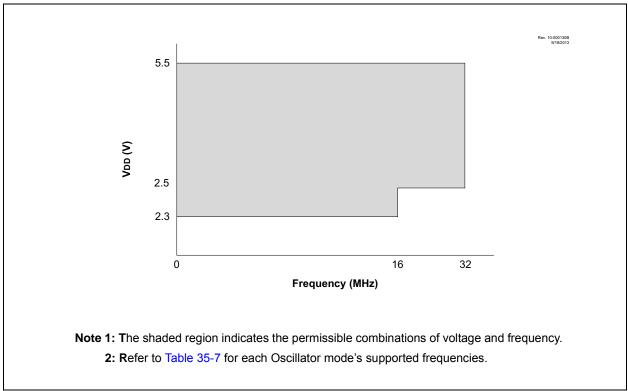
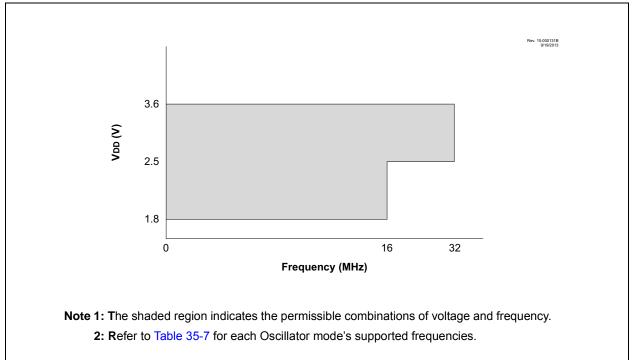


FIGURE 35-2: VOLTAGE FREQUENCY GRAPH, -40°C \leq Ta \leq +125°C, PIC16LF1615/9 ONLY



35.3 DC Characteristics

TABLE 35-1: SUPPLY VOLTAGE

PIC16LF1615/9		Standard Operating Conditions (unless otherwise stated)					
PIC16F1	615/9						
Param. No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
D001	Vdd	Supply Voltage		•			
			VDDMIN		VDDMAX		
			1.8 2.5	_	3.6 3.6	V V	Fosc ≤ 16 MHz Fosc ≤ 32 MHz
D001			2.3	_	5.5	V	$Fosc \le 16 \text{ MHz}$
0001			2.5	_	5.5	v	$Fosc \le 32 \text{ MHz}$
D002*	Vdr	RAM Data Retention Voltage ⁽¹⁾	1				•
			1.5	_	_	V	Device in Sleep mode
D002*			1.7	_	_	V	Device in Sleep mode
D002A*	VPOR	Power-on Reset Release Voltage	(2)				
			—	1.6	_	V	
D002A*			—	1.6		V	
D002B* VPORR* Power-on Reset Rearm Voltage ⁽²⁾							
			—	0.8		V	
D002B*			—	1.5		V	
D003	VFVR	Fixed Voltage Reference Voltage					- -
				1.024	_	V	$-40^{\circ}C \leq TA \leq +85^{\circ}C$
D003				1.024	—	V	$-40^{\circ}C \leq TA \leq +85^{\circ}C$
D003A	VADFVR	FVR Gain Voltage Accuracy for ADC					
			-4	_	+4	%	$1x \text{ VFVR}, \text{ VDD} \geq 2.5 \text{ V}$
					• 4	70	$2x VFVR, VDD \ge 2.5V$
D003A			-5	_	+5	%	1x VFVR, VDD \ge 2.5V 2x VFVR, VDD \ge 2.5V
							$4x VFVR, VDD \ge 4.75V$
D003B	VCDAFVR	FVR Gain Voltage Accuracy for C	omparato	or/DAC			
						01	1x VFVR, VDD \geq 2.5V
			-4	_	+4	%	$2x$ VFVR, VDD ≥ 2.5 V
D003B			-7	_	+7	%	1x VFVR, VDD \geq 2.5V
					. 1	70	$2x \text{ VFVR, VDD} \ge 2.5\text{V}$ $4x \text{ VFVR, VDD} \ge 4.75\text{V}$
D004*	SVDD	VDD Rise Rate ⁽²⁾					$\neg v = v r, v \cup v \ge \neg .10 v$
0004	0000		0.05			V/ms	Ensures that the Power-on Reset
			0.00			v/113	signal is released properly.
D004*			0.05	_		V/ms	Ensures that the Power-on Reset
							signal is released properly.

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: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: See Figure 35-3, POR and POR REARM with Slow Rising VDD.



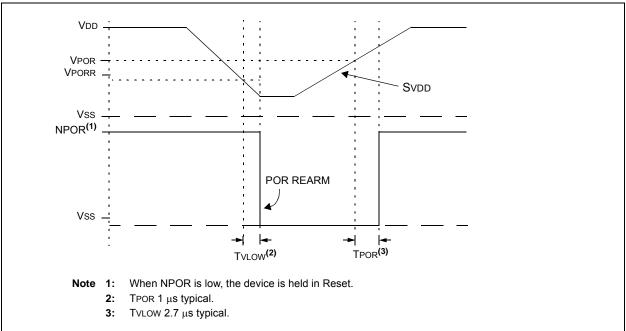


TABLE 35-2: SUPPLY CURRENT (IDD)^(1,2)

1615/9	Stand	Standard Operating Conditions (unless otherwise stated)								
615/9										
Param. Device		Turnt	Mox		Conditions					
Characteristics	win.	турт	wax.	Units	Vdd	Note				
	—	30	90	μA	1.8	Fosc = 1 MHz,				
	—	55	110	μA	3.0	External Clock (ECM), Medium-Power mode				
	_	65	120	μA	2.3	Fosc = 1 MHz,				
		85	150	μA	3.0	External Clock (ECM),				
		115	200	μA	5.0	Medium-Power mode				
	_	115	260	μA	1.8	Fosc = 4 MHz,				
	—	210	380	μA	3.0	External Clock (ECM), Medium-Power mode				
		180	310	μA	2.3	Fosc = 4 MHz,				
		240	410	μA	3.0	External Clock (ECM),				
		295	520	μA	5.0	Medium-Power mode				
	615/9 Device	615/9 Device Min	Device Characteristics Min. Typ† — 30 — 55 — 65 — 85 — 115 — 115 — 210 — 180 — 240 — 240	Device Characteristics Min. Typ† Max. — 30 90 — 55 110 — 65 120 — 85 150 — 115 200 — 115 260 — 180 310 — 240 410	Device Characteristics Min. Typ† Max. Units — 30 90 μA — 55 110 μA — 65 120 μA — 85 150 μA — 115 200 μA — 115 260 μA — 115 260 μA — 115 260 μA — 210 380 μA — 180 310 μA — 240 410 μA	Device Characteristics Min. Typ† Max. Units VDD — 30 90 μA 1.8 — 55 110 μA 3.0 — 65 120 μA 2.3 — 85 150 μA 3.0 — 115 200 μA 1.8 — 210 380 μA 3.0 — 180 310 μA 2.3 — 240 410 μA 3.0				

These parameters are characterized but not tested.

Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance † only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to Vss; MCLR = VDD; WDT disabled.

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

PIC16LF	1615/9	Stand	Standard Operating Conditions (unless otherwise stated)								
PIC16F1	615/9										
Param.	Device	Min.	Typ†	Max.	Units		Conditions				
No.	Characteristics		וקעי	Wax.	01113	Vdd	Note				
D015		—	9.6	36	μA	1.8	Fosc = 31 kHz,				
		—	16.2	60	μA	3.0	LFINTOSC, -40°C ≤ TA ≤ +85°C				
D015			39	84	μA	2.3	Fosc = 31 kHz,				
		_	45	90	μA	3.0	LFINTOSC, -40°C ≤ Ta ≤ +85°C				
		—	51	108	μA	5.0	-40 C \leq IA \leq +85 C				
D016		—	215	360	μA	1.8	Fosc = 500 kHz,				
		—	275	480	μA	3.0	HFINTOSC				
D016		_	270	450	μA	2.3	Fosc = 500 kHz,				
		_	300	500	μA	3.0	HFINTOSC				
		—	350	620	μA	5.0					
D017*		_	410	800	μA	1.8	Fosc = 8 MHz,				
		—	630	1200	μA	3.0	HFINTOSC				
D017*		_	530	950	μA	2.3	Fosc = 8 MHz,				
		_	660	1300	μA	3.0	HFINTOSC				
		_	730	1400	μA	5.0					
D018		_	600	1200	μA	1.8	Fosc = 16 MHz,				
			970	1850	μA	3.0	HFINTOSC				
D018		_	780	1500	μA	2.3	Fosc = 16 MHz,				
		_	1000	1900	μA	3.0	HFINTOSC				
		—	1090	2100	μA	5.0					

TABLE 35-2:	SUPPLY CURRENT	(IDD) ^(1,2) ((CONTINUED)
-------------	----------------	--------------------------	-------------

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to Vss; MCLR = VDD; WDT disabled.

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

PIC16LF	1615/9	Stand	ard Opera	ating Cor	nditions (u	inless ot	herwise stated)	
PIC16F1	615/9							
Param.	Device	Min.	Typ†	Max.	Units	Conditions		
No.			Vdd	Note				
D019		—	1.6	5.0	mA	3.0	Fosc = 32 MHz, HFINTOSC	
		—	1.9	6.0	mA	3.6		
D019			1.6	5.0	mA	3.0	Fosc = 32 MHz, HFINTOSC	
		—	1.9	6.0	mA	5.0		
D020A		_	1.6	5.0	mA	3.0	Fosc = 32 MHz,	
		—	1.9	6.0	mA	3.6	External Clock (ECH), High-Power mode	
D020A		_	1.6	5.0	mA	3.0	Fosc = 32 MHz,	
		_	1.9	6.0	mA	5.0	External Clock (ECH), High-Power mode	
D020B		_	6	16	μA	1.8	Fosc = 32 kHz,	
		_	8	22	μA	3.0	External Clock (ECL), Low-Power mode	
D020B		_	13	43	μA	2.3	Fosc = 32 kHz,	
		_	15	55	μA	3.0	External Clock (ECL), Low-Power mode	
		—	16	57	μA	5.0	Low-Power mode	
D020C			19	40	μA	1.8	Fosc = 500 kHz,	
			32	60	μA	3.0	External Clock (ECL), Low-Power mode	
D020C		_	31	60	μA	2.3	Fosc = 500 kHz,	
		_	38	90	μA	3.0	External Clock (ECL),	
		_	44	100	μA	5.0	Low-Power mode	

TABLE 35-2: SUPPLY CURRENT (IDD) ^(1,2) (COI	NTINUED)	
--	----------	--

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VSS; MCLR = VDD; WDT disabled.

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

PIC16LF1	615/9		•	ditions: (ep Mode	unless oth	nerwise s	stated)	
PIC16F16	15/9	Low-Po	ower Sle	ep Mode,	VREGPM	= 1		
Param.	Device Characteristics	Min.	Typ†	Max.	Max.	Units		Conditions
No.			1961	+85°C	+125°C	onito	Vdd	Note
D022	Base IPD		0.020	1.0	8.0	μA	1.8	WDT, BOR, FVR disabled, all
		—	0.025	2.0	9.0	μA	3.0	Peripherals inactive
D022	Base IPD		0.25	3.0	10	μA	2.3	WDT, BOR, FVR disabled, all
			0.30	4.0	12	μA	3.0	Peripherals inactive, Low-Power Sleep mode
		—	0.40	6.0	15	μA	5.0	Low-Fower Sleep mode
D022A	Base IPD	_	9.8	16	18	μA	2.3	WDT, BOR, FVR disabled, all
		—	10.3	18	20	μA	3.0	Peripherals inactive,
		_	11.5	21	26	μA	5.0	Normal-Power Sleep mode, VREGPM = 0
D023		_	0.26	2.0	9.0	μA	1.8	WDT Current
		_	0.44	3.0	10	μA	3.0	
D023		_	0.43	6.0	15	μA	2.3	WDT Current
			0.53	7.0	20	μA	3.0	
		_	0.64	8.0	22	μA	5.0	
D023A		_	15	28	30	μA	1.8	FVR Current
		_	18	30	33	μA	3.0	
D023A		_	18	33	35	μA	2.3	FVR Current
			19	35	37	μA	3.0	1
			20	37	39	μA	5.0	
D024		_	6.0	17	20	μA	3.0	BOR Current
D024		_	7.0	17	30	μA	3.0	BOR Current
		—	8.0	20	40	μA	5.0	1
D24A		_	0.1	4.0	10	μA	3.0	LPBOR Current
D24A		_	0.35	5.0	14	μA	3.0	LPBOR Current
		_	0.45	8.0	17	μA	5.0	1
D026			0.11	1.5	9.0	μA	1.8	ADC Current (Note 3),
			0.12	2.7	10	μA	3.0	No conversion in progress
D026		—	0.30	4.0	11	μA	2.3	ADC Current (Note 3),
			0.35	5.0	13	μA	3.0	No conversion in progress
			0.45	8.0	16	μA	5.0	
D026A*		_	250	_	_	μA	1.8	ADC Current (Note 3),
		—	250	—	—	μA	3.0	Conversion in progress
D026A*		_	280	—	—	μA	2.3	ADC Current (Note 3),
		_	280	—	—	μA	3.0	Conversion in progress
		_	280	_	_	μA	5.0	1

TABLE 35-3: POWER-DOWN CURRENTS (IPD)^(1,2)

* 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.

