PIC16(L)F1574/5/8/9

14/20-Pin MCUs with High Precision 16-Bit PWMs

Description

PIC16(L)F1574/5/8/9 microcontrollers combine the capabilities of 16-bit PWMs with Analog to suit a variety of applications. These devices deliver four 16-bit PWMs with independent timers for applications where high resolution is needed, such as LED lighting, stepper motors, power supplies and other general purpose applications. The core independent peripherals (16-bit PWMs, Complementary Waveform Generator), Enhanced Universal Synchronous Asynchronous Receiver Transceiver (EUSART) and Analog (ADCs, Comparator and DAC) enable closed-loop feedback and communication for use in multiple market segments. The Peripheral Pin Select (PPS) functionality allows for I/O pin remapping of the digital peripherals for increased flexibility. The EUSART peripheral enables the communication for applications such as LIN.

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
- Two 8-Bit Timers
- One 16-Bit Timer
- Four additional 16-Bit Timers available using the 16-Bit PWMs
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Low-Power Brown-out Reset (LPBOR)
- Programmable Watchdog Timer (WDT) up to 256s
- Programmable Code Protection

Memory

- Up to 8 KW Flash Program Memory
- Up to 1024 Bytes Data SRAM Memory
- Direct, Indirect and Relative Addressing modes
- 128 Bytes of Nonvolatile Data Storage, High-Endurance Flash (HEF)

Operating Characteristics

- Operating Voltage Range:
  - 1.8V to 3.6V (PIC16LF1574/5/8/9)
  - 2.3V to 5.5V (PIC16F1574/5/8/9)
- Temperature Range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C
- Internal Voltage Reference module
- In-Circuit Serial Programming™ (ICSP™) via Two Pins

Digital Peripherals

- 16-Bit PWM:
  - Four 16-bit PWMs with independent timers
  - Multiple output modes (standard, center aligned, set and toggle on register match)
  - User settings for phase, duty cycle, period, offset and polarity
  - 16-bit timer capability
  - Interrupts generated based on timer matches with offset, duty cycle, period and phase registers
- Complementary Waveform Generator (CWG):
  - Rising and falling edge dead-band control
  - Multiple signal sources
- Enhanced Universal Synchronous Asynchronous Receiver Transceiver (EUSART):
  - Supports LIN applications
- Peripheral Pin Select (PPS):
  - I/O pin remapping of digital peripherals

Device I/O Port Features

- Up to 18 I/Os
- Individually Selectable Weak Pull-ups
- Interrupt-on-Change Pins Option with Edge-Selectable Option

Low-Power Features

- Sleep mode: 20 nA @ 1.8V, typical
- Watchdog Timer: 260 nA @ 1.8V, typical
- Operating Current:
  - 30 uA/MHz @ 1.8V, typical

Analog Peripherals

- 10-Bit Analog-to-Digital Converter (ADC):
  - Up to 12 external channels
  - Conversion available during Sleep
- Two Comparators:
  - Low-Power/High-Speed modes
  - Fixed Voltage Reference at (non)inverting input(s)
  - Comparator outputs externally accessible
  - Synchronization with Timer1 clock source
  - Software hysteresis enable
- 5-Bit Digital-to-Analog Converter (DAC):
  - 5-bit resolution, rail-to-rail
  - Positive Reference Selection
  - Unbuffered I/O pin output
  - Internal connections to ADCs and comparators
- Voltage Reference:
  - Fixed Voltage Reference with 1.024V, 2.048V and 4.096V output levels

Clocking Structure

- Precision Internal Oscillator:
  - Factory calibrated ±1%, typical
  - Software-selectable clock speeds from 31 kHz to 32 MHz
- External Oscillator Block with:
  - Two external clock modes up to 32 MHz
- Digital Oscillator Input Available

TABLE 1: PIC12(L)F1571/2 AND PIC16(L)F1574/5/8/9 FAMILY TYPES

Device	Data Sheet Index	Program Flash Memory (Kwords)	Program Flash Memory (Kbytes)	Data SRAM (bytes)	I/O Pins	8-Bit/16-Bit Timers	Comparators	16-Bit PWM	10-Bit ADC (ch)	5-Bit DAC	CWG	EUSART	PPS	Debug(1)
PIC12(L)F1571	(A)	1	1.75	128	6	2/4(2)	1	3	4	1	1	0	N	I
PIC12(L)F1572	(A)	2	3.5	256	6	2/4(2)	1	3	4	1	1	1	N	I
PIC16(L)F1574	(B)	4	7	512	12	2/5(3)	2	4	8	1	1	1	Y	I
PIC16(L)F1575	(B)	8	14	1024	12	2/5(3)	2	4	8	1	1	1	Y	I
PIC16(L)F1578	(B)	4	7	512	18	2/5(3)	2	4	12	1	1	1	Y	I
PIC16(L)F1579	(B)	8	14	1024	18	2/5(3)	2	4	12	1	1	1	Y	I

Note 1: I – Debugging integrated on chip.
2: Three additional 16-bit timers available when not using the 16-bit PWM outputs.
3: Four additional 16-bit timers available when not using the 16-bit PWM outputs.

Data Sheet Index:

A: DS-40001723 PIC12(L)F1571/2 Data Sheet, 8-Pin Flash, 8-bit MCU with High-Precision 16-bit PWM
B: Future Release PIC16(L)F1574/5/8/9 Data Sheet, 8-Pin Flash, 8-bit MCU with High-Precision 16-bit PWM

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

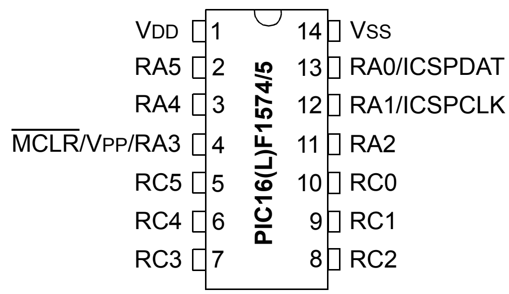
TABLE 2: PACKAGES

Packages	PDIP	SOIC	TSSOP	SSOP	UQFN
PIC16(L)F1574	●	●	●		●
PIC16(L)F1575	●	●	●		●
PIC16(L)F1578	●	●		●	●
PIC16(L)F1579	●	●		●	●

Note: Pin details are subject to change.

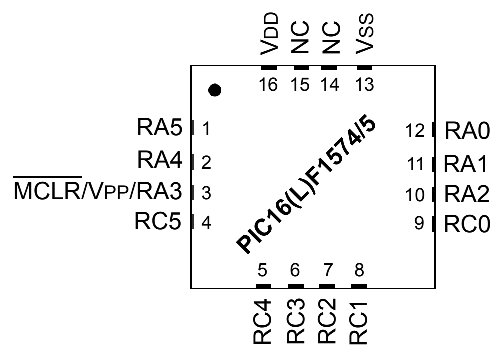
PIN DIAGRAMS

FIGURE 1: 14-PIN PDIP, SOIC, TSSOP



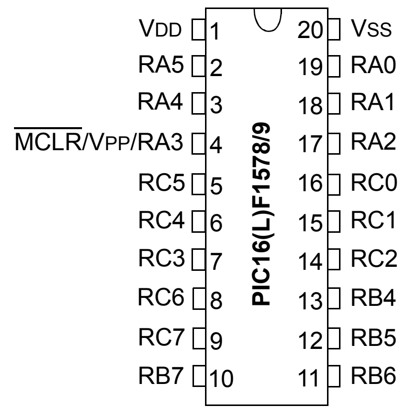
Note: See Table 3 for the pin allocation table.

FIGURE 2: 16-PIN UQFN (4x4)



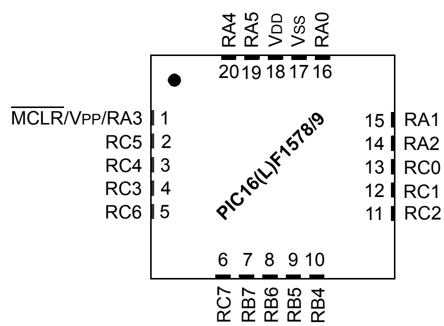
Note: See Table 3 for the pin allocation table.

FIGURE 3: 20-PIN PDIP, SOIC, SSOP



Note: See Table 4 for the pin allocation table.

FIGURE 4: 20-PIN UQFN (4x4)



Note: See Table 4 for the pin allocation table.

PIN ALLOCATION TABLES

TABLE 3: 14/16-PIN ALLOCATION TABLE (PIC16(L)F1574/5)

I/O	14-Pin PDIP/SOIC/TSSOP	16-Pin UQFN	ADC	Reference	Comparator	Timers	PWM	EUSART	CWG	Interrupt	Pull-up	Basic
RA0	13	12	AN0	DAC1OUT1	C1IN+	—	—	—	—	IOC	Y	ICSPDAT
RA1	12	11	AN1	VREF+	C1IN0-/C2IN0-	—	—	—	—	IOC	Y	ICSPCLK
RA2	11	10	AN2	—	—	T0CKI(1)	—	—	CWG1IN(1)	INT(1)/IOC	Y	—
RA3	4	3	—	—	—	—	—	—	—	IOC	Y	$\overline{MCLR}$/VPP
RA4	3	2	AN3	—	—	T1G(1)	—	—	—	IOC	Y	CLKOUT
RA5	2	1	—	—	—	T1CKI(1)	—	—	—	IOC	Y	CLKIN
RC0	10	9	AN4	—	C2IN+	—	—	—	—	IOC	Y	—
RC1	9	8	AN5	—	C1IN1-/C2IN1-	—	—	—	—	IOC	Y	—
RC2	8	7	AN6	—	C1IN2-/C2IN2-	—	—	—	—	IOC	Y	—
RC3	7	6	AN7	—	C1IN3-/C2IN3-	—	—	—	—	IOC	Y	—
RC4	6	5	ADCACT(1)	—	—	—	—	CK(1)	—	IOC	Y	—
RC5	5	4	—	—	—	—	—	RX(1,3)	—	IOC	Y	—
VDD	1	16	—	—	—	—	—	—	—	—	—	VDD
Vss	14	13	—	—	—	—	—	—	—	—	—	VSS
OUT(2)	—	—	—	—	C1OUT	—	PWM1OUT	DT(3)	CWG1A	—	—	—
	—	—	—	—	C2OUT	—	PWM2OUT	CK	CWG1B	—	—	—
	—	—	—	—	—	—	PWM3OUT	TX	—	—	—	—
	—	—	—	—	—	—	PWM4OUT	—	—	—	—	—

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

TABLE 4: 20-PIN ALLOCATION TABLE (PIC16(L)F1578/9)

I/O	20-Pin PDIP/SOIC/SSOP	20-Pin UQFN	ADC	Reference	Comparator	Timers	PWM	EUSART	CWG	Interrupt	Pull-up	Basic
RA0	19	16	AN0	DAC1OUT1	C1IN+	—	—	—	—	IOC	Y	ICSPDAT
RA1	18	15	AN1	VREF+	C1IN0-/C2IN0-	—	—	—	—	IOC	Y	ICSPCLK
RA2	17	14	AN2	—	—	T0CKI(1)	—	—	CWG1IN(1)	INT(1)/IOC	Y	—
RA3	4	1	—	—	—	—	—	—	—	IOC	Y	$\overline{\text{MCLR}}$/VPP
RA4	3	20	AN3	—	—	T1G(1)	—	—	—	IOC	Y	CLKOUT
RA5	2	19	—	—	—	T1CKI(1)	—	—	—	IOC	Y	CLKIN
RB4	13	10	AN10	—	—	—	—	—	—	IOC	Y	—
RB5	12	9	AN11	—	—	—	—	RX(1,3)	—	IOC	Y	—
RB6	11	8	—	—	—	—	—	—	—	IOC	Y	—
RB7	10	7	—	—	—	—	—	CK(1)	—	IOC	Y	—
RC0	16	13	AN4	—	C2IN+	—	—	—	—	IOC	Y	—
RC1	15	12	AN5	—	C1IN1-/C2IN1-	—	—	—	—	IOC	Y	—
RC2	14	11	AN6	—	C1IN2-/C2IN2-	—	—	—	—	IOC	Y	—
RC3	7	4	AN7	—	C1IN3-/C2IN3-	—	—	—	—	IOC	Y	—
RC4	6	3	ADCACT(1)	—	—	—	—	—	—	IOC	Y	—
RC5	5	2	—	—	—	—	—	—	—	IOC	Y	—
RC6	8	5	AN8	—	—	—	—	—	—	IOC	Y	—
RC7	9	6	AN9	—	—	—	—	—	—	IOC	Y	—
VDD	1	18	—	—	—	—	—	—	—	—	—	VDD
Vss	20	17	—	—	—	—	—	—	—	—	—	Vss
OUT(2)	—	—	—	—	C1OUT	—	PWM1OUT	DT(3)	CWG1A	—	—	—
	—	—	—	—	C2OUT	—	PWM2OUT	CK	CWG1B	—	—	—
	—	—	—	—	—	—	PWM3OUT	TX	—	—	—	—
	—	—	—	—	—	—	PWM4OUT	—	—	—	—	—

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

TABLE OF CONTENTS

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

- Microchip's Worldwide Web site; **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 web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

The PIC16(L)F1574/5/8/9 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 pinout descriptions are shown in Table 1-2 and Table 1-3.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC16(L)F1574	PIC16(L)F1575	PIC16(L)F1578	PIC16(L)F1579
Analog-to-Digital Converter (ADC)		●	●	●	●
Complementary Wave Generator (CWG)		●	●	●	●
Digital-to-Analog Converter (DAC)		●	●	●	●
Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART)		●	●	●	●
Fixed Voltage Reference (FVR)		●	●	●	●
Temperature Indicator		●	●	●	●
Comparators					
	C1	●	●	●	●
	C2	●	●	●	●
PWM Modules					
	PWM1	●	●	●	●
	PWM2	●	●	●	●
	PWM3	●	●	●	●
	PWM4	●	●	●	●
Timers					
	Timer0	●	●	●	●
	Timer1	●	●	●	●
	Timer2	●	●	●	●

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

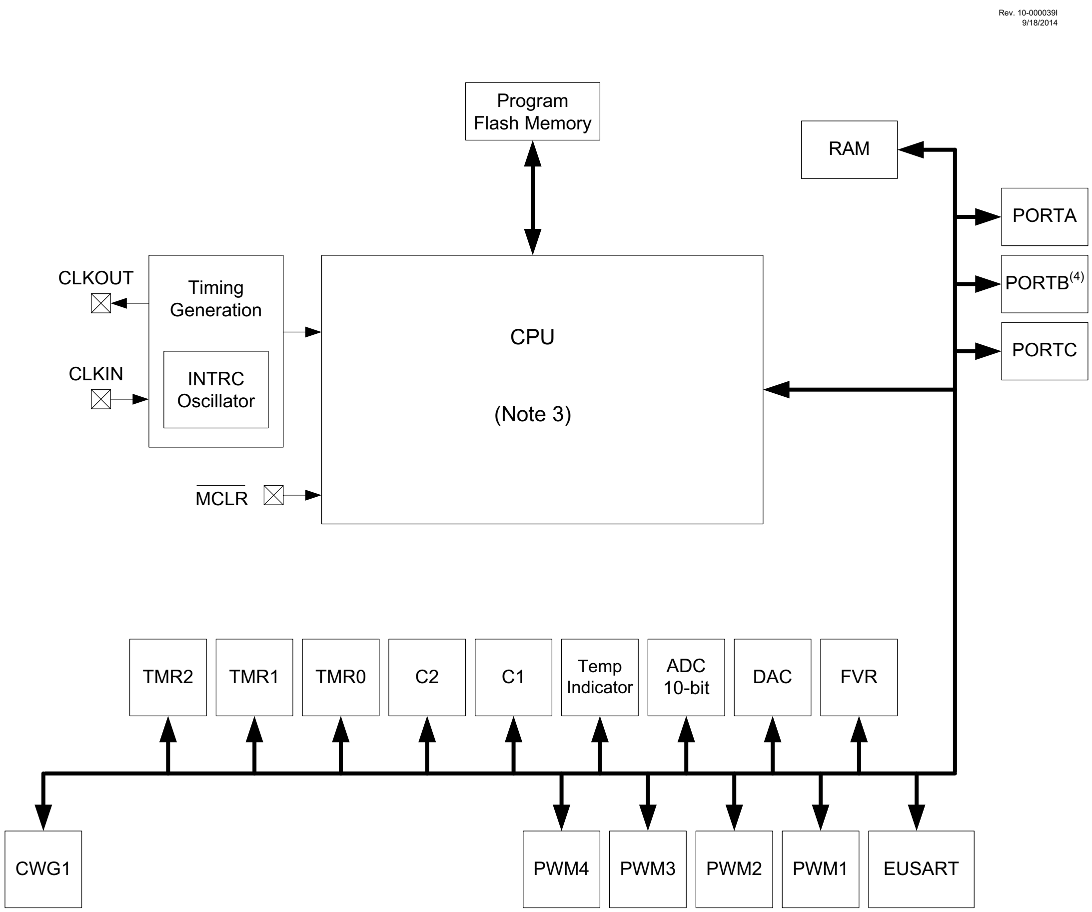
Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

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

FIGURE 1-1: PIC16(L)F1574/5/8/9 BLOCK DIAGRAM



Note 1: See applicable chapters for more information on peripherals.
2: See Table 1-1 for peripherals available on specific devices.
3: See Figure 2-1.
4: PIC16(L)F1578/9 only.

TABLE 1-2: PIC16(L)F1574/5 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT1/ ICSPDAT	RA0	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	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-/ ICSPCLK	RA1	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN1	AN	—	ADC Channel input.
	VREF+	AN	—	Voltage Reference input.
	C1IN0-	AN	—	Comparator negative input.
	C2IN0-	AN	—	Comparator negative input.
	ICSPCLK	ST	—	ICSP Programming Clock.
RA2/AN2/T0CKI(1)/CWG1IN(1)/ INT(1)	RA2	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN2	AN	—	ADC Channel input.
	T0CKI	TTL/ST	—	Timer0 clock input.
	CWG1IN	TTL/ST	—	CWG complementary input.
	INT	TTL/ST	—	External interrupt.
RA3/VPP/$\overline{MCLR}$	RA3	TTL/ST	—	General purpose input with IOC and WPU.
	VPP	HV	—	Programming voltage.
	$\overline{MCLR}$	ST	—	Master Clear with internal pull-up.
RA4/AN3/T1G(1)/CLKOUT	RA4	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN3	AN	—	ADC Channel input.
	T1G	TTL/ST	—	Timer1 Gate input.
	CLKOUT	CMOS/OD	CMOS	FOSC/4 output.
RA5/CLKIN/T1CKI(1)	RA5	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	CLKIN	CMOS	—	External clock input (EC mode).
	T1CKI	TTL/ST	—	Timer1 clock input.
RC0/AN4/C2IN+	RC0	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN4	AN	—	ADC Channel input.
	C2IN+	AN	—	Comparator positive input.
RC1/AN5/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN5	AN	—	ADC Channel input.
	C1IN1-	AN	—	Comparator negative input.
	C2IN1-	AN	—	Comparator negative input.
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN6	AN	—	ADC Channel input.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
RC3/AN7/C1IN3-/C2IN3-	RC3	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN7	AN	—	ADC Channel input.
	C1IN3-	AN	—	Comparator negative input.
	C2IN3-	AN	—	Comparator negative input.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open-Drain
TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I^{2}C™ = Schmitt Trigger input with I^{2}C™ levels
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 12-1.

3: These USART functions are bi-directional. The output pin selections must be the same as the input pin selections.

TABLE 1-2: PIC16(L)F1574/5 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC4/ADCACT(1)/CK(1)	RC4	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	ADCACT	TTL/ST	—	ADC Auto-conversion Trigger input.
	CK	ST	CMOS	USART synchronous clock.
RC5/RX(1,3)	RC5	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	RX	ST	—	USART asynchronous input.
OUT(2)	C1OUT	—	CMOS	Comparator output.
	C2OUT	—	CMOS	Comparator output.
	PWM1OUT	—	CMOS	PWM1 output.
	PWM2OUT	—	CMOS	PWM2 output.
	PWM3OUT	—	CMOS	PWM3 output.
	PWM4OUT	—	CMOS	PWM4 output.
	CWG1A	—	CMOS	Complementary Output Generator Output A.
	CWG1B	—	CMOS	Complementary Output Generator Output B.
	TX/CK	—	CMOS	USART asynchronous TX data/synchronous clock output.
	DT(3)	—	CMOS	USART synchronous data output.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

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 I^2C™ = Schmitt Trigger input with I^2C™ levels
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 12-1.

3: These USART functions are bi-directional. The output pin selections must be the same as the input pin selections.

TABLE 1-3: PIC16(L)F1578/9 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT/ ICSPDAT	RA0	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	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-/ ICSPCLK	RA1	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN1	AN	—	ADC Channel input.
	VREF+	AN	—	Voltage Reference input.
	C1IN0-	AN	—	Comparator negative input.
	C2IN0-	AN	—	Comparator negative input.
	ICSPCLK	ST	—	ICSP Programming Clock.
RA2/AN2/T0CKI(1)/CWG1IN(1)/ INT(1)	RA2	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN2	AN	—	ADC Channel input.
	T0CKI	TTL/ST	—	Timer0 clock input.
	CWG1IN	TTL/ST	—	CWG complementary input.
	INT	TTL/ST	—	External interrupt.
RA3/VPP/$\overline{MCLR}$	RA3	TTL/ST	—	General purpose input with IOC and WPU.
	VPP	HV	—	Programming voltage.
	$\overline{MCLR}$	ST	—	Master Clear with internal pull-up.
RA4/AN3/T1G(1)/CLKOUT	RA4	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN3	AN	—	ADC Channel input.
	T1G	TTL/ST	—	Timer1 Gate input.
	CLKOUT	—	CMOS	FOSC/4 output.
RA5/CLKIN/T1CKI(1)	RA5	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	CLKIN	CMOS	—	External clock input (EC mode).
	T1CKI	TTL/ST	—	Timer1 clock input.
RB4/AN10	RB4	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN10	AN	—	ADC Channel input.
RB5/AN11/RX(1)	RB5	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN11	AN	—	ADC Channel input.
	RX	ST	—	USART asynchronous input.
RB6	RB6	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
RB7/CK	RB7	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	CK	ST	CMOS	USART synchronous clock.
RC0/AN4/C2IN+	RC0	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN4	AN	—	ADC Channel input.
	C2IN+	AN	—	Comparator positive input.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open-Drain; TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I^{2}C™ = Schmitt Trigger input with I^{2}C™ levels; 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 12-1.

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

TABLE 1-3: PIC16(L)F1578/9 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC1/AN5/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN5	AN	—	ADC Channel input.
	C1IN1-	AN	—	Comparator negative input.
	C2IN1-	AN	—	Comparator negative input.
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN6	AN	—	ADC Channel input.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
RC3/AN7/C1IN3-/C2IN3-	RC3	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	AN7	AN	—	ADC Channel input.
	C1IN3-	AN	—	Comparator negative input.
	C2IN3-	AN	—	Comparator negative input.
RC4/ADCACT(1)	RC4	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
	ADCACT	TTL/ST	—	ADC Auto-conversion Trigger input.
RC5	RC5	TTL/ST	CMOS/OD	General purpose input with IOC and WPU.
OUT(2)	C1OUT	—	CMOS	Comparator output.
	C2OUT	—	CMOS	Comparator output.
	PWM1OUT	—	CMOS	PWM1 output.
	PWM2OUT	—	CMOS	PWM2 output.
	PWM3OUT	—	CMOS	PWM3 output.
	PWM4OUT	—	CMOS	PWM4 output.
	CWG1A	—	CMOS	Complementary Output Generator Output A.
	CWG1B	—	CMOS	Complementary Output Generator Output B.
	TX/CK	—	CMOS	USART asynchronous TX data/synchronous clock output.
	DT(3)	—	CMOS	USART synchronous data output.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open-Drain; TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I²C™ = Schmitt Trigger input with I²C™ levels; 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 12-1.

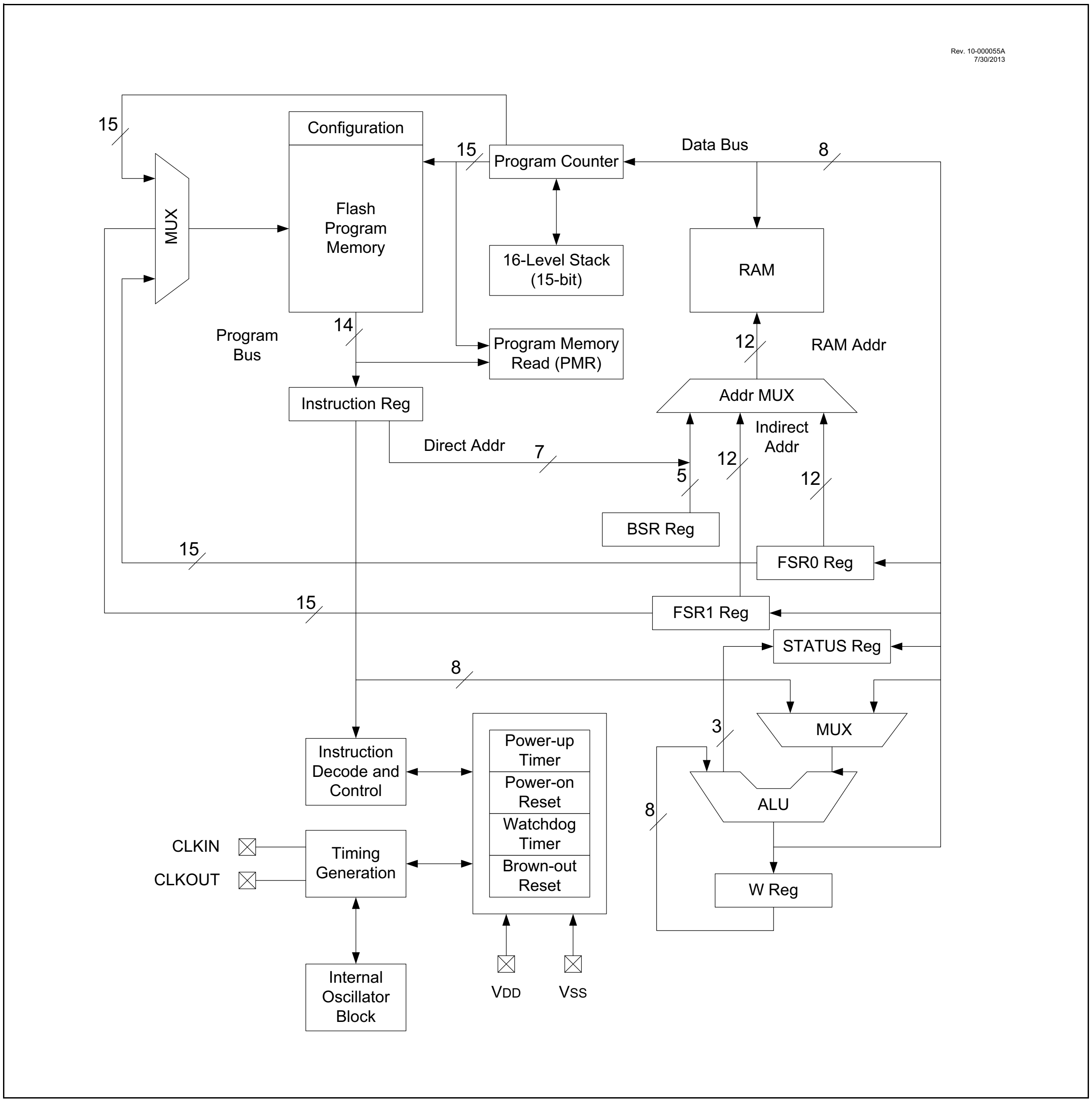
3: These USART functions are bidirectional. The output pin selections must be the same as the input pin selections.

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

FIGURE 2-1: CORE BLOCK DIAGRAM



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 26.0 "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 3.3.2 "Special Function Register"** for more information about using the SFR registers to read byte data stored in PFM.