Legend: TBD = To Be Determined

Note 1: The peripheral ∆ current can be determined by subtracting the base 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 Vss.

3: ADC clock source is FRC.

TABLE 35-3: POWER-DOWN CURRENTS (IPD)^(1,2) (CONTINUED)

PIC16LF16		Operating Conditions: (unless otherwise stated) Low-Power Sleep Mode								
PIC16F161	5/9	Low-Po	Low-Power Sleep Mode, VREGPM = 1							
Param.		Min	Truck	Max.	Max.	11	Conditions			
No.	Device Characteristics	Min. Ty	Тур†	+85°C	+125°C	Units	Vdd	Note		
D027			7	22	25	μA	1.8	Comparator,		
		_	8	23	27	μA	3.0	CxSP = 0		
D027		—	17	35	37	μA	2.3	Comparator,		
		_	18	37	38	μA	3.0	CxSP = 0		
		_	19	38	40	μA	5.0			

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Legend: TBD = To Be Determined

Note 1: The peripheral ∆ current can be determined by subtracting the base 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 Vss.

3: ADC clock source is FRC.

TABLE 35-4: I/O PORTS

Standard	d Operat	ing Conditions (unless otherw	ise stated)				
Param. No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
	VIL	Input Low Voltage					
		I/O PORT:					
D030		with TTL buffer		_	0.8	V	$4.5V \leq V\text{DD} \leq 5.5V$
D030A			—	—	0.15 VDD	V	$1.8V \leq V\text{DD} \leq 4.5V$
D031		with Schmitt Trigger buffer	_	—	0.2 Vdd	V	$2.0V \leq V\text{DD} \leq 5.5V$
D032		MCLR	_	_	0.2 VDD	V	
	Viн	Input High Voltage I/O PORT:					
D040		with TTL buffer	2.0	_	_	V	$4.5V \leq V\text{DD} \leq 5.5V$
D040A			0.25 VDD + 0.8	—	—	V	$1.8V \le V\text{DD} \le 4.5V$
D041		with Schmitt Trigger buffer	0.8 VDD	_	_	V	$2.0V \leq V\text{DD} \leq 5.5V$
D042		MCLR	0.8 VDD	_	_	V	
	lı∟	Input Leakage Current ⁽¹⁾					
D060		I/O Ports	—	± 5	± 125	nA	$Vss \le VPIN \le VDD$, Pin at high-impedance, 85°C
			—	± 5	± 1000	nA	$Vss \le VPIN \le VDD$, Pin at high-impedance, 125°C
D061		MCLR ⁽³⁾	_	± 50	± 200	nA	$Vss \le VPIN \le VDD$, Pin at high-impedance, 85°C
	IPUR	Weak Pull-up Current					
D070*			25	100	200	μΑ	VDD = 3.3V, VPIN = VSS
			25	140	300	μΑ	VDD = 5.0V, VPIN = VSS
	Vol	Output Low Voltage ⁽³⁾				•	
D080		I/O Ports	_	_	0.6	V	IOL = 8.0 mA, VDD = 5.0V IOL = 6.0 mA, VDD = 3.3V IOL = 1.8 mA, VDD = 1.8V
D080A		High Drive I/O ⁽¹⁾	_	2.5V	_	V	IOL = 100 mA, VDD = 5.0V
	Voн	Output High Voltage ⁽³⁾					
D090		I/O Ports	Vdd - 0.7	_		V	IOH = 3.5 mA, VDD = 5.0V IOH = 3.0 mA, VDD = 3.3V IOH = 1.0 mA, VDD = 1.8V
D090A		High Drive I/O ⁽¹⁾	_	2.5V	_	V	IOL = 100 mA, VDD = 5.0V
D101A*	CIO	All I/O pins			50		
D101A*	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.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Excluding OSC2 in CLKOUT mode.

Param. No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
		Program Memory Programming Specifications					
D110	VIHH	Voltage on MCLR/VPP pin	8.0	_	9.0	V	(Note 2)
D111	IDDP	Supply Current during Programming	_	—	10	mA	
D112	VBE	VDD for Bulk Erase	2.7	_	VDDMAX	V	
D113	VPEW	VDD for Write or Row Erase	VDDMIN	_	VDDMAX	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	10K	—	_	E/W	-40°C ≤ TA ≤ +85°C (Note 1)
D122	VPRW	VDD for Read/Write	VDDMIN	_	VDDMAX	V	
D123	Tiw	Self-timed Write Cycle Time	_	2	2.5	ms	
D124	Tretd	Characteristic Retention	—	40	_	Year	Provided no other specifications are violated
D125	EHEFC	High-Endurance Flash Cell	100K		—	E/W	$0^{\circ}C \le TA \le +60^{\circ}C$, lower byte last 128 addresses

TABLE 35-5: MEMORY PROGRAMMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)

† 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 35-6: THERMAL CHARACTERISTICS

u Operating	conditions (unless otherwise stated)			
Sym.	Characteristic	Тур.	Units	Conditions
θJA	Thermal Resistance Junction to Ambient	62.2	°C/W	20-pin DIP package
		77.7	°C/W	20-pin SOIC package
		87.3	°C/W	20-pin SSOP package
		43	°C/W	20-pin QFN 4X4mm package
θJC	Thermal Resistance Junction to Case	27.5	°C/W	20-pin DIP package
		23.1	°C/W	20-pin SOIC package
		31.1	°C/W	20-pin SSOP package
		5.3	°C/W	20-pin QFN 4X4mm package
TJMAX	Maximum Junction Temperature	150	°C	
PD	Power Dissipation	_	W	PD = PINTERNAL + PI/O
PINTERNAL	Internal Power Dissipation	_	W	PINTERNAL = IDD x VDD ⁽¹⁾
Pı/o	I/O Power Dissipation		W	$PI/O = \Sigma (IOL * VOL) + \Sigma (IOH * (VDD - VOH))$
Pder	Derated Power		W	Pder = PDmax (Tj - Ta)/θja ⁽²⁾
	Bym. θJA θJC ΠJMAX PD PINTERNAL PI/O	θJA Thermal Resistance Junction to Ambient θJC Thermal Resistance Junction to Case TJMAX Maximum Junction Temperature PD Power Dissipation PINTERNAL Internal Power Dissipation PI/O I/O Power Dissipation	Sym.CharacteristicTyp.θJAThermal Resistance Junction to Ambient62.277.787.387.343θJCThermal Resistance Junction to Case27.523.131.131.15.3TJMAXMaximum Junction Temperature150PDPower Dissipation—PINTERNALInternal Power Dissipation—PI/OI/O Power Dissipation—	Sym.CharacteristicTyp.UnitsθJAThermal Resistance Junction to Ambient62.2°C/W77.7°C/W77.7°C/W87.3°C/W43°C/W0 JCThermal Resistance Junction to Case27.5°C/W0 JCThermal Resistance Junction to Case27.5°C/W1 JI°C/W31.1°C/W1 JI°C/W5.3°C/W1 JII°C/W5.3°C/W1 JII°C/W5.3°C/W1 JIIPower DissipationWPINTERNALInternal Power DissipationWPI/OI/O Power DissipationW

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; TJ = Junction Temperature

35.4 AC Characteristics

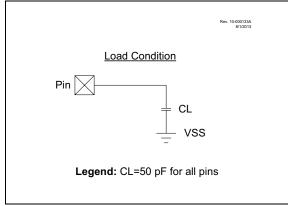
Timing Parameter Symbology has been created with one of the following formats:

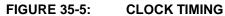
1. TppS2ppS

2. TppS

T				
-				
F	Frequency	Т	Time	
Lowerc	case letters (pp) and their meanings:			
рр				
сс	CCP1	osc	CLKIN	
ck	CLKOUT	rd	RD	
CS	CS	rw	RD or WR	
di	SDIx	sc	SCKx	
do	SDO	SS	SS	
dt	Data in	tO	TOCKI	
io	I/O PORT	t1	T1CKI	
mc	MCLR	wr	WR	
Upperc	case 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 35-4: LOAD CONDITIONS





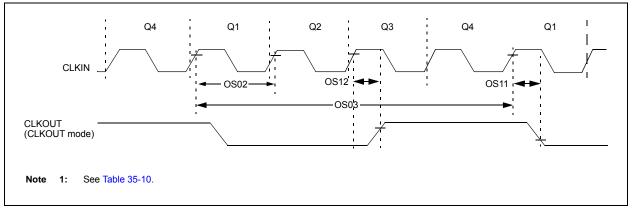


TABLE 35-7: CLOCK OSCILLATOR TIMING REQUIREMENTS

	-	-	-										
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	—	32	MHz	External Clock (ECH)						
OS02	Tosc	External CLKIN Period ⁽¹⁾	31.25		×	ns	External Clock (EC)						
OS03	Тсү	Instruction Cycle Time ⁽¹⁾	200	TCY	DC	ns	Tcy = 4/Fosc						

Standard Operating Conditions (unless otherwise stated)

* 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 CLKIN pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

TABLE 35-8: 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 ⁽¹⁾	_	_	16.0	_	MHz	(Note 2)
OS09	LFosc	Internal LFINTOSC Frequency	—	_	31	_	kHz	(Note 3)
OS10	Twarm	HFINTOSC Wake-up from Sleep Start-up Time	—		5	15	μS	
		LFINTOSC Wake-up from Sleep Start-up Time	—	_	0.5	_	ms	

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:To ensure these oscillator frequency tolerances, VDD and Vss must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

2: See Figure 35-6: "HFINTOSC Frequency Accuracy over Device VDD and Temperature",

3: See Figure 36-43: "LFINTOSC Frequency over VDD and Temperature, PIC16LF1615/9 Only", and Figure 36-44: "LFINTOSC Frequency over VDD and Temperature, PIC16F1615/9 Only".



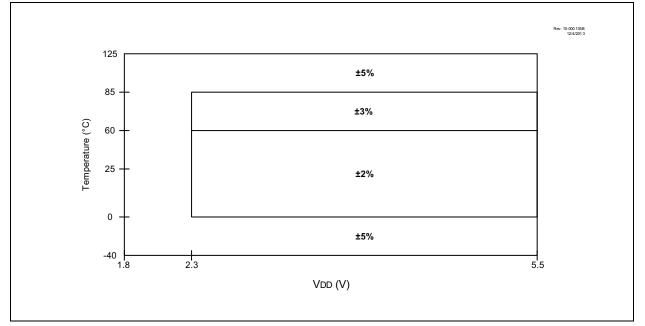


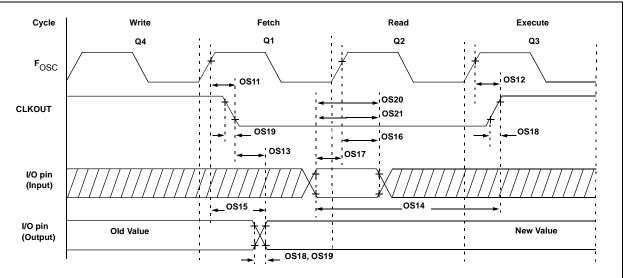
TABLE 35-9: PLL CLOCK TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)								
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions	
F10	Fosc	Oscillator Frequency Range	4	—	8	MHz		
F11	Fsys	On-Chip VCO System Frequency	16	_	32	MHz		
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms		
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	_	+0.25%	%		

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.