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range (1)
PIC16(L)F1574/8	4,096	0FFFh	0F80h-0FFFh
PIC16(L)F1575/9	8,192	1FFFh	1F80h-1FFFh

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

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1574/8

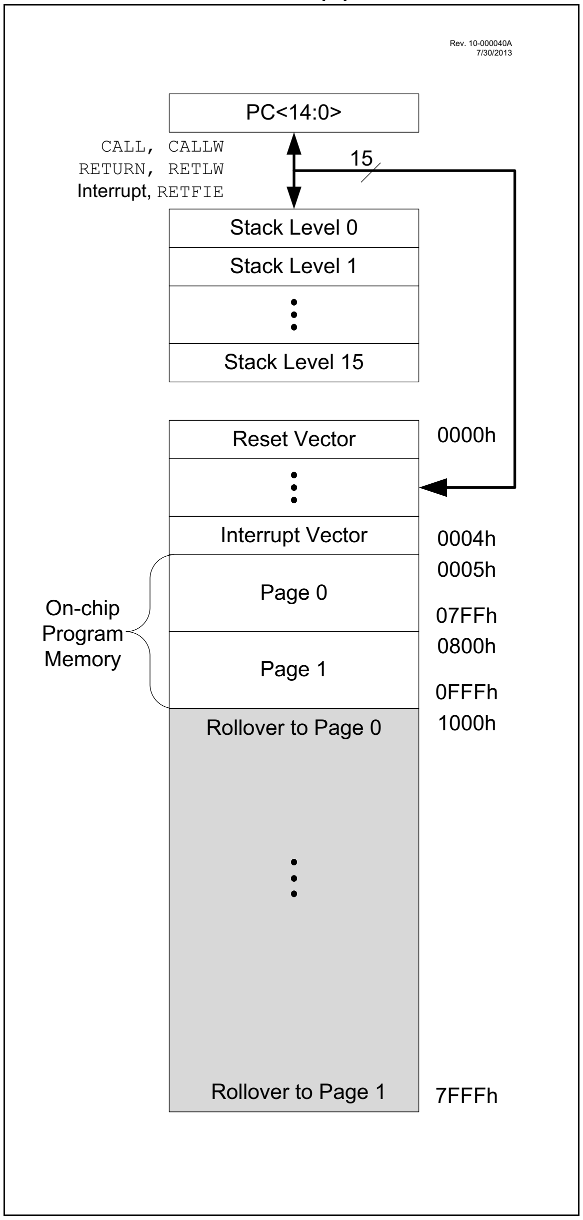
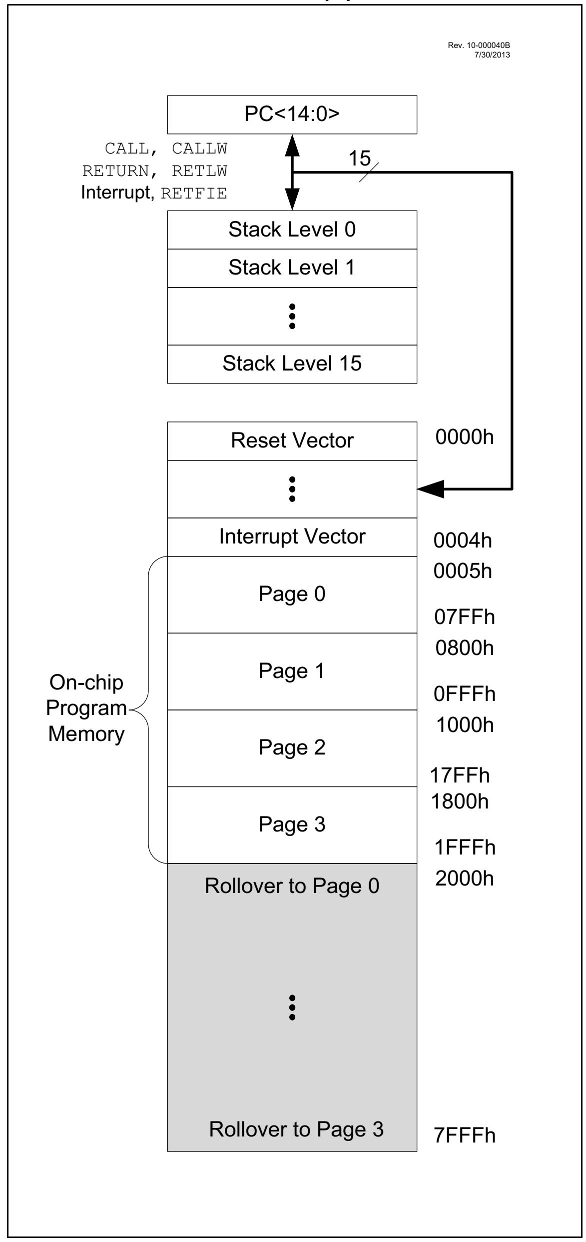


FIGURE 3-2: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1575/9



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 CODE…
    MOVLW      DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

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

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

```
constants
    RETLW DATA0          ;Index0 data
    RETLW DATA1          ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;… LOTS OF CODE…
    MOVLW   LOW constants
    MOVWF   FSR1L
    MOVLW   HIGH constants
    MOVWF   FSR1H
    MOVIW   0[FSR1]
;THE PROGRAM 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 3-3):

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

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.6 "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/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-14.

TABLE 3-2: CORE REGISTERS

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

3.3.1.1 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 $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (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 26.0 "Instruction Set Summary"**).

> **Note 1:** The C and DC bits operate as $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bits, respectively, in subtraction.

REGISTER 3-1: STATUS: STATUS REGISTER

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC(1)	C(1)
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '`0`'

bit 4 **$\overline{\text{TO}}$:** Time-Out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{\text{PD}}$:** 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/$\overline{\text{Digit Borrow}}$ 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/$\overline{\text{Borrow}}$ 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 $\overline{\text{Borrow}}$, 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 non-banked 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 as shown in Table 3-3 through Table 3-13.

FIGURE 3-3: BANKED MEMORY PARTITIONING

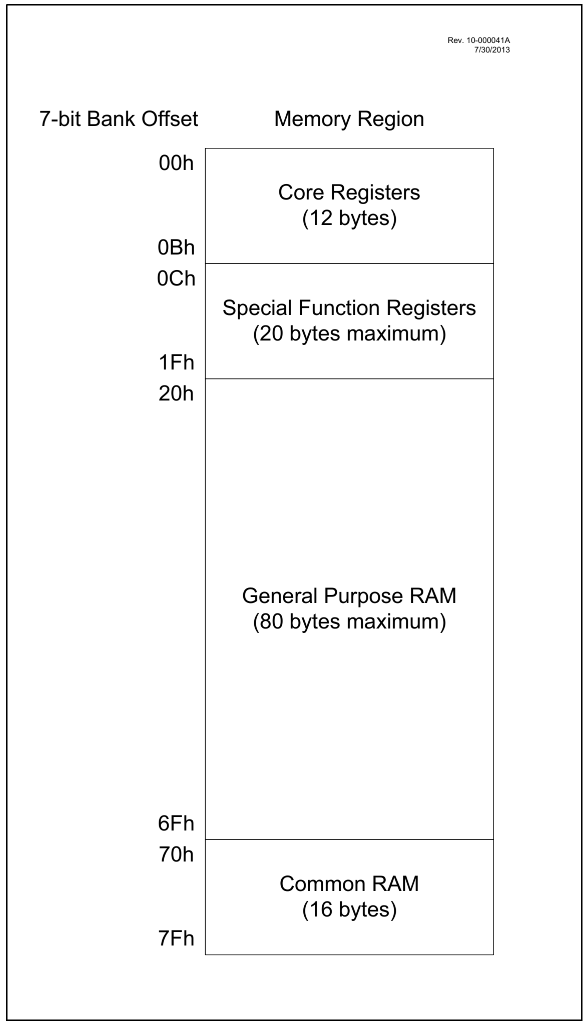


TABLE 3-3: PIC16(L)F1574 MEMORY MAP, BANKS 0-7

Addr	BANK0	Addr	BANK1	Addr	BANK2	Addr	BANK3	Addr	BANK4	Addr	BANK5	Addr	BANK6	Addr	BANK7
000h–00Bh	Core Registers (Table 3-2)	080h–08Bh	Core Registers (Table 3-2)	100h–10Bh	Core Registers (Table 3-2)	180h–18Bh	Core Registers (Table 3-2)	200h–20Bh	Core Registers (Table 3-2)	280h–28Bh	Core Registers (Table 3-2)	300h–30Bh	Core Registers (Table 3-2)	380h–38Bh	Core Registers (Table 3-2)
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	—	08Dh	—	10Dh	—	18Dh	—	20Dh	—	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	—	291h	—	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	—	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	—	293h	—	313h	—	393h	IOCAF
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h	—	294h	—	314h	—	394h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	—	295h	—	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	—	296h	—	316h	—	396h	—
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON(1)	217h	—	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	—	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	—	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	—	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	—
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h–06Fh	General Purpose Register 80 Bytes	0A0h–0EFh	General Purpose Register 80 Bytes	120h–16Fh	General Purpose Register 80 Bytes	1A0h–1EFh	General Purpose Register 80 Bytes	220h–26Fh	General Purpose Register 80 Bytes	2A0h–2EFh	General Purpose Register 80 Bytes	320h–32Fh 330h–36Fh	General Purpose Register 16 Bytes Unimplemented Read as '0'	3A0h–3EFh	Unimplemented Read as '0'
070h–07Fh	Common RAM	0F0h–0FFh	Accesses 70h – 7Fh	170h–17Fh	Accesses 70h – 7Fh	1F0h–1FFh	Accesses 70h – 7Fh	270h–27Fh	Accesses 70h – 7Fh	2F0h–2FFh	Accesses 70h – 7Fh	370h–37Fh	Accesses 70h – 7Fh	3F0h–3FFh	Accesses 70h – 7Fh

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

Note 1: Unimplemented on PIC16LF1574.

TABLE 3-4: PIC16(L)F1575 MEMORY MAP, BANKS 0-7

	BANK0		BANK1		BANK2		BANK3		BANK4		BANK5		BANK6		BANK7
000h 00Bh	Core Registers (Table 3-2)	080h 08Bh	Core Registers (Table 3-2)	100h 10Bh	Core Registers (Table 3-2)	180h 18Bh	Core Registers (Table 3-2)	200h 20Bh	Core Registers (Table 3-2)	280h 28Bh	Core Registers (Table 3-2)	300h 30Bh	Core Registers (Table 3-2)	380h 38Bh	Core Registers (Table 3-2)
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	—	08Dh	—	10Dh	—	18Dh	—	20Dh	—	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	—	291h	—	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	—	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	—	293h	—	313h	—	393h	IOCAF
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h	—	294h	—	314h	—	394h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	—	295h	—	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	—	296h	—	316h	—	396h	—
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON(1)	217h	—	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	—	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	—	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	—	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	—
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h 06Fh	General Purpose Register 80 Bytes	0A0h 0EFh	General Purpose Register 80 Bytes	120h 16Fh	General Purpose Register 80 Bytes	1A0h 1EFh	General Purpose Register 80 Bytes	220h 26Fh	General Purpose Register 80 Bytes	2A0h 2EFh	General Purpose Register 80 Bytes	320h 36Fh	General Purpose Register 80 Bytes	3A0h 3EFh	General Purpose Register 80 Bytes
070h 07Fh	Common RAM	0F0h 0FFh	Accesses 70h – 7Fh	170h 17Fh	Accesses 70h – 7Fh	1F0h 1FFh	Accesses 70h – 7Fh	270h 27Fh	Accesses 70h – 7Fh	2F0h 2FFh	Accesses 70h – 7Fh	370h 37Fh	Accesses 70h – 7Fh	3F0h 3FFh	Accesses 70h – 7Fh

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

Note 1: Unimplemented on PIC16LF1575.

TABLE 3-5: PIC16(L)F1578 MEMORY MAP, BANKS 0-7

	BANK0		BANK1		BANK2		BANK3		BANK4		BANK5		BANK6		BANK7
000h–00Bh	Core Registers (Table 3-2)	080h–08Bh	Core Registers (Table 3-2)	100h–10Bh	Core Registers (Table 3-2)	180h–18Bh	Core Registers (Table 3-2)	200h–20Bh	Core Registers (Table 3-2)	280h–28Bh	Core Registers (Table 3-2)	300h–30Bh	Core Registers (Table 3-2)	380h–38Bh	Core Registers (Table 3-2)
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCONB	38Dh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	—	291h	—	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	—	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	—	293h	—	313h	—	393h	IOCAF
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h	—	294h	—	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	—	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	—	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON[1]	217h	—	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	—	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	—	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	—	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	—
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h–06Fh	General Purpose Register 80 Bytes	0A0h–0EFh	General Purpose Register 80 Bytes	120h–16Fh	General Purpose Register 80 Bytes	1A0h–1EFh	General Purpose Register 80 Bytes	220h–26Fh	General Purpose Register 80 Bytes	2A0h–2EFh	General Purpose Register 80 Bytes	320h–32Fh	General Purpose Register 16 Bytes	3A0h–3EFh	Unimplemented Read as '0'
												330h–36Fh	Unimplemented Read as '0'		
070h–07Fh	Common RAM	0F0h–0FFh	Accesses 70h – 7Fh	170h–17Fh	Accesses 70h – 7Fh	1F0h–1FFh	Accesses 70h – 7Fh	270h–27Fh	Accesses 70h – 7Fh	2F0h–2FFh	Accesses 70h – 7Fh	370h–37Fh	Accesses 70h – 7Fh	3F0h–3FFh	Accesses 70h – 7Fh

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

Note 1: Unimplemented on PIC16LF1578.

TABLE 3-6: PIC16(L)F1579 MEMORY MAP, BANKS 0-7

Addr	BANK0	Addr	BANK1	Addr	BANK2	Addr	BANK3	Addr	BANK4	Addr	BANK5	Addr	BANK6	Addr	BANK7
000h–00Bh	Core Registers (Table 3-2)	080h–08Bh	Core Registers (Table 3-2)	100h–10Bh	Core Registers (Table 3-2)	180h–18Bh	Core Registers (Table 3-2)	200h–20Bh	Core Registers (Table 3-2)	280h–28Bh	Core Registers (Table 3-2)	300h–30Bh	Core Registers (Table 3-2)	380h–38Bh	Core Registers (Table 3-2)
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCONB	38Dh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	—	291h	—	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	—	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	—	293h	—	313h	—	393h	IOCAF
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h	—	294h	—	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	—	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	—	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON(1)	217h	—	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	—	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	—	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	—	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	—
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h–06Fh	General Purpose Register 80 Bytes	0A0h–0EFh	General Purpose Register 80 Bytes	120h–16Fh	General Purpose Register 80 Bytes	1A0h–1EFh	General Purpose Register 80 Bytes	220h–26Fh	General Purpose Register 80 Bytes	2A0h–2EFh	General Purpose Register 80 Bytes	320h–36Fh	General Purpose Register 80 Bytes	3A0h–3EFh	General Purpose Register 80 Bytes
070h–07Fh	Common RAM	0F0h–0FFh	Accesses 70h – 7Fh	170h–17Fh	Accesses 70h – 7Fh	1F0h–1FFh	Accesses 70h – 7Fh	270h–27Fh	Accesses 70h – 7Fh	2F0h–2FFh	Accesses 70h – 7Fh	370h–37Fh	Accesses 70h – 7Fh	3F0h–3FFh	Accesses 70h – 7Fh

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

Note 1: Unimplemented on PIC16LF1579.

TABLE 3-7: PIC16(L)F1574/8 MEMORY MAP, BANKS 8-15