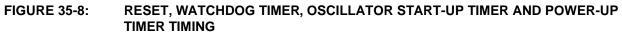


Standard Operating Conditions (unless otherwise stated)								
Param. No.	Sym.	Characteristic	Min.		Max.	Units	Conditions	
OS11	TosH2ckL	Fosc↑ to CLKOUT↓ ⁽¹⁾	_	_	70	ns	$3.3V \le V\text{DD} \le 5.0V$	
OS12	TosH2ckH	Fosc↑ to CLKOUT↑ ⁽¹⁾	_	_	72	ns	$3.3V \le V\text{DD} \le 5.0V$	
OS13	TckL2ioV	CLKOUT↓ to 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	$3.3V \leq V\text{DD} \leq 5.0V$	
OS16	TosH2iol	Fosc [↑] (Q2 cycle) to Port input invalid (I/O in setup time)	50	—	_	ns	$3.3V \le V\text{DD} \le 5.0V$	
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns		
OS18*	TioR	Port output rise time		40 15	72 32	ns	$\begin{array}{l} VDD\texttt{D}\texttt{=}1.8V\\ 3.3V \leq VDD \leq 5.0V \end{array}$	
OS19*	TioF	Port output fall time	—	28 15	55 30	ns	$\begin{array}{l} VDD \mbox{=} \mbox{1.8V} \\ 3.3V \leq VDD \leq 5.0V \end{array}$	
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.

 \dagger Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

Note 1: Measurements are taken in EXTRC mode where CLKOUT output is 4 x Tosc.



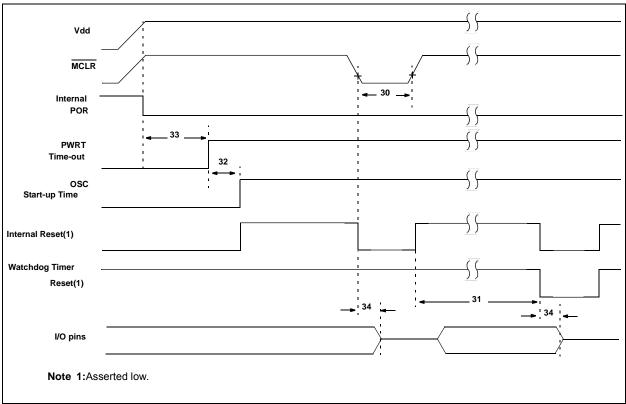


TABLE 35-11:RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER
AND BROWN-OUT RESET PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions	
30	ТмсL	MCLR Pulse Width (low)	2	_	—	μS		
31	TWDTLP	Low-Power Watchdog Timer Time-out Period	10	16	27	ms	VDD = 3.3V-5V, 1:512 Prescaler used	
32	Tost	Oscillator Start-up Timer Period ⁽¹⁾	_	1024	—	Tosc		
33*	TPWRT	Power-up Timer Period	40	65	140	ms	PWRTE = 0	
34*	Tioz	I/O high-impedance from MCLR Low or Watchdog Timer Reset	_	—	2.0	μS		
35	VBOR	Brown-out Reset Voltage ⁽²⁾	2.55	2.70	2.85	V	BORV = 0	
			2.35 1.80	2.45 1.90	2.58 2.05	V V	BORV = 1 (PIC16F1615/9) BORV = 1 (PIC16LF1615/9)	
36*	VHYST	Brown-out Reset Hysteresis	0	25	60	mV	$-40^{\circ}C \leq TA \leq +85^{\circ}C$	
37*	TBORDC	Brown-out Reset DC Response Time	1	16	35	μS	$V \text{DD} \leq V \text{BOR}$	
38	VLPBOR	Low-Power Brown-Out Reset Voltage	1.8	2.1	2.5	V	LPBOR = 1	

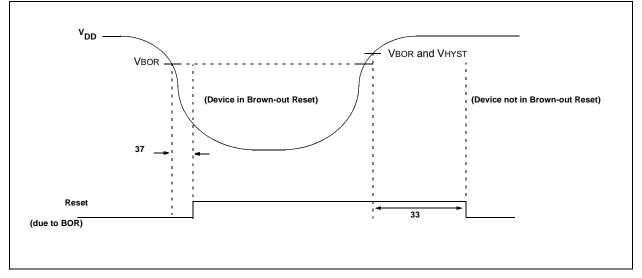
* These parameters are characterized but not tested.

† 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: By design, the Oscillator Start-up Timer (OST) counts the first 1024 cycles, independent of frequency.

2: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.







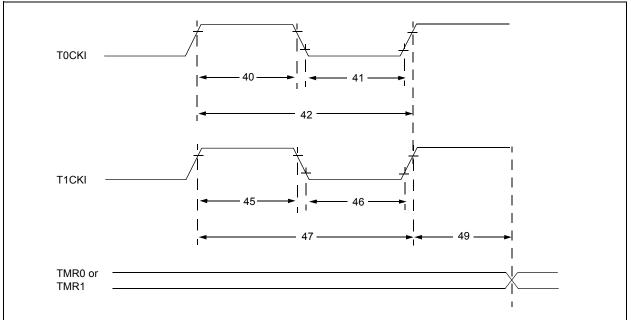


TABLE 35-12: TIMER0 AND TIMER1 EXTERNAL C	LOCK REQUIREMENTS
---	-------------------

Standa	rd Operating (Conditions (u	nless otherwis	e 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*	TT0L	T0CKI Low F	ulse Width	No Prescaler	0.5 TCY + 20	_		ns	
				With Prescaler	10	_		ns	
42*	TT0P	T0CKI Period	od		Greater of: 20 or <u>Tcy + 40</u> N	_	_	ns	N = prescale value
45*	T⊤1H	T1CKI High Time	Synchronous, No Prescaler		0.5 Tcy + 20	_	_	ns	
			Synchronous, with Prescaler		15	_		ns	
			Asynchronous		30	_		ns	
46*	TT1L	T1CKI Low	Synchronous, No Prescaler Synchronous, with Prescaler		0.5 TCY + 20	_		ns	
		Time			15	_		ns	
			Asynchronous		30	_		ns	
47*	TT1P	T1CKI Input Period	Synchronous		Greater of: 30 or <u>Tcy + 40</u> N	_	—	ns	N = prescale value
			Asynchronous		60	_	_	ns	
49*	TCKEZTMR1	Delay from E Increment	xternal Clock Edge 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 t tested.

TABLE 35-13: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS^(1,2,3)

Operating Conditions (unless othe	rwise stated)
-----------------------------------	---------------

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	±2.5	LSb	VREF = 3.0V
AD05	Egn	Gain Error	_	±1	±2.0	LSb	VREF = 3.0V
AD06	VREF	Reference Voltage	1.8	_	VDD	V	VREF = (VRPOS - VRNEG) (Note 4)
AD07	VAIN	Full-Scale Range	Vss	_	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	_	10	kΩ	Can go higher if external $0.01\mu F$ 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 t tested.

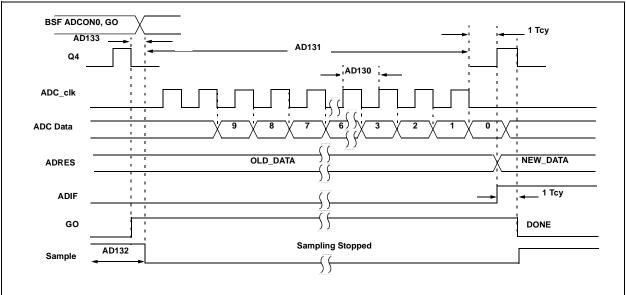
Note 1:Total Absolute Error includes integral, differential, offset and gain errors.

2: The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

3: See Section 36.0 "DC and AC Characteristics Graphs and Charts" for operating characterization.

4: ADC VREF is selected by ADPREF<0> bit.







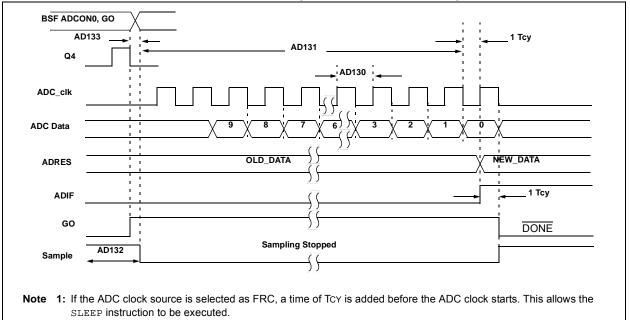


TABLE 35-14: ADC CONVERSION REQUIREMENTS

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	_	μS		
AD133*	Тнср	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.

TABLE 35-15: COMPARATOR SPECIFICATIONS⁽¹⁾

Operating Conditions (unless otherwise stated)

Param. No.	Sym.	Characteristics	Min.	Тур.	Max.	Units	Comments
CM01	Vioff	Input Offset Voltage	—	±7.5	±60	mV	CxSP = 1, Vicm = VDD/2
CM02	Vicm	Input Common Mode Voltage	0	—	VDD	V	
CM03	CMRR	Common Mode Rejection Ratio	_	50	_	dB	
CM04A		Response Time Rising Edge	_	400	800	ns	CxSP = 1
CM04B	Tresp ⁽²⁾	Response Time Falling Edge		200	400	ns	CxSP = 1
CM04C	Tresp. 7	Response Time Rising Edge	_	1200	_	ns	CxSP = 0
CM04D	1	Response Time Falling Edge	_	550	_	ns	CxSP = 0
CM05*	Tmc2ov	Comparator Mode Change to Output Valid	—	—	10	μS	
CM06	CHYSTER	Comparator Hysteresis	_	25	_	mV	CxHYS = 1, CxSP = 1

* These parameters are characterized but not tested.

Note 1: See Section 36.0 "DC and AC Characteristics Graphs and Charts" for operating characterization.

2: Response time measured with one comparator input at VDD/2, while the other input transitions from Vss to VDD.

TABLE 35-16: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS⁽¹⁾

Operating Conditions (unless otherwise stated)

*

Param. No.	Sym.	Characteristics	Min.	Тур.	Max.	Units	Comments	
DAC01*	CLSB	Step Size		VDD/256		V		
DAC02*	CACC	Absolute Accuracy	—	_	± 1.5	LSb		
DAC03*	CR	Unit Resistor Value (R)	—	—	_	Ω		
DAC04*	CST	Settling Time ⁽²⁾		—	10	μS		

These parameters are characterized but not tested.

Note 1: See Section 36.0 "DC and AC Characteristics Graphs and Charts" for operating characterization.

2: Settling time measured while DACR<4:0> transitions from '0000' to '1111'.

TABLE 35-17: ZERO CROSS PIN SPECIFICATIONS

Operating Conditions (unless otherwise stated)

VDD = 3.0V, TA = 25°C								
Param. No.	Sym.	Characteristics	Min.	Тур.	Max.	Units	Comments	
ZC01	ZCPINV	Voltage on Zero Cross Pin	—	0.75	_	V		
ZC02	ZCSRC	Source current	—	-300	-600	μA		
ZC03	ZCSNK	Sink current	—	300	600	μA		
ZC04	Zcisw	Response Time Rising Edge	—	1	_	μS		
		Response Time Falling Edge	—	1	_	μS		
ZC05	ZCOUT	Response Time Rising Edge	_	1	_	μS		
		Response Time Falling Edge	_	1	_	μS		

These parameters are characterized but not tested.

36.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

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

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

Unless otherwise noted, all graphs apply to both the L and LF devices.

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

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

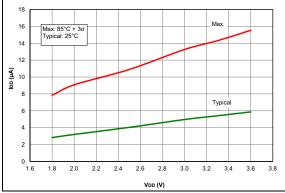


FIGURE 36-1: IDD, EC Oscillator LP Mode, Fosc = 32 kHz, PIC16LF1615/9 Only.

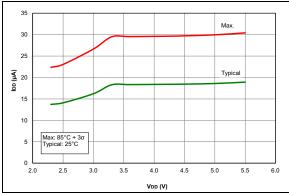


FIGURE 36-2: IDD, EC Oscillator LP Mode, Fosc = 32 kHz, PIC16F1615/9 Only.

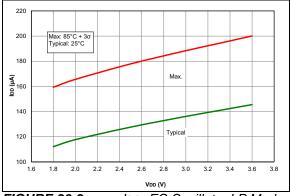


FIGURE 36-3: IDD, EC Oscillator LP Mode, Fosc = 500 kHz, PIC16LF1615/9 Only.

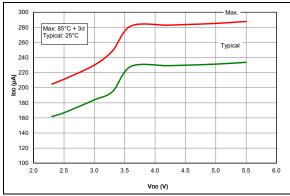


FIGURE 36-4: IDD, EC Oscillator LP Mode, Fosc = 500 kHz, PIC16F1615/9 Only.

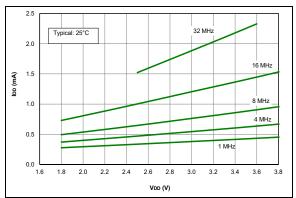


FIGURE 36-5: IDD Typical, EC Oscillator MP Mode, PIC16LF1615/9 Only.

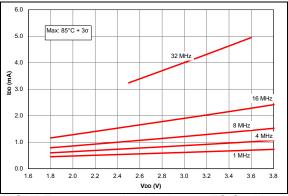


FIGURE 36-6: IDD Maximum, EC Oscillator MP Mode, PIC16LF1615/9 Only.

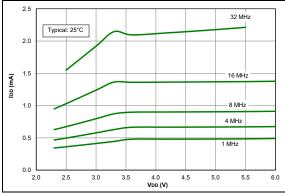


FIGURE 36-7: IDD Typical, EC Oscillator MP Mode, PIC16F1615/9 Only.

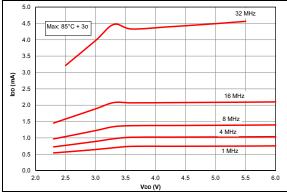


FIGURE 36-8: IDD Maximum, EC Oscillator MP Mode, PIC16F1615/9 Only.

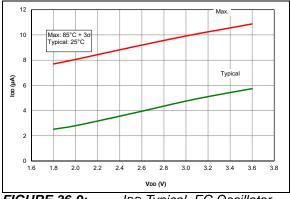


FIGURE 36-9: IDD Typical, EC Oscillator HP Mode, PIC16LF1615/9 Only.

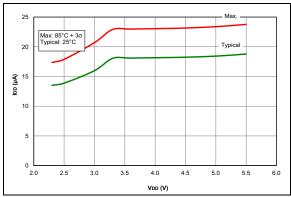


FIGURE 36-10: IDD Maximum, EC Oscillator HP Mode, PIC16LF1615/9 Only.

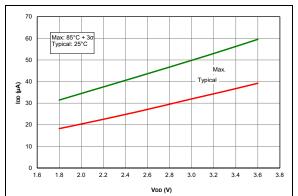


FIGURE 36-11: IDD Typical, EC Oscillator HP Mode, PIC16F1615/9 Only.

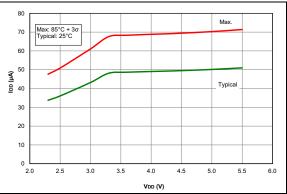
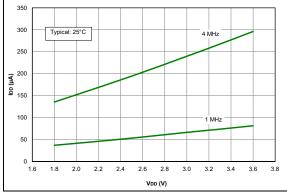
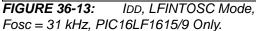


FIGURE 36-12: IDD Maximum, EC Oscillator HP Mode, PIC16F1615/9 Only.





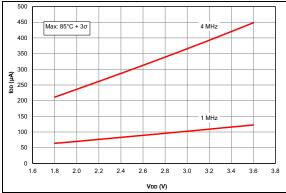


FIGURE 36-14: IDD, LFINTOSC Mode, Fosc = 31 kHz, PIC16F1615/9 Only.

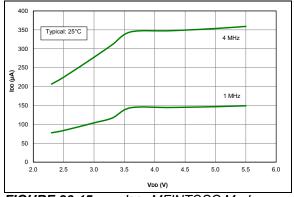


FIGURE 36-15: IDD, MFINTOSC Mode, Fosc = 500 kHz, PIC16LF1615/9 Only.

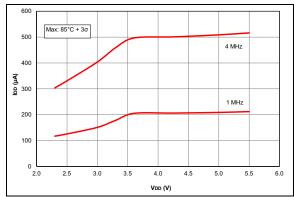


FIGURE 36-16: IDD, MFINTOSC Mode, Fosc = 500 kHz, PIC16F1615/9 Only.

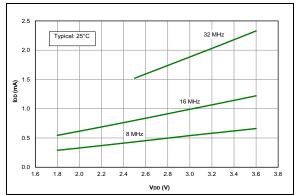


FIGURE 36-17: IDD Typical, HFINTOSC Mode, PIC16LF1615/9 Only.

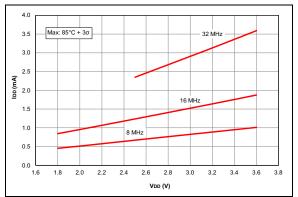


FIGURE 36-18: IDD Maximum, HFINTOSC Mode, PIC16LF1615/9 Only.

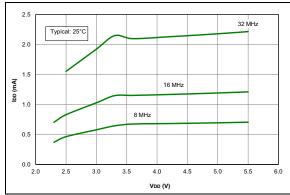


FIGURE 36-19: IDD Typical, HFINTOSC Mode, PIC16F1615/9 Only.

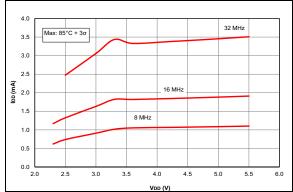


FIGURE 36-20: IDD Maximum, HFINTOSC Mode, PIC16F1615/9 Only.

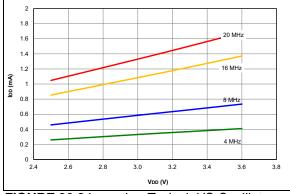


FIGURE 36-21: IDD Typical, HS Oscillator, 25°C, PIC16LF1615/9 Only.

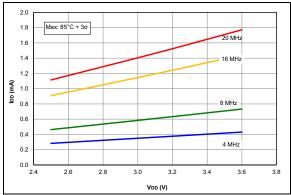


FIGURE 36-22: IDD Maximum, HS Oscillator, PIC16LF1615/9 Only.

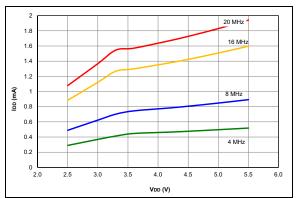


FIGURE 36-23: IDD Typical, HS Oscillator, 25°C, PIC16F1615/9 Only.

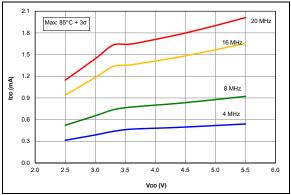


FIGURE 36-24: IDD Maximum, HS Oscillator, PIC16F1615/9 Only.

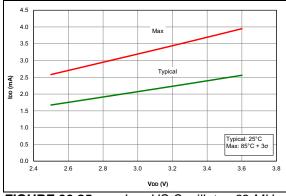


FIGURE 36-25: IDD, HS Oscillator, 32 MHz (8 MHz + 4x PLL), PIC16LF1615/9 Only.

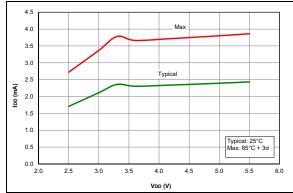


FIGURE 36-26: IDD, HS Oscillator, 32 MHz (8 MHz + 4x PLL), PIC16F1615/9 Only.

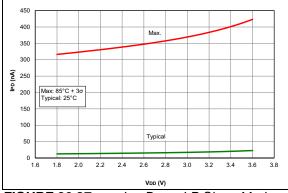


FIGURE 36-27: IPD Base, LP Sleep Mode, PIC16LF1615/9 Only.

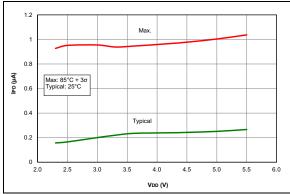


FIGURE 36-28: IPD Base, LP Sleep Mode (VREGPM = 1), PIC16F1615/9 Only.

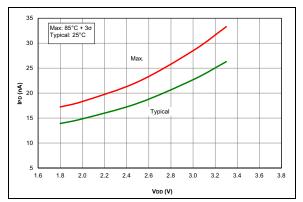


FIGURE 36-29: IPD, Fixed Voltage Reference (FVR), PIC16LF1615/9 Only.

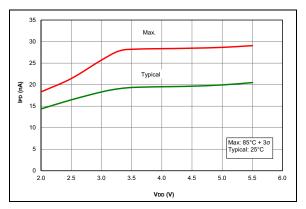
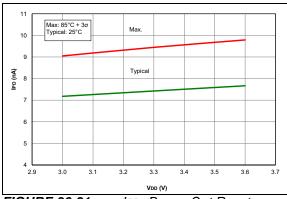


FIGURE 36-30: IPD, Fixed Voltage Reference (FVR), PIC16F1615/9 Only.





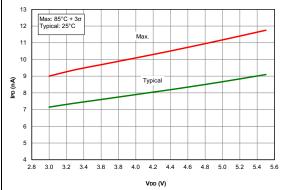


FIGURE 36-32: IPD, Brown-Out Reset (BOR), BORV = 1, PIC16F1615/9 Only.

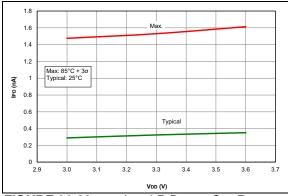


FIGURE 36-33: IPD, LP Brown-Out Reset (LPBOR = 0), PIC16LF1615/9 Only.

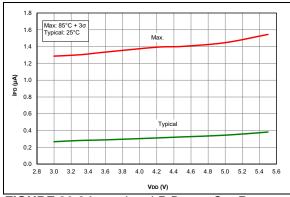


FIGURE 36-34: IPD, LP Brown-Out Reset (LPBOR = 0), PIC16F1615/9 Only.

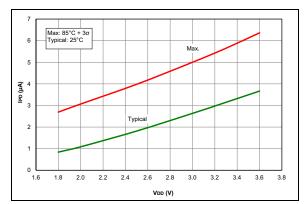


FIGURE 36-35: IPD, Timer1 Oscillator, Fosc = 32 kHz, PIC16LF1615/9 Only.

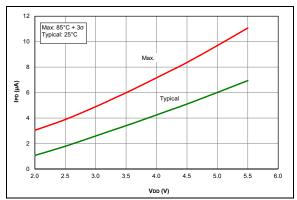


FIGURE 36-36: IPD, Timer1 Oscillator, Fosc = 32 kHz, PIC16F1615/9 Only.

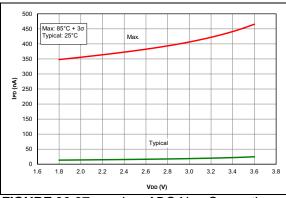


FIGURE 36-37: IPD, ADC Non-Converting, PIC16LF1615/9 Only.

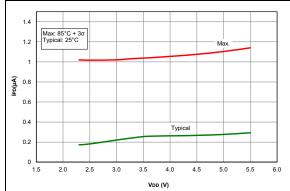


FIGURE 36-38: IPD, ADC Non-Converting, PIC16F1615/9 Only.

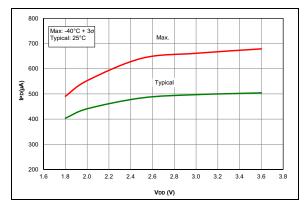


FIGURE 36-39: IPD, Comparator, NP Mode (CxSP = 1), PIC16LF1615/9 Only.

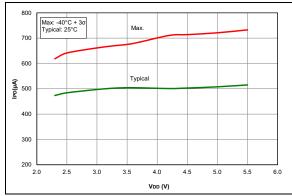


FIGURE 36-40: IPD, Comparator, NP Mode (CxSP = 1), PIC16F1615/9 Only.

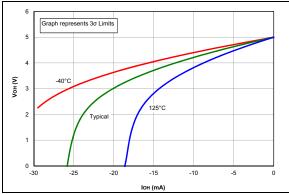


FIGURE 36-41: VOH vs. IOH Over Temperature, VDD = 5.0V, PIC16F1615/9 Only.

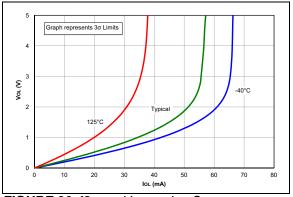


FIGURE 36-42: VOL vs. IOL Over Temperature, VDD = 5.0V, PIC16F1615/9 Only.

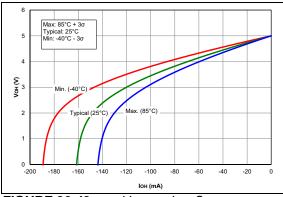


FIGURE 36-43: Voн vs. Ioн Over Temperature for High Drive Pins, Vod = 5.0V, PIC16F1615/9 Only.

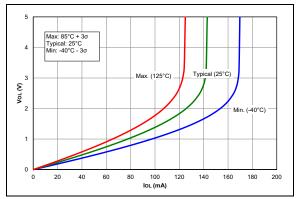


FIGURE 36-44: Vol. vs. Iol. Over Temperature for High Drive Pins, VDD = 5.0V, PIC16F1615/9 Only.