Addr	BANK 8	Addr	BANK 9	Addr	BANK 10	Addr	BANK 11	Addr	BANK 12	Addr	BANK 13	Addr	BANK 14	Addr	BANK 15
400h–40Bh	Core Registers (Table 3-2)	480h–48Bh	Core Registers (Table 3-2)	500h–50Bh	Core Registers (Table 3-2)	580h–58Bh	Core Registers (Table 3-2)	600h–60Bh	Core Registers (Table 3-2)	680h–68Bh	Core Registers (Table 3-2)	700h–70Bh	Core Registers (Table 3-2)	780h–78Bh	Core Registers (Table 3-2)
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	CWG1DBR	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	CWG1CON0	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	CWG1CON1	714h	—	794h	—
415h	—	495h	—	515h	—	595h	—	615h	—	695h	CWG1CON2	715h	—	795h	—
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h–46Fh	Unimplemented Read as '0'	4A0h–4EFh	Unimplemented Read as '0'	520h–56Fh	Unimplemented Read as '0'	5A0h–5EFh	Unimplemented Read as '0'	620h–66Fh	Unimplemented Read as '0'	6A0h–6EFh	Unimplemented Read as '0'	720h–76Fh	Unimplemented Read as '0'	7A0h–7EFh	Unimplemented Read as '0'
470h–47Fh	Accesses 70h – 7Fh	4F0h–4FFh	Accesses 70h – 7Fh	570h–57Fh	Accesses 70h – 7Fh	5F0h–5FFh	Accesses 70h – 7Fh	670h–67Fh	Accesses 70h – 7Fh	6F0h–6FFh	Accesses 70h – 7Fh	770h–77Fh	Accesses 70h – 7Fh	7F0h–7FFh	Accesses 70h – 7Fh

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

TABLE 3-8: PIC16(L)F1575/9 MEMORY MAP, BANKS 8-15

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15
400h–40Bh	Core Registers (Table 3-2)	480h–48Bh	Core Registers (Table 3-2)	500h–50Bh	Core Registers (Table 3-2)	580h–58Bh	Core Registers (Table 3-2)	600h–60Bh	Core Registers (Table 3-2)	680h–68Bh	Core Registers (Table 3-2)	700h–70Bh	Core Registers (Table 3-2)	780h–78Bh	Core Registers (Table 3-2)
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	CWG1DBR	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	CWG1CON0	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	CWG1CON1	714h	—	794h	—
415h	—	495h	—	515h	—	595h	—	615h	—	695h	CWG1CON2	715h	—	795h	—
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h–46Fh	General Purpose Register 80 Bytes	4A0h–4EFh	General Purpose Register 80 Bytes	520h–56Fh	General Purpose Register 80 Bytes	5A0h–5EFh	General Purpose Register 80 Bytes	620h–63Fh	General Purpose Register 32 Bytes	6A0h–6EFh	Unimplemented Read as ‘0’	720h–76Fh	Unimplemented Read as ‘0’	7A0h–7EFh	Unimplemented Read as ‘0’
								640h–66Fh	Unimplemented Read as ‘0’						
470h–47Fh	Accesses 70h – 7Fh	4F0h–4FFh	Accesses 70h – 7Fh	570h–57Fh	Accesses 70h – 7Fh	5F0h–5FFh	Accesses 70h – 7Fh	670h–67Fh	Accesses 70h – 7Fh	6F0h–6FFh	Accesses 70h – 7Fh	770h–77Fh	Accesses 70h – 7Fh	7F0h–7FFh	Accesses 70h – 7Fh

Legend: ▢ = Unimplemented data memory locations, read as ‘0’

TABLE 3-9: PIC16(L)F1574/5/8/9 MEMORY MAP, BANKS 16-23

	BANK16		BANK17		BANK18		BANK19
800h 80Bh	Core Registers (Table 3-2)	880h 88Bh	Core Registers (Table 3-2)	900h 90Bh	Core Registers (Table 3-2)	980h 98Bh	Core Registers (Table 3-2)
80Ch	—	88Ch	—	90Ch	—	98Ch	—
80Dh	—	88Dh	—	90Dh	—	98Dh	—
80Eh	—	88Eh	—	90Eh	—	98Eh	—
80Fh	—	88Fh	—	90Fh	—	98Fh	—
810h	—	890h	—	910h	—	990h	—
811h	—	891h	—	911h	—	991h	—
812h	—	892h	—	912h	—	992h	—
813h	—	893h	—	913h	—	993h	—
814h	—	894h	—	914h	—	994h	—
815h	—	895h	—	915h	—	995h	—
816h	—	896h	—	916h	—	996h	—
817h	—	897h	—	917h	—	997h	—
818h	—	898h	—	918h	—	998h	—
819h	—	899h	—	919h	—	999h	—
81Ah	—	89Ah	—	91Ah	—	99Ah	—
81Bh	—	89Bh	—	91Bh	—	99Bh	—
81Ch	—	89Ch	—	91Ch	—	99Ch	—
81Dh	—	89Dh	—	91Dh	—	99Dh	—
81Eh	—	89Eh	—	91Eh	—	99Eh	—
81Fh	—	89Fh	—	91Fh	—	99Fh	—
820h 86Fh	Unimplemented Read as '0'	8A0h 8EFh	Unimplemented Read as '0'	920h 96Fh	Unimplemented Read as '0'	9A0h 9EFh	Unimplemented Read as '0'
870h 87Fh	Accesses 70h – 7Fh	8F0h 8FFh	Accesses 70h – 7Fh	970h 97Fh	Accesses 70h – 7Fh	9F0h 9FFh	Accesses 70h – 7Fh

	BANK20		BANK21		BANK22		BANK23
A00h A0Bh	Core Registers (Table 3-2)	A80h A8Bh	Core Registers (Table 3-2)	B00h B0Bh	Core Registers (Table 3-2)	B80h B8Bh	Core Registers (Table 3-2)
A0Ch	—	A8Ch	—	B0Ch	—	B8Ch	—
A0Dh	—	A8Dh	—	B0Dh	—	B8Dh	—
A0Eh	—	A8Eh	—	B0Eh	—	B8Eh	—
A0Fh	—	A8Fh	—	B0Fh	—	B8Fh	—
A10h	—	A90h	—	B10h	—	B90h	—
A11h	—	A91h	—	B11h	—	B91h	—
A12h	—	A92h	—	B12h	—	B92h	—
A13h	—	A93h	—	B13h	—	B93h	—
A14h	—	A94h	—	B14h	—	B94h	—
A15h	—	A95h	—	B15h	—	B95h	—
A16h	—	A96h	—	B16h	—	B96h	—
A17h	—	A97h	—	B17h	—	B97h	—
A18h	—	A98h	—	B18h	—	B98h	—
A19h	—	A99h	—	B19h	—	B99h	—
A1Ah	—	A9Ah	—	B1Ah	—	B9Ah	—
A1Bh	—	A9Bh	—	B1Bh	—	B9Bh	—
A1Ch	—	A9Ch	—	B1Ch	—	B9Ch	—
A1Dh	—	A9Dh	—	B1Dh	—	B9Dh	—
A1Eh	—	A9Eh	—	B1Eh	—	B9Eh	—
A1Fh	—	A9Fh	—	B1Fh	—	B9Fh	—
A20h A6Fh	Unimplemented Read as '0'	AA0h AEFh	Unimplemented Read as '0'	B20h B6Fh	Unimplemented Read as '0'	BA0h BEFh	Unimplemented Read as '0'
A70h A7Fh	Accesses 70h – 7Fh	AF0h AFFh	Accesses 70h – 7Fh	B70h B7Fh	Accesses 70h – 7Fh	BF0h BFFh	Accesses 70h – 7Fh

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

TABLE 3-10: PIC16(L)F1574/5/8/9 MEMORY MAP, BANKS 24-31

	BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31
C00h–C0Bh	Core Registers (Table 3-2)	C80h–C8Bh	Core Registers (Table 3-2)	D00h–D0Bh	Core Registers (Table 3-2)	D80h–D8Bh	Core Registers (Table 3-2)	E00h–E0Bh	Core Registers (Table 3-2)	E80h–E8Bh	Core Registers (Table 3-2)	F00h–F0Bh	Core Registers (Table 3-2)	F80h–F8Bh	Core Registers (Table 3-2)
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch–DEFh	See Table 3-11	E0Ch–E6Fh	See Table 3-12	E8Ch–EEFh	See Table 3-12	F0Ch	—	F8Ch–FEFh	See Table 3-13
C0Dh	—	C8Dh	—	D0Dh	—							F0Dh	—		
C0Eh	—	C8Eh	—	D0Eh	—							F0Eh	—		
C0Fh	—	C8Fh	—	D0Fh	—							F0Fh	—		
C10h	—	C90h	—	D10h	—							F10h	—		
C11h	—	C91h	—	D11h	—							F11h	—		
C12h	—	C92h	—	D12h	—							F12h	—		
C13h	—	C93h	—	D13h	—							F13h	—		
C14h	—	C94h	—	D14h	—							F14h	—		
C15h	—	C95h	—	D15h	—							F15h	—		
C16h	—	C96h	—	D16h	—							F16h	—		
C17h	—	C97h	—	D17h	—							F17h	—		
C18h	—	C98h	—	D18h	—							F18h	—		
C19h	—	C99h	—	D19h	—							F19h	—		
C1Ah	—	C9Ah	—	D1Ah	—							F1Ah	—		
C1Bh	—	C9Bh	—	D1Bh	—							F1Bh	—		
C1Ch	—	C9Ch	—	D1Ch	—							F1Ch	—		
C1Dh	—	C9Dh	—	D1Dh	—							F1Dh	—		
C1Eh	—	C9Eh	—	D1Eh	—							F1Eh	—		
C1Fh	—	C9Fh	—	D1Fh	—							F1Fh	—		
C20h–C6Fh	Unimplemented Read as '0'	CA0h–CEFh	Unimplemented Read as '0'	D20h–D6Fh	Unimplemented Read as '0'							F20h–F6Fh	Unimplemented Read as '0'		
C70h–CFFh	Accesses 70h – 7Fh	CF0h–CFFh	Accesses 70h – 7Fh	D70h–D7Fh	Accesses 70h – 7Fh	DF0h–DFFh	Accesses 70h – 7Fh	E70h–E7Fh	Accesses 70h – 7Fh	EF0h–EFFh	Accesses 70h – 7Fh	F70h–F7Fh	Accesses 70h – 7Fh	FF0h–FFFh	Accesses 70h – 7Fh

Legend: (shaded) = Unimplemented data memory locations, read as '0'

TABLE 3-11: PIC16(L)F1574/5/8/9 MEMORY MAP, BANK 27

Address	Bank 27
D8Ch	—
D8Dh	—
D8Eh	PWMEN
D8Fh	PWMLD
D90h	PWMOUT
D91h	PWM1PHL
D92h	PWM1PHH
D93h	PWM1DCL
D94h	PWM1DCH
D95h	PWM1PRL
D96h	PWM1PRH
D97h	PWM1OFL
D98h	PWM1OFH
D99h	PWM1TMRL
D9Ah	PWM1TMRH
D9Bh	PWM1CON
D9Ch	PWM1INTE
D9Dh	PWM1INTF
D9Eh	PWM1CLKCON
D9Fh	PWM1LDCON
DA0h	PWM1OFCON
DA1h	PWM2PHL
DA2h	PWM2PHH
DA3h	PWM2DCL
DA4h	PWM2DCH
DA5h	PWM2PRL
DA6h	PWM2PRH
DA7h	PWM2OFL
DA8h	PWM2OFH
DA9h	PWM2TMRL
DAAh	PWM2TMRH
DABh	PWM2CON
DACh	PWM2INTE
DADh	PWM2INTF
DAEh	PWM2CLKCON
DAFh	PWM2LDCON
DB0h	PWM2OFCON
DB1h	PWM3PHL
DB2h	PWM3PHH
DB3h	PWM3DCL
DB4h	PWM3DCH
DB5h	PWM3PRL
DB6h	PWM3PRH
DB7h	PWM3OFL
DB8h	PWM3OFH
DB9h	PWM3TMRL
DBAh	PWM3TMRH
DBBh	PWM3CON
DBCh	PWM3INTE
DBDh	PWM3INTF
DBEh	PWM3CLKCON
DBFh	PWM3LDCON
DC0h	PWM3OFCON
DC1h	PWM4PHL
DC2h	PWM4PHH
DC3h	PWM4DCL
DC4h	PWM4DCH
DC5h	PWM4PRL
DC6h	PWM4PRH
DC7h	PWM4OFL
DC8h	PWM4OFH
DC9h	PWM4TMRL
DCAh	PWM4TMRH
DCBh	PWM4CON
DCCh	PWM4INTE
DCDh	PWM4INTF
DCEh	PWM4CLKCON
DCFh	PWM4LDCON
DD0h	PWM4OFCON
DD1h–DEFh	—

Legend: ▒ = Unimplemented data memory locations, read as ‘0’.

TABLE 3-12: PIC16(L)F1574/5/8/9 MEMORY MAP, BANK 28-29

Address	Bank 28	Address	Bank 29
E0Ch	—	E8Ch	—
E0Dh	—	E8Dh	—
E0Eh	—	E8Eh	—
E0Fh	PPSLOCK	E8Fh	—
E10h	INTPPS	E90h	RA0PPS
E11h	T0CKIPPS	E91h	RA1PPS
E12h	T1CKIPPS	E92h	RA2PPS
E13h	T1GPPS	E93h	—
E14h	CWG1PPS	E94h	RA4PPS
E15h	RXPPS	E95h	RA5PPS
E16h	CKPPS	E96h	—
E17h	ADCACTPPS	E97h	—
E18h	—	E98h	—
E19h	—	E99h	—
E1Ah	—	E9Ah	—
E1Bh	—	E9Bh	—
E1Ch	—	E9Ch	RB4PPS(1)
E1Dh	—	E9Dh	RB5PPS(1)
E1Eh	—	E9Eh	RB6PPS(1)
E1Fh	—	E9Fh	RB7PPS(1)
E20h	—	EA0h	RC0PPS
E21h	—	EA1h	RC1PPS
E22h	—	EA2h	RC2PPS
E23h	—	EA3h	RC3PPS
E24h	—	EA4h	RC4PPS
E25h	—	EA5h	RC5PPS
E26h	—	EA6h	RC6PPS(1)
E27h	—	EA7h	RC7PPS(1)
E28h	—	EA8h	—
E29h	—	EA9h	—
E2Ah	—	EAAh	—
E2Bh	—	EABh	—
E2Ch	—	EACh	—
E2Dh	—	EADh	—
E2Eh	—	EAEh	—
E2Fh	—	EAFh	—
E30h	—	EB0h	—
E31h	—	EB1h	—
E32h	—	EB2h	—
E33h	—	EB3h	—
E34h	—	EB4h	—
E35h	—	EB5h	—
E36h	—	EB6h	—
E37h	—	EB7h	—
E38h	—	EB8h	—
E39h	—	EB9h	—
E3Ah	—	EBAh	—
E3Bh	—	EBBh	—
E3Ch	—	EBCh	—
E3Dh	—	EBDh	—
E3Eh	—	EBEh	—
E3Fh	—	EBFh	—
E40h–E6Fh	—	EC0h–EEFh	—

Legend: ▒ = Unimplemented data memory locations, read as ‘0’

Note 1: Unimplemented on PIC16(L)F1574/5.

TABLE 3-13: PIC16(L)F1574/5/8/9 MEMORY MAP, BANK 31

Address	Bank 31
F8Ch – FE3h	Unimplemented Read as '0'
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

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

3.3.6 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-14 can be addressed from any Bank.

TABLE 3-14: CORE FUNCTION REGISTERS SUMMARY

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 0-31											
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								`xxxx xxxx`	`uuuu uuuu`
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								`xxxx xxxx`	`uuuu uuuu`
x02h or x82h	PCL	Program Counter (PC) Least Significant Byte								`0000 0000`	`0000 0000`
x03h or x83h	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	`---1 1000`	`---q quuu`
x04h or x84h	FSR0L	Indirect Data Memory Address 0 Low Pointer								`0000 0000`	`uuuu uuuu`
x05h or x85h	FSR0H	Indirect Data Memory Address 0 High Pointer								`0000 0000`	`0000 0000`