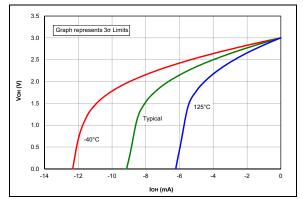


FIGURE 36-45: VOH vs. IOH Over Temperature, VDD = 3.0V.

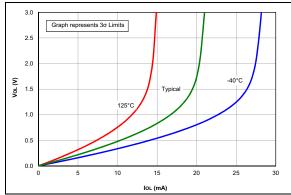


FIGURE 36-46: Vol. vs. Iol. Over Temperature, VDD = 3.0V.

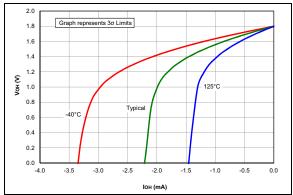


FIGURE 36-47:VOH vs. IOH OverTemperature, VDD = 1.8V, PIC16LF1615/9 Only.

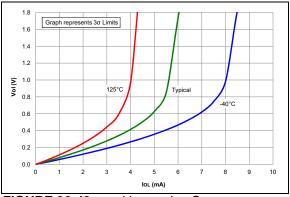
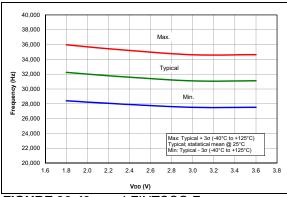
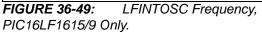


FIGURE 36-48: Vol. vs. Iol. Over Temperature, VDD = 1.8V, PIC16LF1615/9 Only.





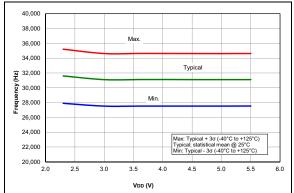


FIGURE 36-50: LFINTOSC Frequency, PIC16F1615/9 Only.

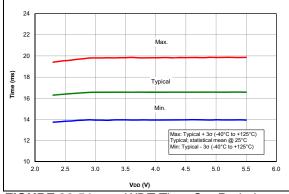


FIGURE 36-51: WDT Time-Out Period, PIC16F1615/9 Only.

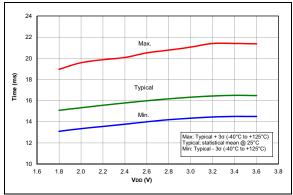


FIGURE 36-52: WDT Time-Out Period, PIC16LF1615/9 Only.

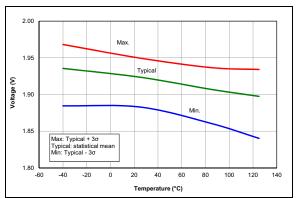


FIGURE 36-53: Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16LF1615/9 Only.

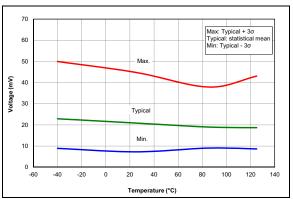


FIGURE 36-54: Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16LF1615/9 Only.

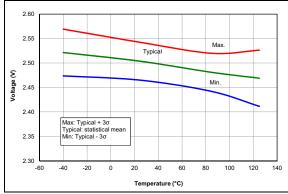


FIGURE 36-55: Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16F1615/9 Only.

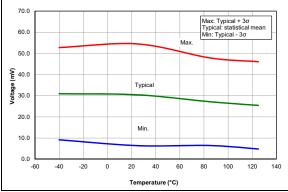


FIGURE 36-56: Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16F1615/9 Only.

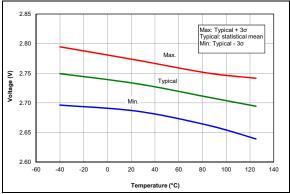


FIGURE 36-57: Brown-Out Reset Voltage, High Trip Point (BORV = 0).

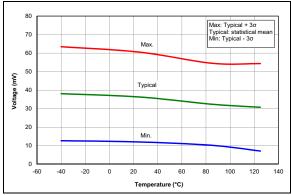


FIGURE 36-58: Brown-Out Reset Hysteresis, High Trip Point (BORV = 0).

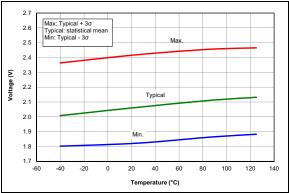
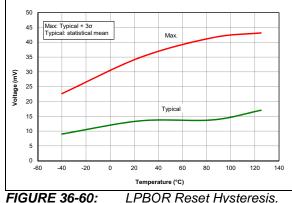


FIGURE 36-59: LPBOR Reset Voltage.



LPBOR Reset Hysteresis.

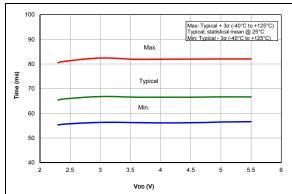


FIGURE 36-61: PWRT Period, PIC16F1615/9 Only.

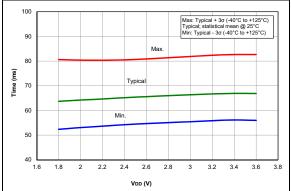
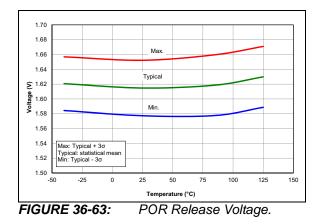


FIGURE 36-62: PWRT Period, PIC16LF1615/9 Only.



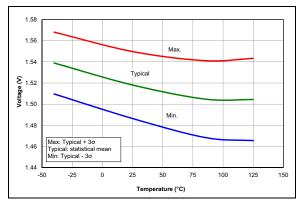


FIGURE 36-64: POR Rearm Voltage, NP Mode (VREGPM1 = 0), PIC16F1615/9 Only.

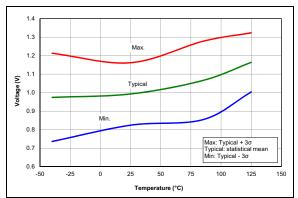


FIGURE 36-65: POR Rearm Voltage, NP Mode, PIC16LF1615/9 Only.

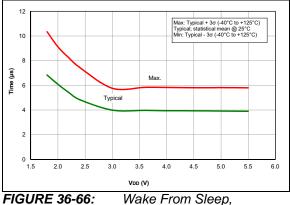
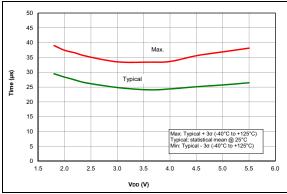
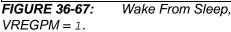


FIGURE 36-66: Wake From Sleep, VREGPM = 0.





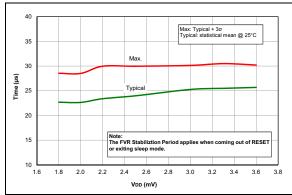


FIGURE 36-68: FVR Stabilization Period, PIC16LF1615/9 Only.

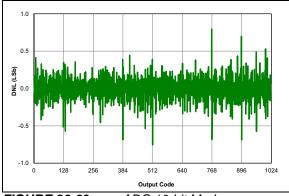


FIGURE 36-69: ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1μ S, 25° C.

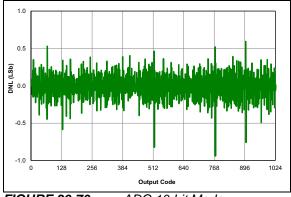


FIGURE 36-70: ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 4μ S, 25° C.

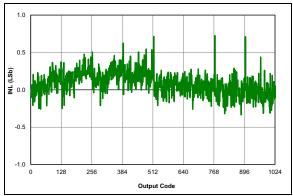


FIGURE 36-71: ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1μ S, 25° C.

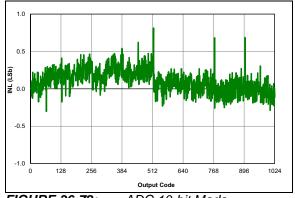


FIGURE 36-72: ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4μ S, 25° C

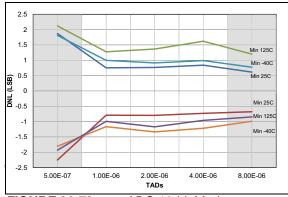


FIGURE 36-73: ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.

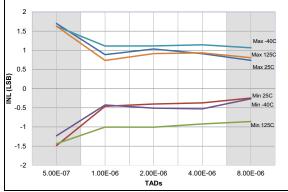


FIGURE 36-74: ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, VREF = 3.0V.

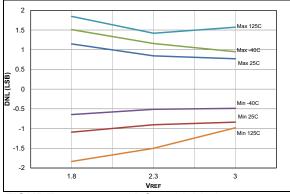


FIGURE 36-75: ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, $TAD = 1 \mu S$.

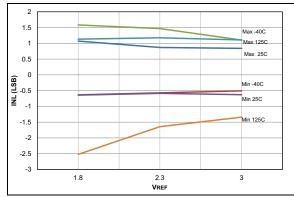


FIGURE 36-76: ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = $1 \mu S$.

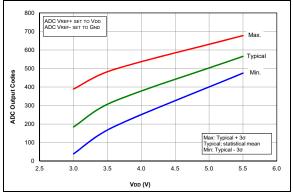


FIGURE 36-77: Temp. Indicator Initial Offset, High Range, Temp. = 20°C, PIC16F1615/9 Only.

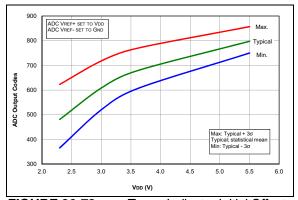


FIGURE 36-78: Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16F1615/9 Only.

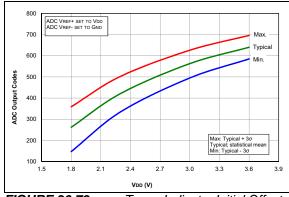


FIGURE 36-79: Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16LF1615/9 Only.

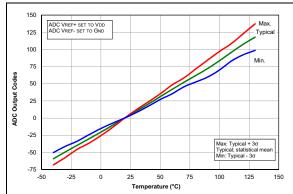


FIGURE 36-80: Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 5.5V, PIC16F1615/9 Only.

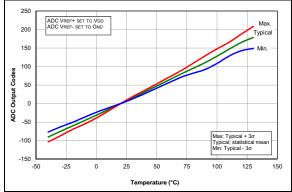


FIGURE 36-81: Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 3.0V, PIC16F1615/9 Only.

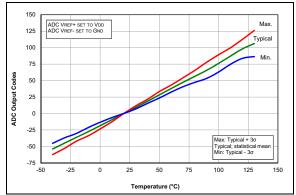


FIGURE 36-82: Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0V, PIC16F1615/9 Only.

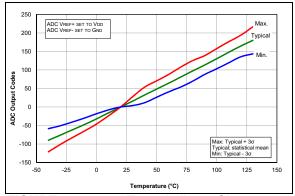


FIGURE 36-83: Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 1.8V, PIC16LF1615/9 Only.

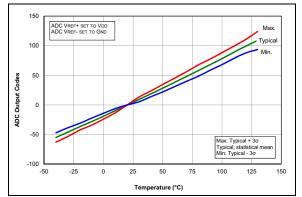


FIGURE 36-84: Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0V, PIC16LF1615/9 Only.

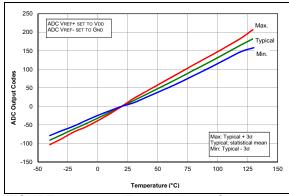


FIGURE 36-85: Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 3.6V, PIC16LF1615/9 Only.

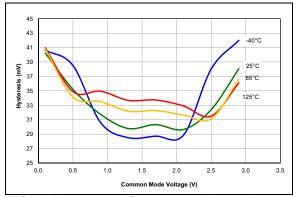


FIGURE 36-86: Comparator Hysteresis, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values.

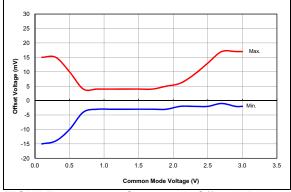


FIGURE 36-87: Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values at 25°C.

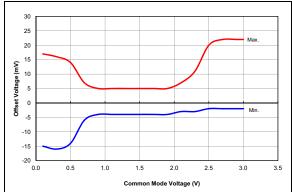


FIGURE 36-88: Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values From -40°C to 125°C.

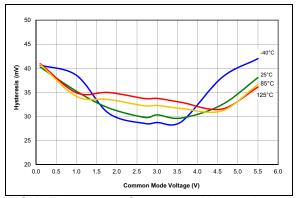


FIGURE 36-89: Comparator Hysteresis, NP Mode (CxSP = 1), VDD = 5.5V, Typical Measured Values, PIC16F1615/9 Only.

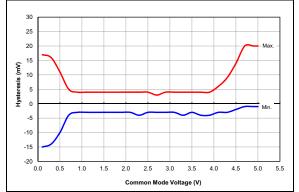


FIGURE 36-90: Comparator Offset, NP Mode (CxSP = 1), VDD = 5.0V, Typical Measured Values at 25°C, PIC16F1615/9 Only.

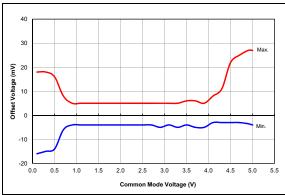


FIGURE 36-91:Comparator Offset, NP Mode(CxSP = 1), VDD = 5.5V, Typical Measured ValuesFrom -40°C to 125°C, PIC16F1615/9 Only.

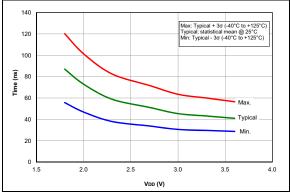


FIGURE 36-92: Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values, PIC16LF1615/9 Only.

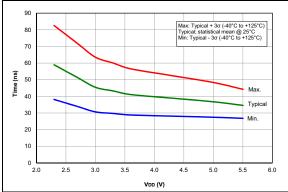


FIGURE 36-93: Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values, PIC16F1615/9 Only.

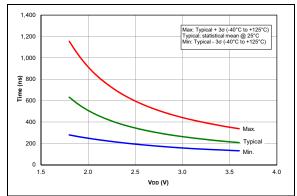


FIGURE 36-94: Comparator Output Filter Delay Time Over Temp., NP Mode (CxSP = 1), Typical Measured Values, PIC16LF1615/9 Only.

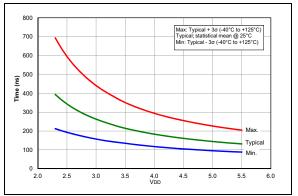


FIGURE 36-95: Comparator Output Filter Delay Time Over Temp., NP Mode (CxSP = 1), Typical Measured Values, PIC16F1615/9 Only.

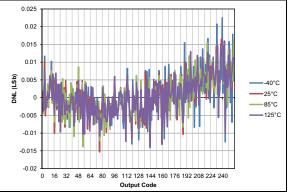
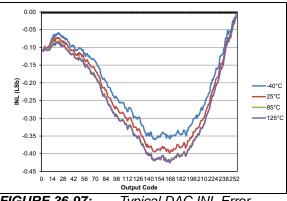
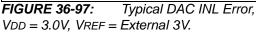


FIGURE 36-96: Typical DAC DNL Error, VDD = 3.0V, VREF = External 3V.





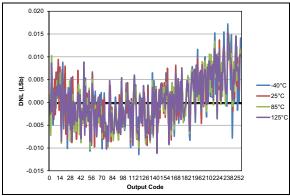


FIGURE 36-98: Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC16F1615/9 Only.

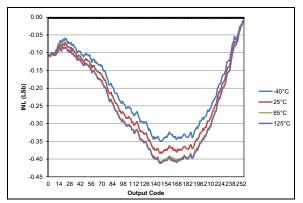
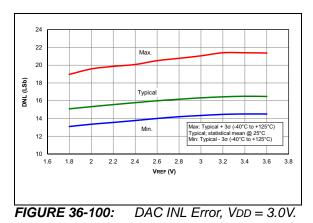


FIGURE 36-99: Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC16F1615/9 Only.



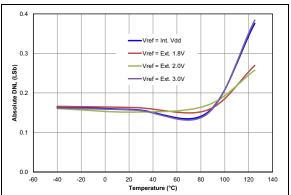


FIGURE 36-101: Absolute Value of DAC DNL Error, VDD = 3.0V, VREF = VDD.

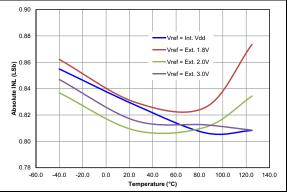


FIGURE 36-102: Absolute Value of DAC INL Error, VDD = 3.0V, VREF = VDD.

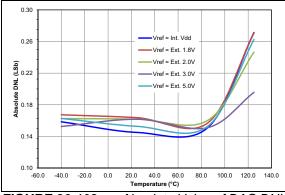


FIGURE 36-103: Absolute Value of DAC DNL Error, VDD = 5.0V, VREF = VDD, PIC16F1615/9 Only.

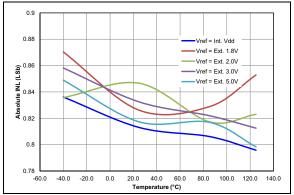


FIGURE 36-104: Absolute Value of DAC INL Error, VDD = 5.0V, VREF = VDD, PIC16F1615/9 Only.

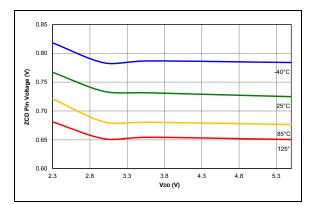


FIGURE 36-105: ZCD Pin Voltage, Typical Measured Values.

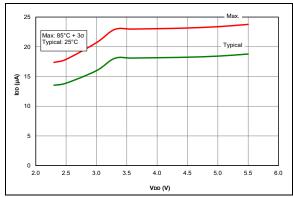


FIGURE 36-106: ZCD Response Time over Voltage Typical Measured Values.

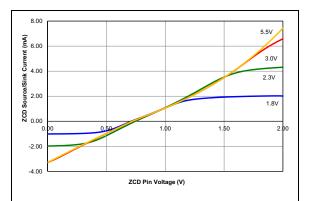


FIGURE 36-107: ZCD Pin Current over ZCD Pin Voltage, Typical Measured Values from -40°C to 125°C.

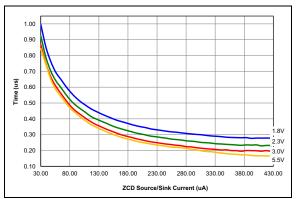


FIGURE 36-108: ZCD Pin Response Timer over Current, Typical Measured Values from -40°C to 125°C.

37.0 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

37.1 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

37.2 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

37.3 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

37.4 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

37.5 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

37.6 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.

37.7 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 upgradeable 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.

37.8 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.

37.9 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[™]).

37.10 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.

37.11 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.

37.12 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[®]

38.0 PACKAGING INFORMATION

38.1 Package Marking Information

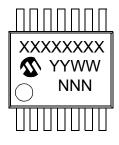
14-Lead PDIP



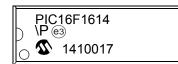
14-Lead SOIC (.150")



14-Lead TSSOP



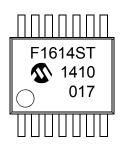
Example



Example

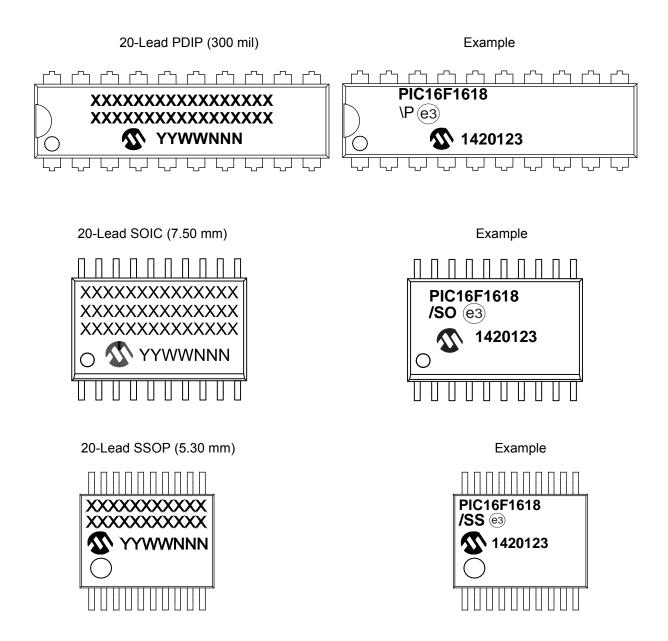


Example

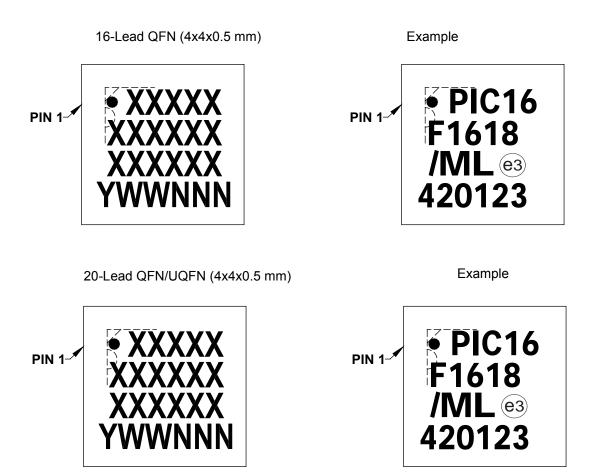


Legend	XXX Y YY WW NNN @3 *	Customer-specific information Year code (last digit of calendar year) Year code (last 2 digits of calendar year) Week code (week of January 1 is week '01') Alphanumeric traceability code Pb-free JEDEC [®] designator for Matte Tin (Sn) This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.
	be carrie	nt the full Microchip part number cannot be marked on one line, it will d over to the next line, thus limiting the number of available s for customer-specific information.

38.1 Package Marking Information (Continued)



38.1 Package Marking Information (Continued)

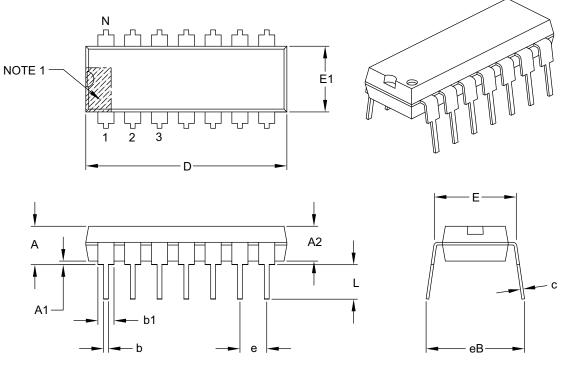


38.2 Package Details

The following sections give the technical details of the packages.

14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



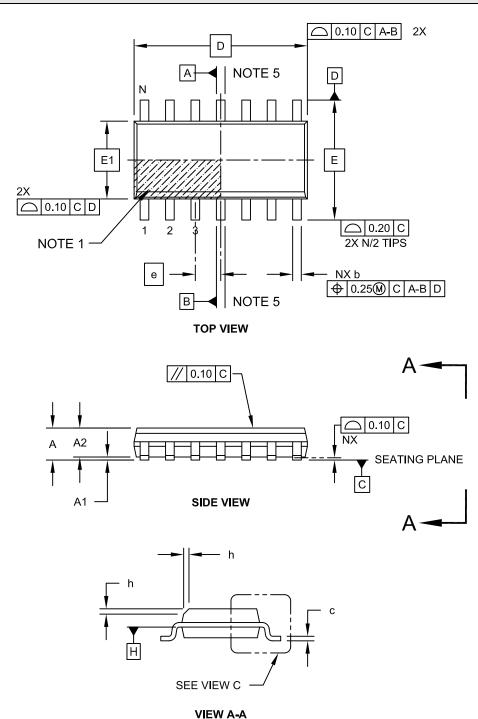
	Units			
Dimens	ion Limits	MIN	NOM	MAX
Number of Pins	Ν		14	
Pitch	е		.100 BSC	
Top to Seating Plane	А	-	-	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	-	_
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	С	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	-	-	.430

Notes:

- 1. Pin 1 visual index feature may vary, but must be located with 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 C04-005B



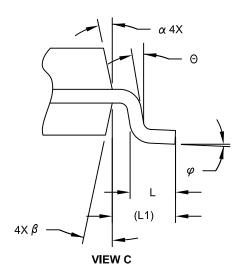
14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

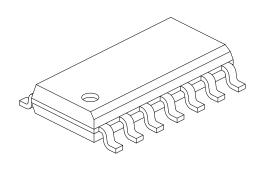
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

Microchip Technology Drawing No. C04-065C Sheet 1 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 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 Lin	nits	MIN	NOM	MAX
Number of Pins	N		14	
Pitch	е		1.27 BSC	
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	8.65 BSC		
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1		1.04 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	С	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