x06h or x86h	FSR1L	Indirect Data Memory Address 1 Low Pointer								`0000 0000`	`uuuu uuuu`
x07h or x87h	FSR1H	Indirect Data Memory Address 1 High Pointer								`0000 0000`	`0000 0000`
x08h or x88h	BSR	—	—	—	BSR<4:0>					`---0 0000`	`---0 0000`
x09h or x89h	WREG	Working Register								`0000 0000`	`uuuu uuuu`
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							`-000 0000`	`-000 0000`
x0Bh or x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	`0000 0000`	`0000 0000`

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

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 0											
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx
00Dh	PORTB(1)	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	xxxx ----
00Eh	PORTC	RC7(1)	RC6(1)	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	xxxx xxxx
00Fh	—	Unimplemented								—	—
010h	—	Unimplemented								—	—
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	0000 --00	0000 --00
012h	PIR2	—	C2IF	C1IF	—	—	—	—	—	-00- ----	-00- ----
013h	PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	0000 ----	0000 ----
014h	—									—	—
015h	TMR0	Holding Register for the 8-bit Timer0 Count								xxxx xxxx	uuuu uuuu
016h	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count								xxxx xxxx	uuuu uuuu
017h	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								xxxx xxxx	uuuu uuuu
018h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON	0000 -0-0	uuuu -u-u
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu
01Ah	TMR2	Timer2 Module Register								0000 0000	0000 0000
01Bh	PR2	Timer2 Period Register								1111 1111	1111 1111
01Ch	T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
01Dh	—	Unimplemented								—	—
01Eh	—	Unimplemented								—	—
01Fh	—	Unimplemented								—	—

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

Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 1											
08Ch	TRISA	—	—	TRISA5	TRISA4	—(3)	TRISA2	TRISA1	TRISA0	`--11 1111`	`--11 1111`
08Dh	TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	`1111 ----`	`1111 ----`
08Eh	TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`1111 1111`	`1111 1111`
08Fh	—	Unimplemented								—	—
090h	—	Unimplemented								—	—
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	`0000 --00`	`0000 --00`
092h	PIE2	—	C2IE	C1IE	—	—	—	—	—	`-00- ----`	`-00- ----`
093h	PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	`0000 ----`	`0000 ----`
094h	—									—	—
095h	OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			`1111 1111`	`1111 1111`
096h	PCON	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	`00-1 11qq`	`qq-q qquu`
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	`--01 0110`	`--01 0110`
098h	OSCTUNE	—	—	TUN<5:0>						`--00 0000`	`--00 0000`
099h	OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		`0011 1-00`	`0011 1-00`
09Ah	OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	`-0q0 0q00`	`-qqq qqqq`
09Bh	ADRESL	ADC Result Register Low								`xxxx xxxx`	`uuuu uuuu`
09Ch	ADRESH	ADC Result Register High								`xxxx xxxx`	`uuuu uuuu`
09Dh	ADCON0	—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON	`-000 0000`	`-000 0000`
09Eh	ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		`0000 --00`	`0000 --00`
09Fh	ADCON2	TRIGSEL<3:0>				—	—	—	—	`0000 ----`	`0000 ----`

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

Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as ‘1’.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 2											
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	`--xx -xxx`	`--uu -uuu`
10Dh	LATB(1)	LATB7	LATB6	LATB5	LATB4	—	—	—	—	`xxxx ----`	`xxxx ----`
10Eh	LATC	LATC7(1)	LATC6(1)	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	`xxxx xxxx`	`xxxx xxxx`
10Fh	—	Unimplemented								—	—
110h	—	Unimplemented								—	—
111h	CM1CON0	C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	`00-0 -100`	`00-0 -100`
112h	CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			`0000 -000`	`0000 -000`
113h	CM2CON0	C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	`00-0 -100`	`00-0 -100`
114h	CM2CON1	C2INTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			`0000 -000`	`0000 -000`
115h	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	`---- --00`	`---- --00`
116h	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	`10-- ---q`	`uu-- ---u`
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		`0q00 0000`	`0q00 0000`
118h	DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>		—	—	`0-0- 00--`	`0-0- 00--`
119h	DACCON1	—	—	—	DACR<4:0>					`---0 0000`	`---0 0000`
11Ah to 11Fh	—	Unimplemented								—	—

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

Note 1: PIC16(L)F1578/9 only.

2: PIC16F1574/5/8/9 only.

3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 3											
18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 -111	---1 -111
18Dh	ANSELB(1)	—	—	ANSB5	ANSB4	—	—	—	—	--11 ----	--11 ----
18Eh	ANSELC	ANSC7(1)	ANSC6(1)	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111
18Fh	—	Unimplemented								—	—
190h	—	Unimplemented								—	—
191h	PMADRL	Flash Program Memory Address Register Low Byte								0000 0000	0000 0000
192h	PMADRH	—(3)	Flash Program Memory Address Register High Byte							1000 0000	1000 0000
193h	PMDATL	Flash Program Memory Read Data Register Low Byte								xxxx xxxx	uuuu uuuu
194h	PMDATH	—	—	Flash Program Memory Read Data Register High Byte						--xx xxxx	--uu uuuu
195h	PMCON1	—(3)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000
196h	PMCON2	Flash Program Memory Control Register 2								0000 0000	0000 0000
197h	VREGCON(2)	—	—	—	—	—	—	VREGPM	Reserved	---- --01	---- --01
198h	—	Unimplemented								—	—
199h	RCREG	USART Receive Data Register								0000 0000	0000 0000
19Ah	TXREG	USART Transmit Data Register								0000 0000	0000 0000
19Bh	SPBRGL	Baud Rate Generator Data Register Low								0000 0000	0000 0000
19Ch	SPBRGH	Baud Rate Generator Data Register High								0000 0000	0000 0000
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
19Fh	BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00

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

Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 4											
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	`--11 1111`	`--11 1111`
20Dh	WPUB(1)	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	`1111 ----`	`1111 ----`
20Eh	WPUC	WPUC7(1)	WPUC6(1)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	`1111 1111`	`1111 1111`
20Fh to 21Fh	—	Unimplemented								—	—
Bank 5											
28Ch	ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	`--00 -000`	`--00 -000`
28Dh	ODCONB(1)	ODB7	ODB6	ODB5	ODB4	—	—	—	—	`0000 ----`	`0000 ----`
28Eh	ODCONC	ODC7(1)	ODC6(1)	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	`0000 0000`	`0000 0000`
28Fh to 29Fh	—	Unimplemented								—	—
Bank 6											
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	`--11 -111`	`--11 -111`
30Dh	SLRCONB(1)	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	`1111 ----`	`1111 ----`
30Eh	SLRCONC	SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	`1111 1111`	`1111 1111`
30Fh to 31Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.
Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 7											
38Ch	INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	`--11 1111`	`--11 1111`
38Dh	INLVLB(1)	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	`1111 ----`	`1111 ----`
38Eh	INLVLC	INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	`1111 1111`	`1111 1111`
38Fh to 390h	—	Unimplemented								—	—
391h	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	`--00 0000`	`--00 0000`
392h	IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	`--00 0000`	`--00 0000`
393h	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	`--00 0000`	`--00 0000`
394h	IOCBP(1)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	`0000 ----`	`--00 ----`
395h	IOCBN(1)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	`0000 ----`	`--00 ----`
396h	IOCBF(1)	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	`0000 ----`	`--00 ----`
397h	IOCCP	IOCCP7(1)	IOCCP6(1)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	`0000 0000`	`0000 0000`
398h	IOCCN	IOCCN7(1)	IOCCN6(1)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	`0000 0000`	`0000 0000`
399h	IOCCF	IOCCF7(1)	IOCCF6(1)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	`0000 0000`	`0000 0000`
39Ah to 39Fh	—	Unimplemented								—	—
Bank 8											
40Ch to 41Fh	—	Unimplemented								—	—
Bank 9											
48Ch to 49Fh	—	Unimplemented								—	—

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

Note 1: PIC16(L)F1578/9 only.

2: PIC16F1574/5/8/9 only.

3: Unimplemented, read as ‘`1`’.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 10											
50Ch to 51Fh	—	Unimplemented								—	—
Bank 11											
58Ch to 59Fh	—	Unimplemented								—	—
Bank 12											
60Ch to 61Fh	—	Unimplemented								—	—
Bank 13											
68Ch to 690h	—	Unimplemented								—	—
691h	CWG1DBR	—	—	CWG1DBR<5:0>						`--00 0000`	`--00 0000`
692h	CWG1DBF	—	—	CWG1DBF<5:0>						`--xx xxxx`	`--xx xxxx`
693h	CWG1CON0	G1EN	—	—	G1POLB	G1POLA	—	—	G1CS0	`0--0 0--0`	`0--0 0--0`
694h	CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	G1IS<2:0>			`0000 -000`	`0000 -000`
695h	CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC2	G1ASDSC1	G1ASDSPPS	—	`00-- 000-`	`00-- 000-`
696h to 69Fh	—	Unimplemented								—	—
Banks 14-26											
x0Ch/ x8Ch — x1Fh/ x9Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as ‘0’.
Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as ‘1’.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 27											
D8Ch	—	Unimplemented								—	—
D8Dh	—	Unimplemented								—	—
D8Eh	PWMEN	—	—	—	—	PWM4EN_A	PWM3EN_A	PWM2EN_A	PWM1EN_A	---- 0000	---- 0000
D8Fh	PWMLD	—	—	—	—	PWM4LDA_A	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A	---- 0000	---- 0000
D90h	PWMOUT	—	—	—	—	PWM4OUT_A	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A	---- 0000	---- 0000
D91h	PWM1PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
D92h	PWM1PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
D93h	PWM1DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
D94h	PWM1DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
D95h	PWM1PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
D96h	PWM1PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
D97h	PWM1OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
D98h	PWM1OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
D99h	PWM1TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu
D9Ah	PWM1TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu
D9Bh	PWM1CON	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--
D9Ch	PWM1INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000
D9Dh	PWM1INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000
D9Eh	PWM1CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 -000	-000 --00
D9Fh	PWM1LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- -000	00-- --00
DA0h	PWM1OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 -000	-000 --00
DA1h	PWM2PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
DA2h	PWM2PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
DA3h	PWM2DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
DA4h	PWM2DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
DA5h	PWM2PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
DA6h	PWM2PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
DA7h	PWM2OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
DA8h	PWM2OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
DA9h	PWM2TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu
DAAh	PWM2TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu

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

Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 27 (Continued)											
DABh	PWM2CON	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--
DACh	PWM2INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000
DADh	PWM2INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000
DAEh	PWM2CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 -000	-000 --00
DAFh	PWM2LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- -000	00-- --00
DB0h	PWM2OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 -000	-000 --00
DB1h	PWM3PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
DB2h	PWM3PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
DB3h	PWM3DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
DB4h	PWM3DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
DB5h	PWM3PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
DB6h	PWM3PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
DB7h	PWM3OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
DB8h	PWM3OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
DB9h	PWM3TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu
DBAh	PWM3TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu
DBBh	PWM3CON	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--
DBCh	PWM3INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000
DBDh	PWM3INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000
DBEh	PWM3CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 -000	-000 --00
DBFh	PWM3LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- -000	00-- --00
DC0h	PWM3OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 -000	-000 --00
DC1h	PWM4PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
DC2h	PWM4PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
DC3h	PWM4DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
DC4h	PWM4DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
DC5h	PWM4PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
DC6h	PWM4PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
DC7h	PWM4OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
DC8h	PWM4OFH	OF<15:8>								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as ‘0’.
Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as ‘1’.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 27 (Continued)											
DC9h	PWM4TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu
DCAh	PWM4TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu
DCBh	PWM4CON	EN	—	OUT	POL	MODE<1:0>		—	—	0000 00--	0000 00--
DCCh	PWM4INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000
DCDh	PWM4INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000
DCEh	PWM4CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 -000	-000 --00
DCFh	PWM4LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- -000	00-- --00
DD0h	PWM4OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 -000	-000 --00
DD1h to DEFh	—	Unimplemented								—	—
Bank 28											
E0Ch — E0Eh	—	Unimplemented								—	—
E0Fh	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	---- ---0	---- ---0
E10h	INTPPS	—	—	—	INTPPS<4:0>					---0 0010	---u uuuu
E11h	T0CKIPPS	—	—	—	T0CKIPPS<4:0>					---0 0010	---u uuuu
E12h	T1CKIPPS	—	—	—	T1CKIPPS<4:0>					---0 0101	---u uuuu
E13h	T1GPPS	—	—	—	T1GPPS<4:0>					---0 0100	---u uuuu
E14h	CWG1INPPS	—	—	—	CWGINPPS<4:0>					---0 0010	---u uuuu
E15h	RXPPS	—	—	—	RXPPS<4:0>					---1 0101	---u uuuu
E16h	CKPPS	—	—	—	CKPPS<4:0>					---1 0101	---u uuuu
E17h	ADCACTPPS	—	—	—	ADCACTPPS<4:0>					---1 0101	---u uuuu
E18h to E6Fh	—	Unimplemented								—	—

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

Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 29											
E8Ch — E8Fh	—	Unimplemented								—	—
E90h	RA0PPS	—	—	—	RA0PPS<4:0>					---0 0000	---u uuuu
E91h	RA1PPS	—	—	—	RA1PPS<4:0>					---0 0000	---u uuuu
E92h	RA2PPS	—	—	—	RA2PPS<4:0>					---0 0000	---u uuuu
E93h	—	Unimplemented								—	—
E94h	RA4PPS	—	—	—	RA4PPS<4:0>					---0 0000	---u uuuu
E95h	RA5PPS	—	—	—	RA5PPS<4:0>					---0 0000	---u uuuu
E96h — E9Bh	—	Unimplemented								—	—
E9Ch	RB4PPS(1)	—	—	—	RB4PPS<4:0>					---0 0000	---u uuuu
E9Dh	RB5PPS(1)	—	—	—	RB5PPS<4:0>					---0 0000	---u uuuu
E9Eh	RB6PPS(1	—	—	—	RB6PPS<4:0>					---0 0000	---u uuuu
E9Fh	RB7PPS(1)	—	—	—	RB7PPS<4:0>					---0 0000	---u uuuu
EA0h	RC0PPS	—	—	—	RC0PPS<4:0>					---0 0000	---u uuuu
EA1h	RC1PPS	—	—	—	RC1PPS<4:0>					---0 0000	---u uuuu
EA2h	RC2PPS	—	—	—	RC2PPS<4:0>					---0 0000	---u uuuu
EA3h	RC3PPS	—	—	—	RC3PPS<4:0>					---0 0000	---u uuuu
EA4h	RC4PPS	—	—	—	RC4PPS<4:0>					---0 0000	---u uuuu
EA5h	RC5PPS	—	—	—	RC5PPS<4:0>					---0 0000	---u uuuu
EA6h	RC6PPS(1)	—	—	—	RC6PPS<4:0>					---0 0000	---u uuuu
EA7h	RC7PPS(1)	—	—	—	RC7PPS<4:0>					---0 0000	---u uuuu
EA8h — EEFh	—	Unimplemented								—	—
Bank 30											
F0Ch — F1Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.
Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

TABLE 3-15: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

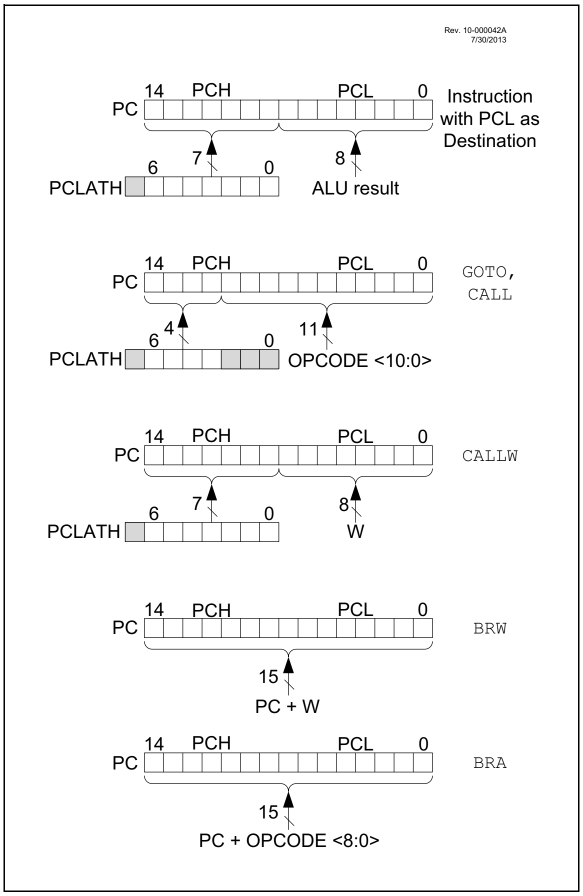
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 31											
F8Ch — FE3h	—	Unimplemented								—	—
FE4h	STATUS_SHAD	—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	`---- -xxx`	`---- -uuu`
FE5h	WREG_SHAD	Working Register Shadow								`xxxx xxxx`	`uuuu uuuu`
FE6h	BSR_SHAD	—	—	—	Bank Select Register Shadow					`---x xxxx`	`---u uuuu`
FE7h	PCLATH_SHAD	—	Program Counter Latch High Register Shadow							`-xxx xxxx`	`uuuu uuuu`
FE8h	FSR0L_SHAD	Indirect Data Memory Address 0 Low Pointer Shadow								`xxxx xxxx`	`uuuu uuuu`
FE9h	FSR0H_SHAD	Indirect Data Memory Address 0 High Pointer Shadow								`xxxx xxxx`	`uuuu uuuu`
FEAh	FSR1L_SHAD	Indirect Data Memory Address 1 Low Pointer Shadow								`xxxx xxxx`	`uuuu uuuu`
FEBh	FSR1H_SHAD	Indirect Data Memory Address 1 High Pointer Shadow								`xxxx xxxx`	`uuuu uuuu`
FECh	—	Unimplemented								—	—
FEDh	STKPTR	—	—	—	Current Stack Pointer					`---1 1111`	`---1 1111`
FEEh	TOSL	Top-of-Stack Low byte								`xxxx xxxx`	`uuuu uuuu`
FEFh	TOSH	—	Top-of-Stack High byte							`-xxx xxxx`	`-uuu uuuu`

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.
Note 1: PIC16(L)F1578/9 only.
2: PIC16F1574/5/8/9 only.
3: Unimplemented, read as '1'.

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 3-4 shows the five situations for the loading of the PC.

FIGURE 3-4: 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 and those being written to the PCL 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 3-5 through 3-8). 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 3-5 through Figure 3-8 for examples of accessing the stack.

FIGURE 3-5: ACCESSING THE STACK EXAMPLE 1

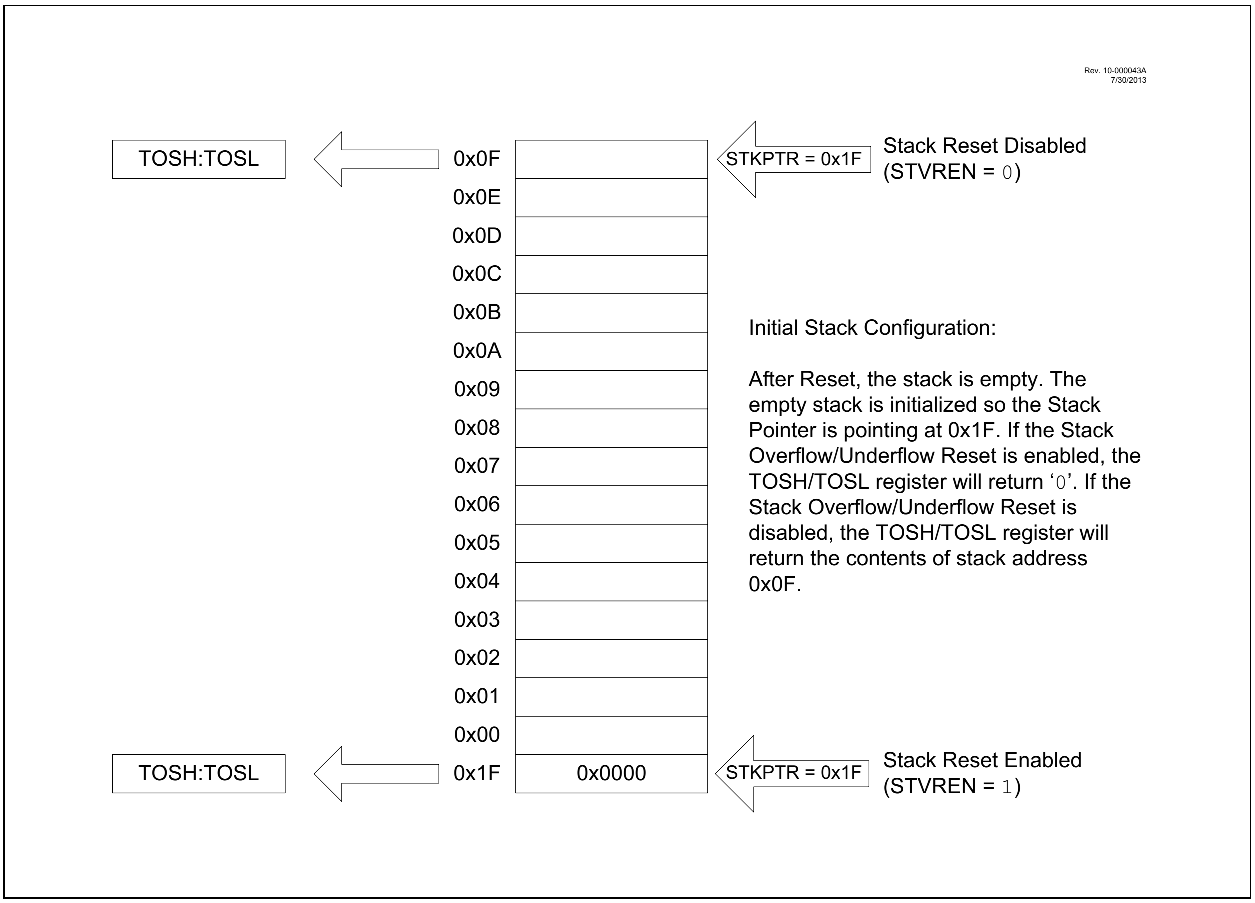


FIGURE 3-6: ACCESSING THE STACK EXAMPLE 2

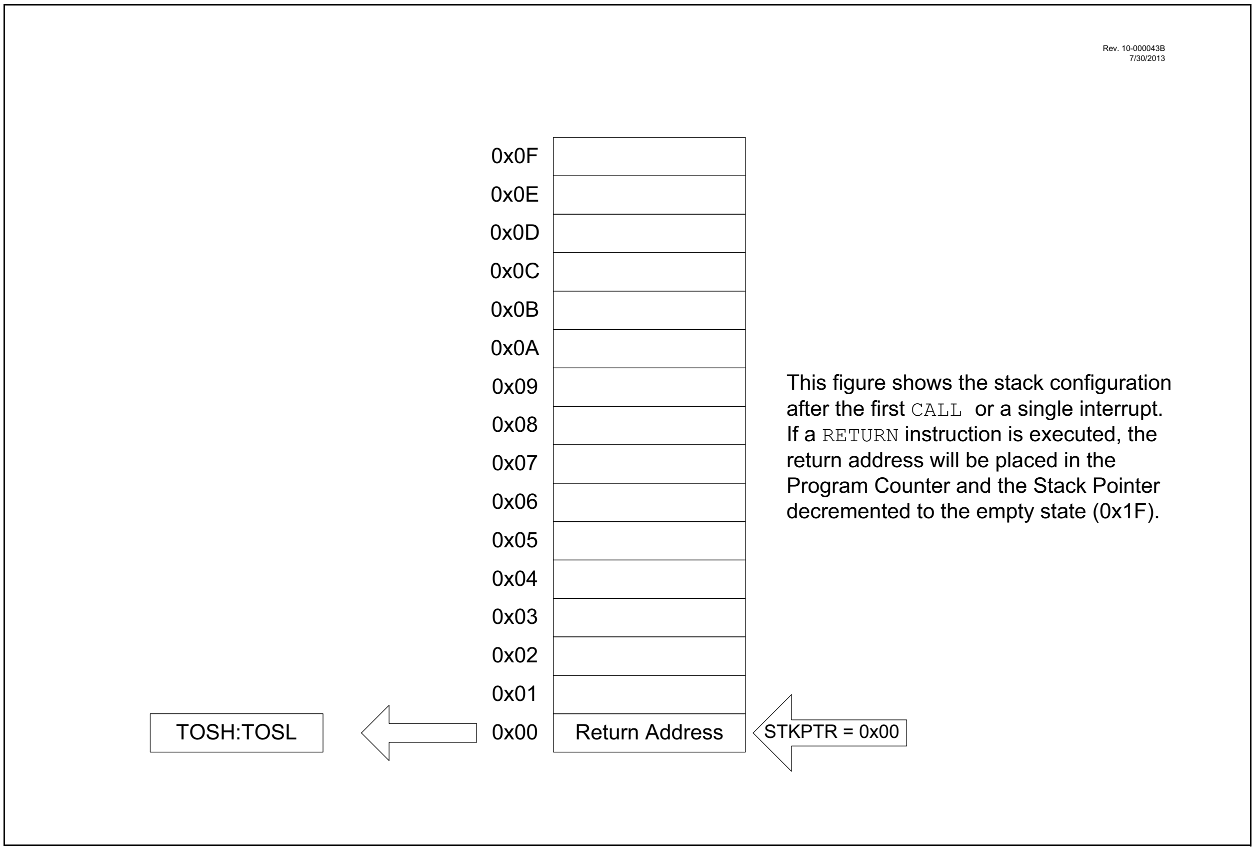


FIGURE 3-7: ACCESSING THE STACK EXAMPLE 3

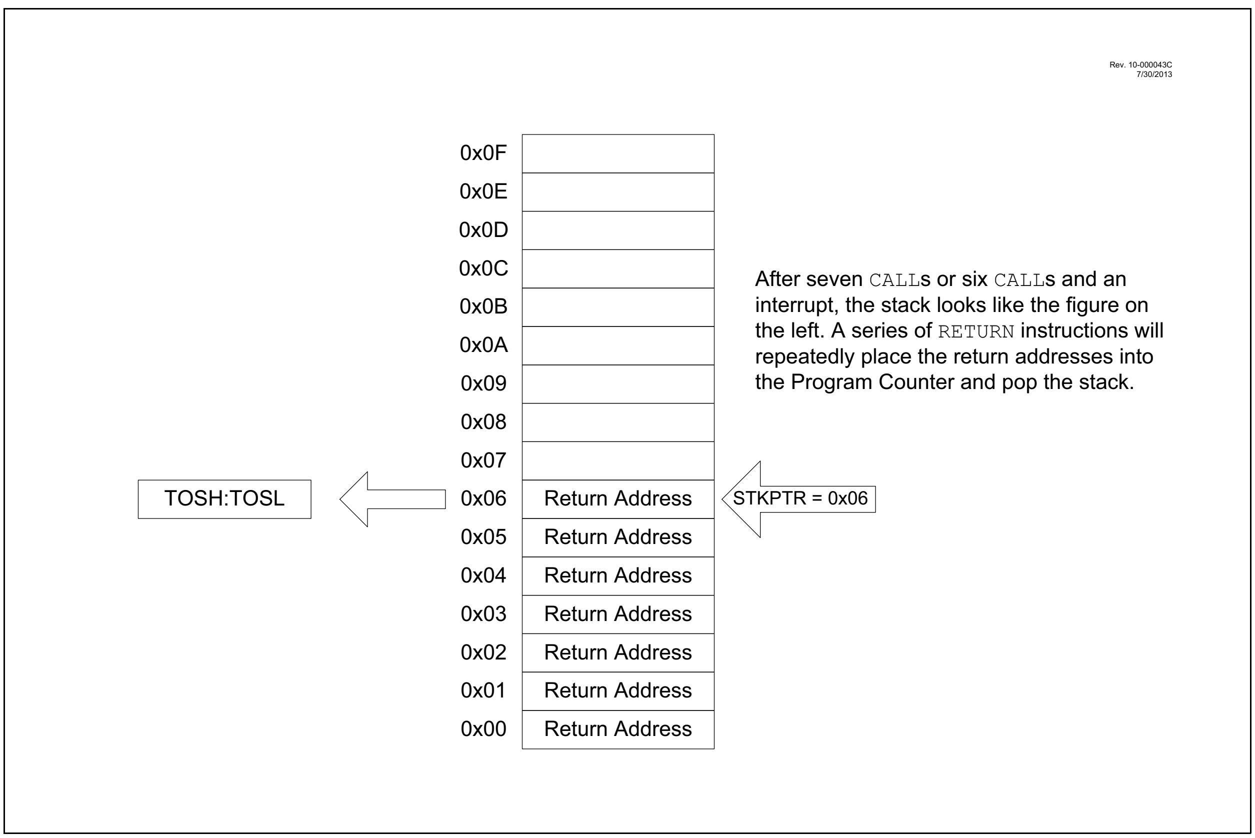
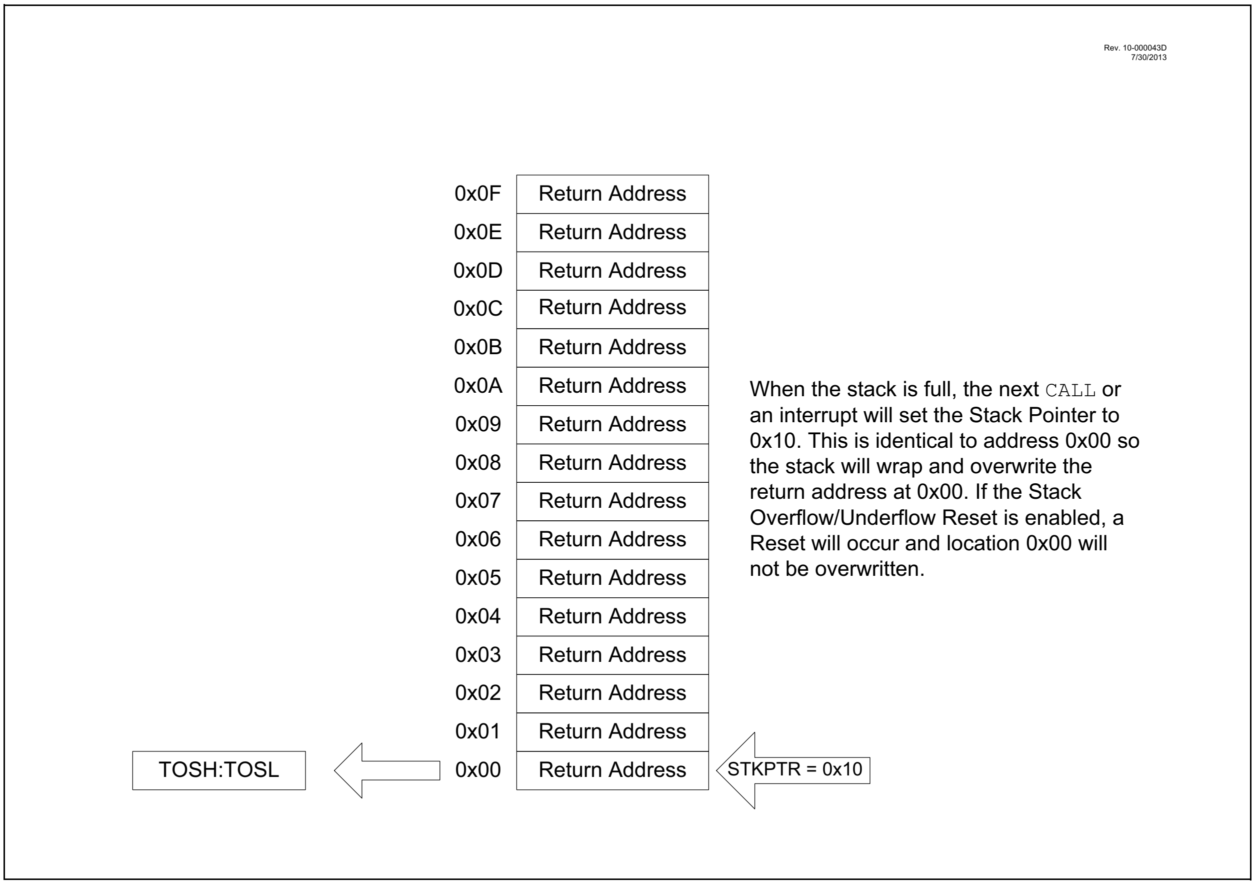


FIGURE 3-8: ACCESSING THE STACK EXAMPLE 4



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.

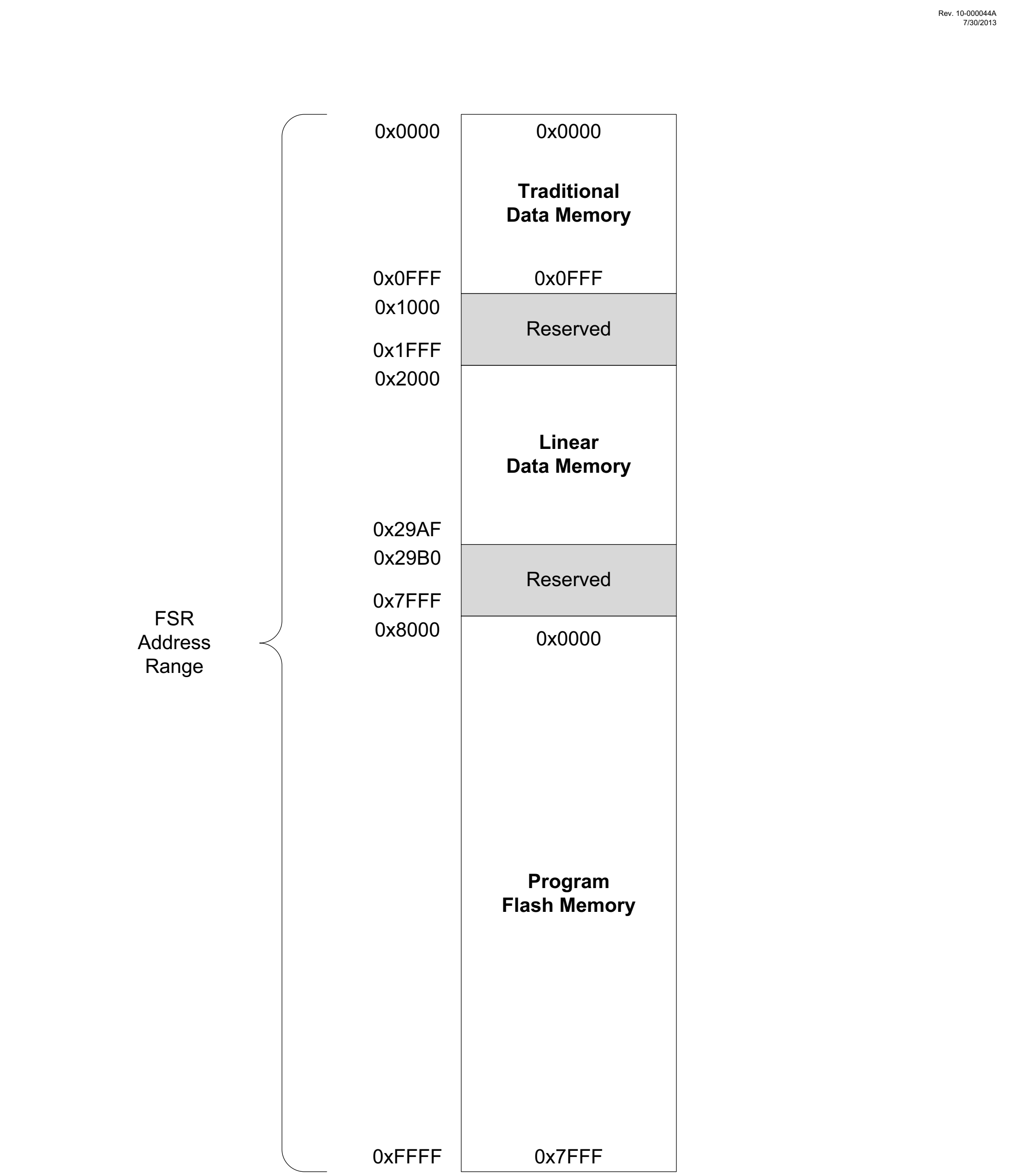
3.6 Indirect Addressing

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

FIGURE 3-9: INDIRECT ADDRESSING

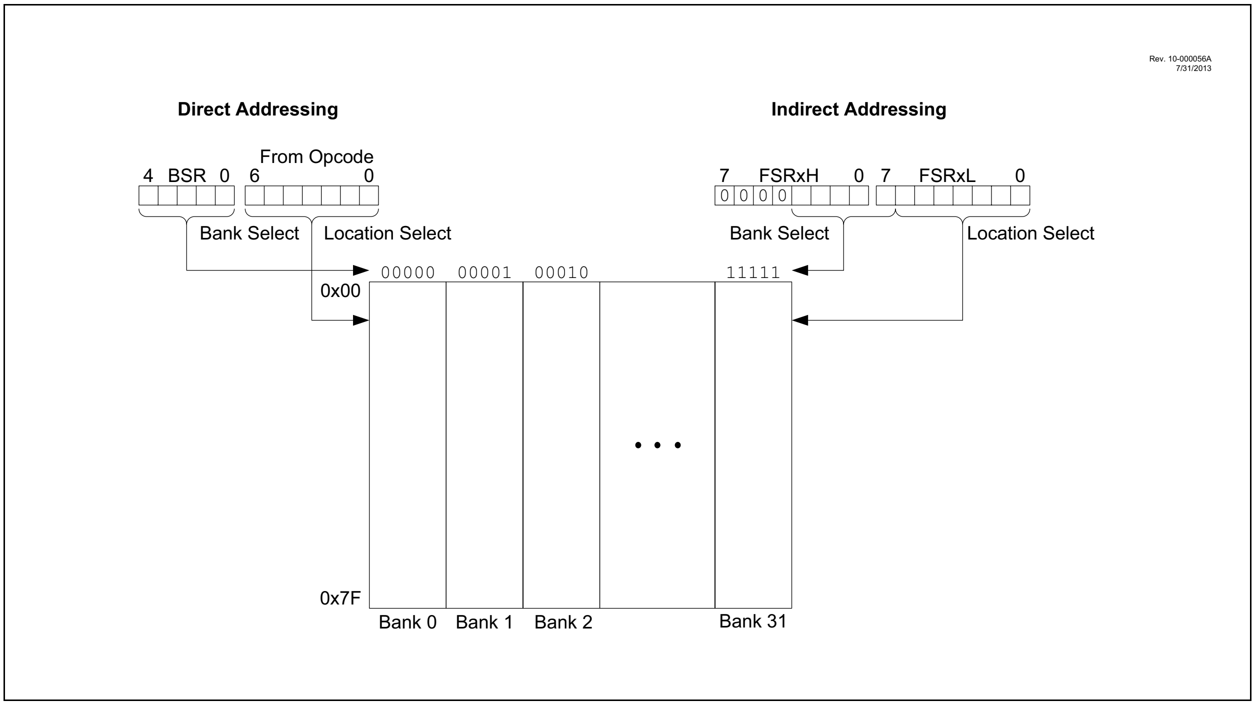


Note: Not all memory regions are completely implemented. Consult device memory tables for memory limits.

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 3-10: TRADITIONAL DATA MEMORY MAP



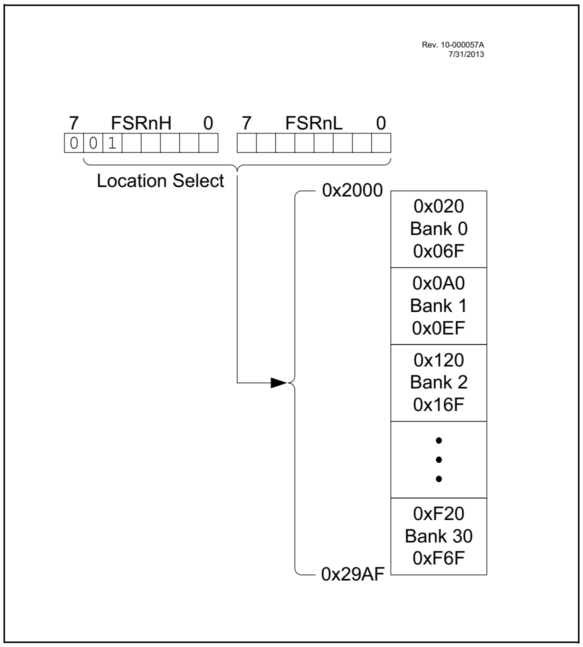
3.6.2 LINEAR DATA MEMORY