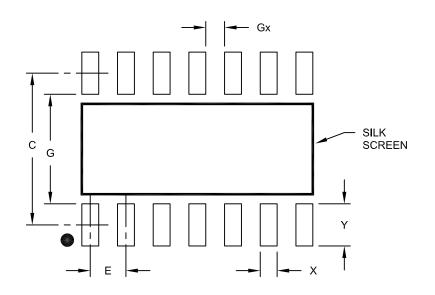
Notes:

- 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 No. C04-065C Sheet 2 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 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

	Units		MILLIMETERS		
Dimension	Dimension Limits		NOM	MAX	
Contact Pitch	E	1.27 BSC			
Contact Pad Spacing	С		5.40		
Contact Pad Width	Х			0.60	
Contact Pad Length	Y			1.50	
Distance Between Pads	Gx	0.67			
Distance Between Pads	G	3.90			

Notes:

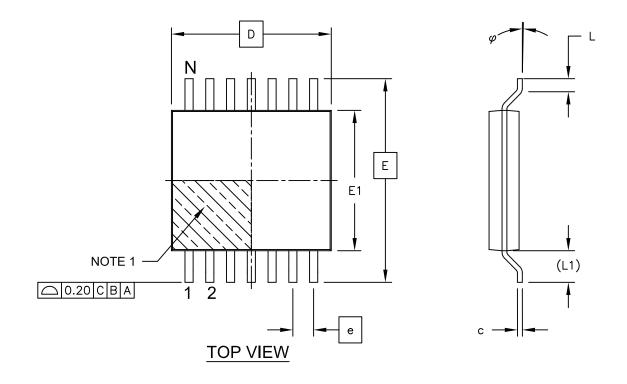
1. Dimensioning and tolerancing per ASME Y14.5M

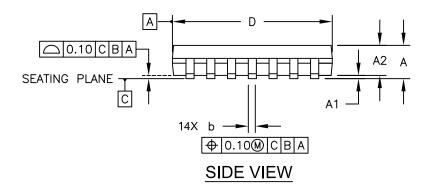
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

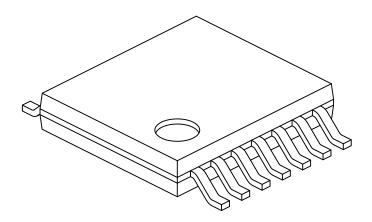




Microchip Technology Drawing C04-087C Sheet 1 of 2

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	N	IILLIMETER	S	
Dimension	Limits	MIN	NOM	MAX
Number of Pins	N		14	
Pitch	е		0.65 BSC	
Overall Height	Α	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)		1.00 REF	
Foot Angle	φ	0°	-	8°
Lead Thickness	С	0.09	-	0.20
Lead Width	b	0.19	-	0.30

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

3. Dimensioning and tolerancing per ASME Y14.5M

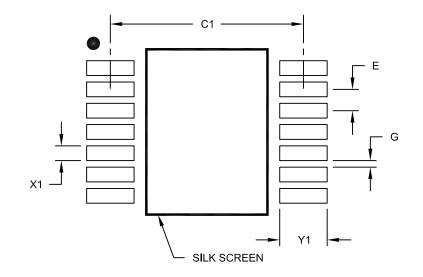
BSC: Basic Dimension. Theoretically exact value shown without tolerances. REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

^{2.} Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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		0.65 BSC	
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

Notes:

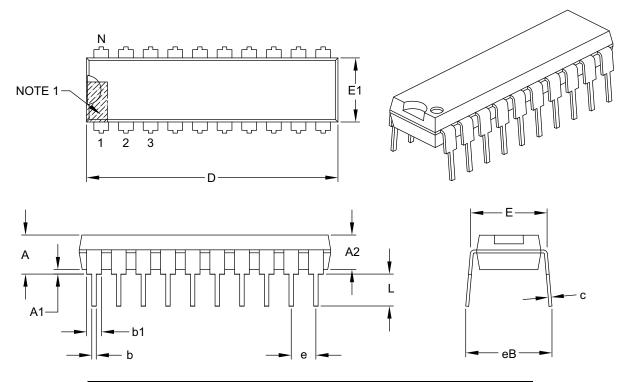
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

20-Lead Plastic Dual In-Line (P) – 300 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		20		
Pitch	е		.100 BSC		
Top to Seating Plane	А	-	-	.210	
Molded Package Thickness	A2	.115	.130	.195	
Base to Seating Plane	A1	.015	-	-	
Shoulder to Shoulder Width	E	.300	.310	.325	
Molded Package Width	E1	.240	.250	.280	
Overall Length	D	.980	1.030	1.060	
Tip to Seating Plane	L	.115	.130	.150	
Lead Thickness	С	.008	.010	.015	
Upper Lead Width	b1	.045	.060	.070	
Lower Lead Width	b	.014	.018	.022	
Overall Row Spacing §	eB	-	-	.430	

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.

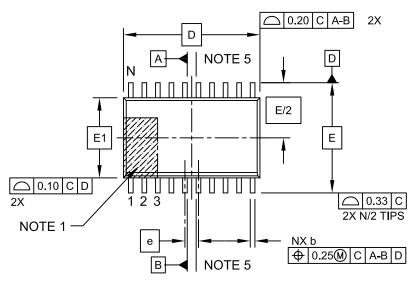
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

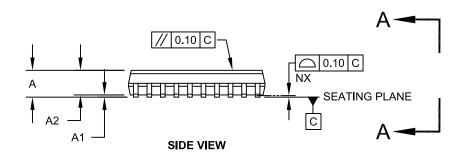
Microchip Technology Drawing C04-019B

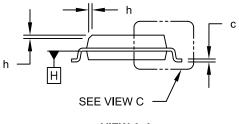
20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging







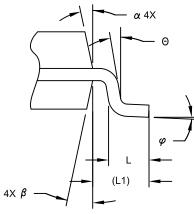


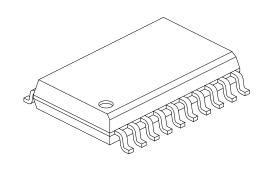
VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging





VIEW C

Units		MILLIMETERS			
Dimension Lim	nits	MIN	NOM	MAX	
Number of Pins	N	20			
Pitch	е		1.27 BSC		
Overall Height	Α	-	-	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	12.80 BSC			
Chamfer (Optional)	h	0.25	-	0.75	
Foot Length	L	0.40	-	1.27	
Footprint	L1		1.40 REF		
Lead Angle	Θ	0°	-	-	
Foot Angle	φ	0° - 8°			
Lead Thickness	С	0.20 - 0.33			
Lead Width	b	0.31	-	0.51	
Mold Draft Angle Top	α	5°	-	15°	
Mold Draft Angle Bottom	β	5°	-	15°	

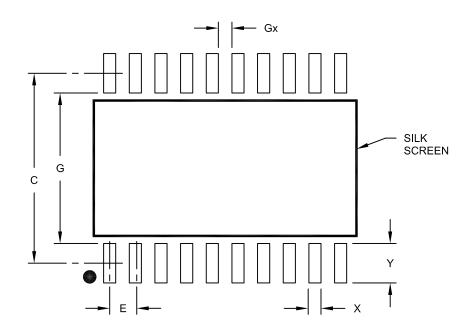
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic
- 3. 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 No. C04-094C Sheet 2 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS			
Dimension	Dimension Limits		NOM	MAX	
Contact Pitch	E		1.27 BSC		
Contact Pad Spacing	С		9.40		
Contact Pad Width (X20)	X			0.60	
Contact Pad Length (X20)	Y			1.95	
Distance Between Pads	Gx	0.67			
Distance Between Pads	G	7.45			

Notes:

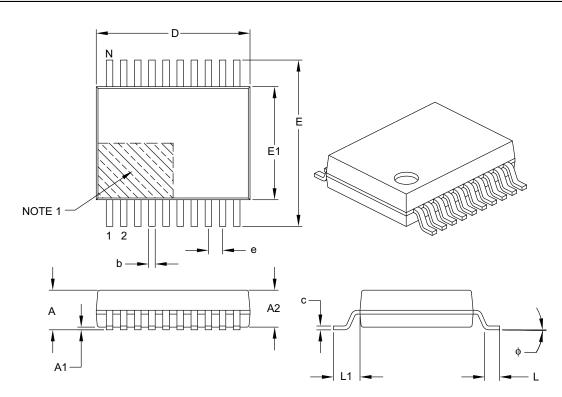
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A



Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		MILLIMETERS	5
Dimension	Dimension Limits		NOM	MAX
Number of Pins	Ν		20	
Pitch	е		0.65 BSC	
Overall Height	А	-	-	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	-	-
Overall Width	Е	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	С	0.09	_	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	_	0.38

Notes:

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.

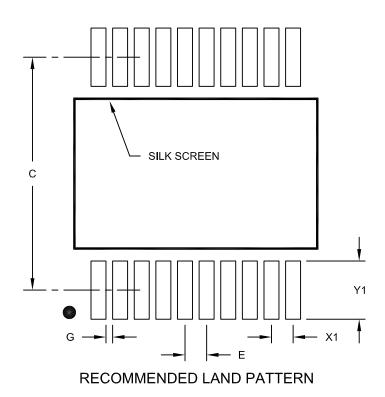
- 3. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

20-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



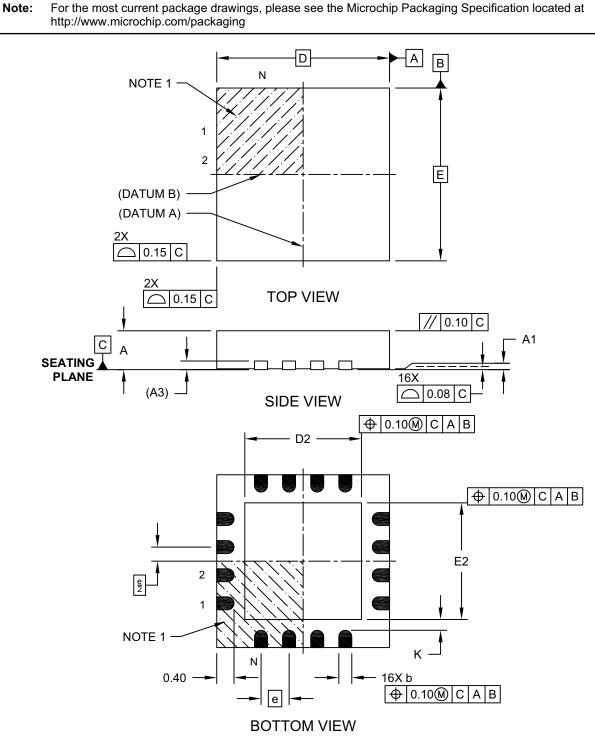
Units		MILLIMETERS			
Dimension Limits		MIN	NOM	MAX	
Contact Pitch	E	0.65 BSC			
Contact Pad Spacing	С		7.20		
Contact Pad Width (X20)	X1			0.45	
Contact Pad Length (X20)	Y1			1.75	
Distance Between Pads	G	0.20			

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

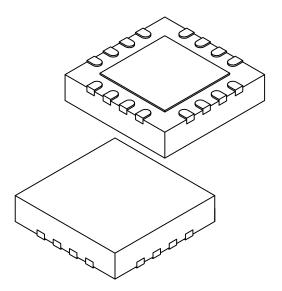


16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

Microchip Technology Drawing C04-127D Sheet 1 of 2

16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		N	IILLIMETER	S
Dimension Limits		MIN	NOM	MAX
Number of Pins	Ν		16	
Pitch	е		0.65 BSC	
Overall Height	Α	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.50	2.65	2.80
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.50	2.65	2.80
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.30 0.40 0.50		
Contact-to-Exposed Pad	К	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated

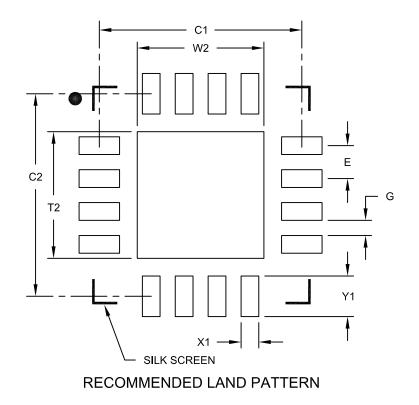
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances. REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127D Sheet 2 of 2

16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



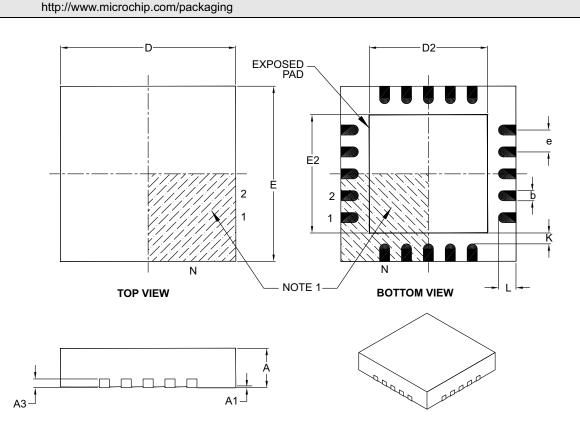
Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	E 0.65 B		
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X28)	X1			0.35
Contact Pad Length (X28)	Y1			0.80
Distance Between Pads	G	0.30		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2127A



For the most current package drawings, please see the Microchip Packaging Specification located at

20-Lead Plastic Quad Flat, No Lead Package (ML) – 4x4x0.9 mm Body [QFN]

Units MILLIMETERS Dimension Limits MIN NOM MAX Number of Pins Ν 20 Pitch 0.50 BSC е **Overall Height** А 0.80 0.90 1.00 Standoff A1 0.00 0.02 0.05 Contact Thickness A3 0.20 REF Overall Width 4.00 BSC Е Exposed Pad Width E2 2.60 2.70 2.80 **Overall Length** D 4.00 BSC Exposed Pad Length D2 2.60 2.70 2.80 Contact Width 0.18 0.25 0.30 b Contact Length L 0.30 0.40 0.50 Contact-to-Exposed Pad Κ 0.20 _ _

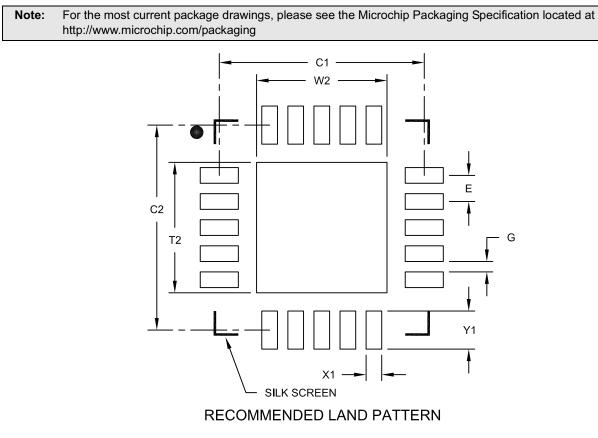
Notes:

Note:

- 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 C04-126B

20-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4 mm Body [QFN] With 0.40 mm Contact Length



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
Contact Pad Spacing	C1		3.93	
Contact Pad Spacing	C2		3.93	
Contact Pad Width	X1			0.30
Contact Pad Length	Y1			0.73
Distance Between Pads	G	0.20		

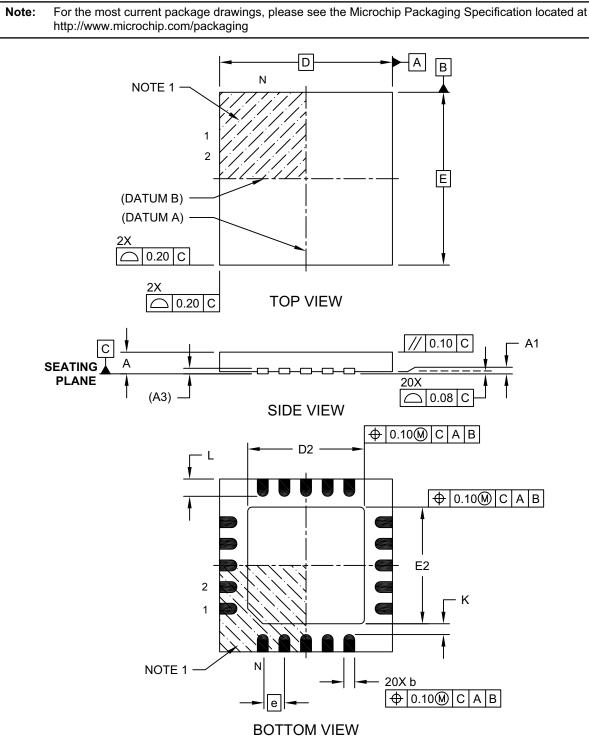
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2126A

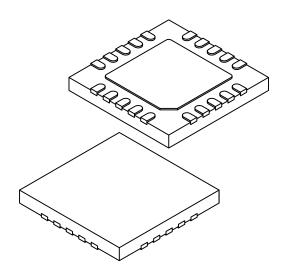
20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]



Microchip Technology Drawing C04-255A Sheet 1 of 2

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 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	Dimension Limits		NOM	MAX
Number of Terminals	N		20	
Pitch	е		0.50 BSC	
Overall Height	Α	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.127 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.60	2.70	2.80
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.60	2.70	2.80
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Terminal-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

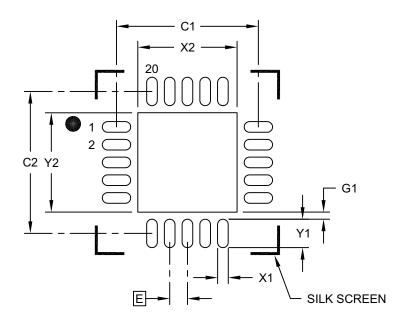
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances. REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-255A Sheet 2 of 2

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	ontact Pitch E		0.50 BSC	
Optional Center Pad Width	X2			2.80
Optional Center Pad Length	Y2			2.80
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X20)	X1			0.30
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2255A

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (10/2014)

Original release.

Revision B (4/2015)

Added High-Current pins.

Updated PIC12/16(L)F161X Family Types table and Packages Table

Added Figures 36-7 and 36-8 for $\mathsf{VOH}\xspace$ vs. $\mathsf{IOH}\xspace$ for high drive pins.

Deleted Figures 36-27 and 36-28.

Updated Example 3-2.

Updated Figures 5-1, 23-1, 23-2, 28-12, 31-1, 31-2, 31-4, 1, and 35-6.

Updated Registers 23-4, 27-2, 31-4 and 31-7.

Updated Sections 26.3, 25.4.2, 27.1, 30.0, 35.0, and 35.1.

Updated Table 1-2, 1-3, 3-5, 3-16, 13-2, 23-3, 23-4, 23-7, 28-1, 35-4, 35-8, and 35-17.

Updated Section 23 - added missing modes/mode summary table, reworded text to be more clear/descriptive.

Minor typos corrected.

Revision C (5/2016)

Added High endurance column to Table 1: PIC12/16(L)F161x Family Types.

Minor typos corrected.

Updated the High-Endurance Flash data memory information on the cover page. Updated Register 21-1.

Updated Section 19.6, 19.7. Updated Registers 19-2, 29-1, and 31-22. Updated Table 3: Pin Allocations Table and Table 5-1. Updated Figure 19-2.

Updated Package Drawings C04-127.

Revision D (11/2017)

Added Equation 21-2: R-C Calculations and Example 21-1. Added Sections 5.3.2 Clock Switching Before Sleep; 20.5.1: Correction by AC Coupling, and 23.2: PRx Period Register.

Updated Example 17-1; Figure 16-1; Register 12-8, 12-16, 24-4; Sections 11.3, 11.5, 17.1.2, 17.2.6, 23.1 and 24.6; Tables 1, 1-3, 5.1, 24-4, 35-8 and 35-11.

Removed Figures 36-29, 36-30, IPD Watchdog Timer.

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 Micro-chip 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://www.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⁾ - <u>X</u> /XX	<u>xxx</u>	Examples:
Device	Tape and Reel Temperature Package Option Range	Pattern	a) PIC16LF1615T - I/SL Tape and Reel, Industrial temperature, SOIC package b) PIC16F1619 - I/P
Device:	PIC16LF1615, PIC16F1615, PIC16LF1619, PIC16F1619		c) PIC16F1619 - I/F Industrial temperature PDIP package c) PIC16F1619 - E/ML 298 Extended temperature,
Tape and Reel Option:	Blank = Standard packaging (tube or tray) T = Tape and Reel ⁽¹⁾		QFN package QTP pattern #298
Temperature Range:	I = -40° C to $+85^{\circ}$ C (Industrial) E = -40° C to $+125^{\circ}$ C (Extended)		
Package: ⁽²⁾	ML = QFN (16-Lead and 20-Lead) P = Plastic DIP SL = SOIC (14-Lead) ST = TSSOP GZ = UQFN (20-Lead)		Note 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 with your Microchip Sales Office for package availability with the Tape and Reel option.
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)		2: For other small form-factor package availability and marking information, please visit www.microchip.com/packaging or contact your local sales office.

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, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA 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.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., 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 trademark 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.

 $\textcircled{\mbox{\sc c}}$ 2014-2017, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-5224-2316-4



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

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 Tel: 317-536-2380

Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800

Raleigh, NC Tel: 919-844-7510

New York, NY Tel: 631-435-6000

San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270

Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney Tel: 61-2-9868-6733

China - Beijing Tel: 86-10-8569-7000 China - Chengdu

Tel: 86-28-8665-5511 China - Chongqing Tel: 86-23-8980-9588

China - Dongguan Tel: 86-769-8702-9880

China - Guangzhou Tel: 86-20-8755-8029

China - Hangzhou Tel: 86-571-8792-8115

China - Hong Kong SAR Tel: 852-2943-5100

China - Nanjing Tel: 86-25-8473-2460

China - Qingdao Tel: 86-532-8502-7355

China - Shanghai Tel: 86-21-3326-8000

China - Shenyang Tel: 86-24-2334-2829

China - Shenzhen Tel: 86-755-8864-2200

China - Suzhou Tel: 86-186-6233-1526

China - Wuhan Tel: 86-27-5980-5300

China - Xian Tel: 86-29-8833-7252

China - Xiamen

ASIA/PACIFIC

India - Bangalore Tel: 91-80-3090-4444

India - New Delhi Tel: 91-11-4160-8631 India - Pune

Tel: 91-20-4121-0141 Japan - Osaka

Tel: 81-6-6152-7160 Japan - Tokyo

Tel: 81-3-6880- 3770 Korea - Daegu

Tel: 82-53-744-4301 Korea - Seoul

Tel: 82-2-554-7200

Malaysia - Kuala Lumpur Tel: 60-3-7651-7906

Malaysia - Penang Tel: 60-4-227-8870

Philippines - Manila Tel: 63-2-634-9065

Singapore Tel: 65-6334-8870

Taiwan - Hsin Chu Tel: 886-3-577-8366

Taiwan - Kaohsiung Tel: 886-7-213-7830

Taiwan - Taipei Tel: 886-2-2508-8600

Thailand - Bangkok Tel: 66-2-694-1351

Vietnam - Ho Chi Minh Tel: 84-28-5448-2100

Netherlands - Drunen Tel: 31-416-690399

EUROPE

Austria - Wels

Tel: 43-7242-2244-39

Tel: 45-4450-2828

Fax: 45-4485-2829

Tel: 358-9-4520-820

Tel: 33-1-69-53-63-20

Fax: 33-1-69-30-90-79

Germany - Garching

Tel: 49-2129-3766400

Germany - Heilbronn

Germany - Karlsruhe

Tel: 49-721-625370

Germany - Munich

Tel: 49-89-627-144-0

Fax: 49-89-627-144-44

Germany - Rosenheim

Tel: 49-8031-354-560

Israel - Ra'anana

Italy - Milan

Italy - Padova

Tel: 972-9-744-7705

Tel: 39-0331-742611

Fax: 39-0331-466781

Tel: 39-049-7625286

Tel: 49-7131-67-3636

Tel: 49-8931-9700

Germany - Haan

Finland - Espoo

France - Paris

Fax: 43-7242-2244-393

Denmark - Copenhagen

Fax: 31-416-690340

Norway - Trondheim Tel: 47-7289-7561

Poland - Warsaw Tel: 48-22-3325737

Romania - Bucharest Tel: 40-21-407-87-50

Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

Sweden - Gothenberg Tel: 46-31-704-60-40

Sweden - Stockholm Tel: 46-8-5090-4654

UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820

Tel: 86-592-2388138 China - Zhuhai

Tel: 86-756-3210040

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU category:

Click to view products by Microchip manufacturer:

Other Similar products are found below :

009936B CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X S9S08SL8F1CTJR STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR 400801H LC87F0G08AUJA-AH 026923G