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.

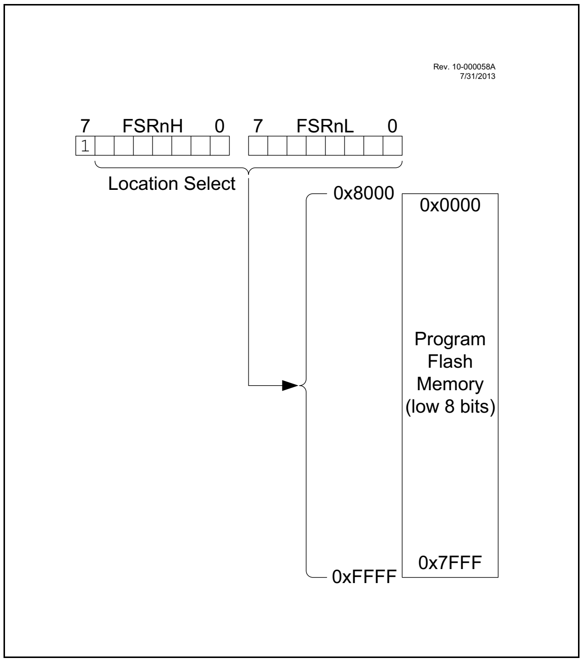
FIGURE 3-11: LINEAR DATA MEMORY MAP



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 3-12: 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 and Configuration Word 2 at 8008h.

Note:	The $\overline{\text{DEBUG}}$ 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

REGISTER 4-1: CONFIGURATION WORD 1

U-1	U-1	R/P-1	R/P-1	R/P-1	U-1
—	—	CLKOUTEN	BOREN<1:0>[1]		—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1
CP[2]	MCLRE	PWRTE[1]	WDTE<1:0>		—	FOSC<1:0>	
bit 7							bit 0

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

bit 13-12 **Unimplemented:** Read as '1'

bit 11 **CLKOUTEN:** Clock Out Enable bit
`1 = OFF` - CLKOUT function is disabled. I/O or oscillator function on CLKOUT pin
`0 = ON` - CLKOUT function is enabled on CLKOUT pin

bit 10-9 **BOREN<1:0>:** Brown-out Reset Enable bits[1]
`11 = ON` - Brown-out Reset enabled. The SBOREN bit is ignored.
`10 = SLEEP` - Brown-out Reset enabled while running and disabled in Sleep. The SBOREN bit is ignored.
`01 = SBODEN` - Brown-out Reset controlled by the SBOREN bit in the BORCON register
`00 = OFF` - Brown-out Reset disabled. The SBOREN bit is ignored.

bit 8 **Unimplemented:** Read as '1'

bit 7 **CP:** Flash Program Memory Code Protection bit[2]
`1 = OFF` - Code protection off. Program Memory can be read and written.
`0 = ON` - Code protection on. Program Memory cannot be read or written externally.

bit 6 **MCLRE:** MCLR/VPP Pin Function Select bit
If LVP bit = `1 (ON)`:
This bit is ignored. MCLR/VPP pin function is MCLR; Weak pull-up enabled.
If LVP bit = `0 (OFF)`:
`1 = ON` - MCLR/VPP pin function is MCLR; Weak pull-up enabled.
`0 = OFF` - MCLR/VPP pin function is digital input; MCLR internally disabled; Weak pull-up under control of pin's WPU control bit.

bit 5 **PWRTE:** Power-up Timer Enable bit[1]
`1 = OFF` - PWRT disabled
`0 = ON` - PWRT enabled

bit 4-3 **WDTE<1:0>:** Watchdog Timer Enable bit
`11 = ON` - WDT enabled. SWDTEN is ignored.
`10 = SLEEP` - WDT enabled while running and disabled in Sleep. SWDTEN is ignored.
`01 = SWDTEN` - WDT controlled by the SWDTEN bit in the WDTCON register
`00 = OFF` - WDT disabled. SWDTEN is ignored.

bit 2 **Unimplemented:** Read as '1'

bit 1-0 **FOSC<1:0>:** Oscillator Selection bits
`11 = ECH` - External Clock, High-Power mode: CLKI on CLKI
`10 = ECM` - External Clock, Medium-Power mode: CLKI on CLKI
`01 = ECL` - External Clock, Low-Power mode: CLKI on CLKI
`00 = INTOSC` - I/O function on CLKI

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.

REGISTER 4-2: CONFIGURATION WORD 2

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
LVP(1)	DEBUG(2)	LPBOREN	BORV(3)	STVREN	PLLEN
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1	R/P-1
—	—	—	—	—	PPS1WAY	WRT<1:0>	
bit 7							bit 0

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

bit 13 **LVP:** Low-Voltage Programming Enable bit(1)
`1 = ON` - Low-voltage programming enabled. MCLR/VPP pin function is MCLR. MCLRE Configuration bit is ignored.
`0 = OFF` - High Voltage on MCLR/VPP must be used for programming

bit 12 **DEBUG:** Debugger Mode bit(2)
`1 = OFF` - In-Circuit Debugger disabled; ICSPCLK and ICSPDAT are general purpose I/O pins.
`0 = ON` - In-Circuit Debugger enabled; ICSPCLK and ICSPDAT are dedicated to the debugger.

bit 11 **LPBOREN:** Low-Power Brown-out Reset Enable bit
`1 = OFF` - Low-power Brown-out Reset is disabled
`0 = ON` - Low-power Brown-out Reset is enabled

bit 10 **BORV:** Brown-out Reset Voltage Selection bit(3)
`1 = LOW` - Brown-out Reset voltage (VBOR), low trip point selected
`0 = HIGH` - Brown-out Reset voltage (VBOR), high trip point selected

bit 9 **STVREN:** Stack Overflow/Underflow Reset Enable bit
`1 = ON` - Stack Overflow or Underflow will cause a Reset
`0 = OFF` - Stack Overflow or Underflow will not cause a Reset

bit 8 **PLLEN:** PLL Enable bit
`1 = ON` - 4xPLL enabled
`0 = OFF` - 4xPLL disabled

bit 7-3 **Unimplemented:** Read as '1'

bit 2 **PPS1WAY**: PPSLOCK Bit One-Way Set Enable bit
`1 = ON` The PPSLOCK bit can only be set once after an unlocking sequence is executed; once PPSLOCK is set, all future changes to PPS registers are prevented
`0 = OFF` The PPSLOCK bit can be set and cleared as needed (provided an unlocking sequence is executed)

Note 1: This bit cannot be programmed to '0' when programming mode is entered via LVP.
2: 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'.
3: See VBOR parameter for specific trip point voltages.

REGISTER 4-2: CONFIGURATION WORD 2 (CONTINUED)

bit 1-0 **WRT<1:0>:** Flash Memory Self-Write Protection bits

4 kW Flash memory: (PIC16(L)F1574/8):

`11 = OFF`	Write protection off
`10 = BOOT`	0000h to 1FFh write protected, 0200h to 0FFFh may be modified by PMCON control
`01 = HALF`	0000h to 07FFh write protected, 0800h to 0FFFh may be modified by PMCON control
`00 = ALL`	0000h to 0FFFh write protected, no addresses may be modified by PMCON control

8 kW Flash memory: (PIC16(L)F1575/9)

`11 = OFF`	Write protection off
`10 = BOOT`	0000h to 1FFh write protected, 0200h to 1FFFh may be modified by PMCON control
`01 = HALF`	0000h to 0FFFh write protected, 1000h to 1FFFh may be modified by PMCON control
`00 = ALL`	0000h to 1FFFh write protected, no addresses may be modified by PMCON control

Note 1: This bit cannot be programmed to '0' when programming mode is entered via LVP.

2: The $\overline{\text{DEBUG}}$ 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'.

3: See V_{BOR} parameter for specific trip point voltages.

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 $\overline{\text{CP}}$ bit in Configuration Words. When $\overline{\text{CP}}$ = `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 bootloader 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 *"PIC16(L)F157x Memory Programming Specification"* (DS40001766).

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 4-3: DEVICEID: DEVICE ID REGISTER(1)

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		
‘0’ = Bit is cleared	‘1’ = Bit is set	x = Bit is unknown

bit 13-0 **DEV<13:0>:** Device ID bits
Refer to Table 4-1 to determine what these bits will read on which device. A value of 3FFFh is invalid.

Note 1: This location cannot be written.

REGISTER 4-4: REVISIONID: REVISION ID REGISTER(1)

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		
‘0’ = Bit is cleared	‘1’ = Bit is set	x = Bit is unknown

bit 13-0 **REV<13:0>:** Revision ID bits
These bits are used to identify the device revision.

Note 1: This location cannot be written.

TABLE 4-1: DEVICE ID VALUES

DEVICE	Device ID	Revision ID
PIC16F1574	3000h	2xxxh
PIC16F1575	3001h	2xxxh
PIC16F1578	3002h	2xxxh
PIC16F1579	3003h	2xxxh
PIC16LF1574	3004h	2xxxh
PIC16LF1575	3005h	2xxxh
PIC16LF1578	3006h	2xxxh
PIC16LF1579	3007h	2xxxh

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 logic level clocks. 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.

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

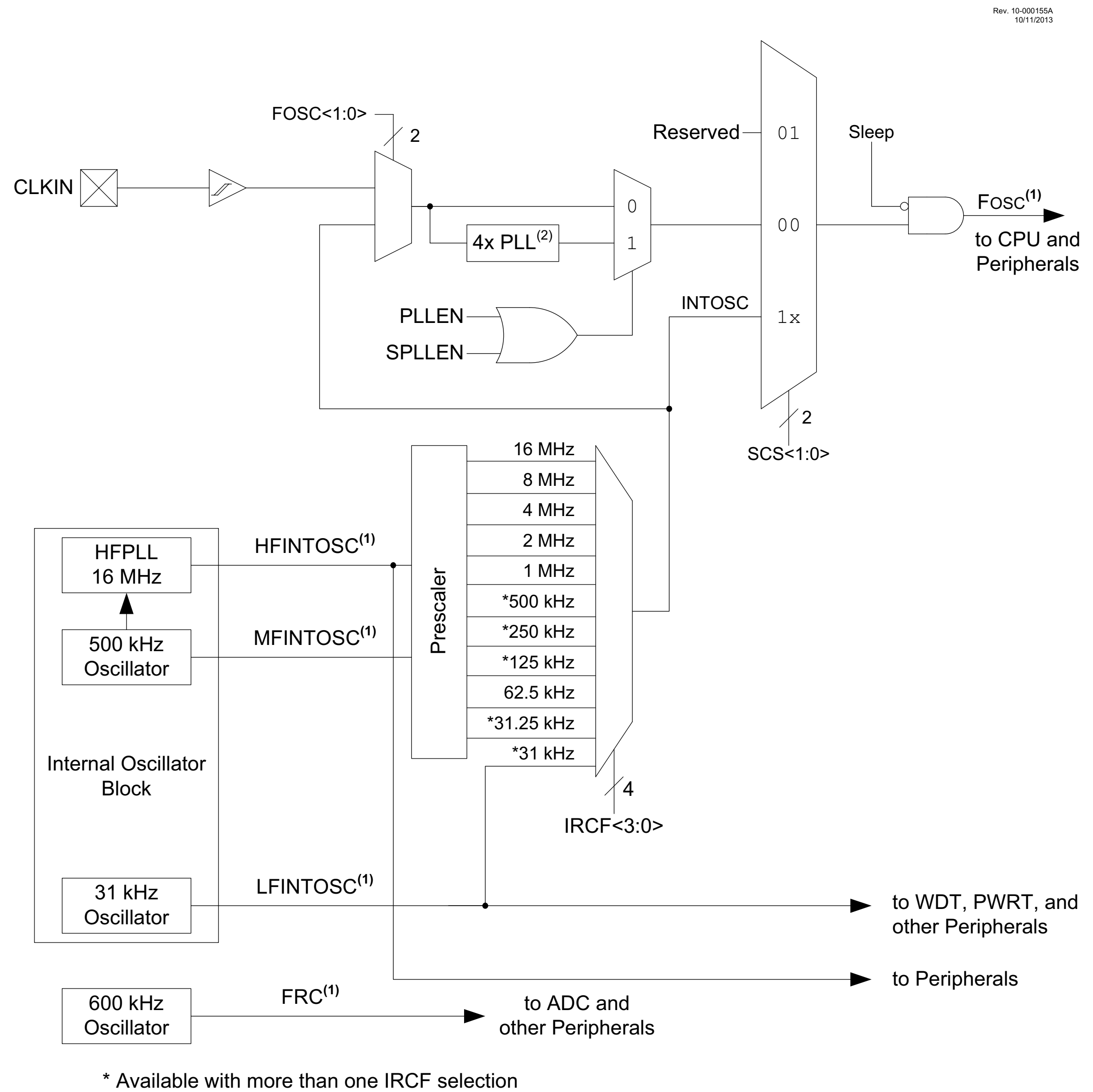
1. ECL – External Clock Low-Power mode (0 MHz to 0.5 MHz)
2. ECM – External Clock Medium-Power mode (0.5 MHz to 4 MHz)
3. ECH – External Clock High-Power mode (4 MHz to 32 MHz)
4. INTOSC – Internal oscillator (31 kHz to 32 MHz).

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 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.

FIGURE 5-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



Note 1: See **Section 5.2 "Clock Source Types"**.
2: ST Buffer is high-speed type when using T1CKI.
3: If FOSC<1:0> = 00, 4x PLL can only be used if IRCF<3:0> = 1110.

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.

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 **Section 5.3 "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:
  - Timer1 oscillator during run-time, or
  - An external clock source determined by the value of the FOSC bits.

See **Section 5.3 "Clock Switching"** for more information.

5.2.1.1 EC Mode

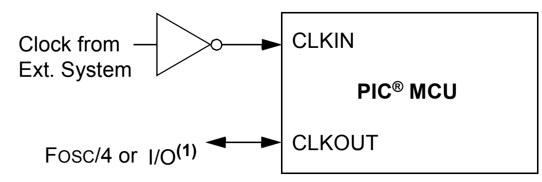
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. CLKOUT is available for general purpose I/O or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

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

FIGURE 5-2: EXTERNAL CLOCK (EC) MODE OPERATION



Note 1: Output depends upon $\overline{\text{CLKOUTEN}}$ bit of the Configuration Words.

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 **Section 5.3 "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 $\overline{\text{CLKOUTEN}}$ 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.

1. 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 user-adjusted via software using the OSCTUNE register (Register 5-3).
2. 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 **Section 5.2.2.8 "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 **Section 5.2.2.8 "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 **Section 5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT) and Watchdog Timer (WDT).

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

- Configure the IRCF<3:0> bits of the OSCCON register for the desired LF frequency, and
- FOSC<1:0> = `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)

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:

- 32 MHz (requires 4x PLL)
- 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. 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 the 8 MHz HFINTOSC set to use (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 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:

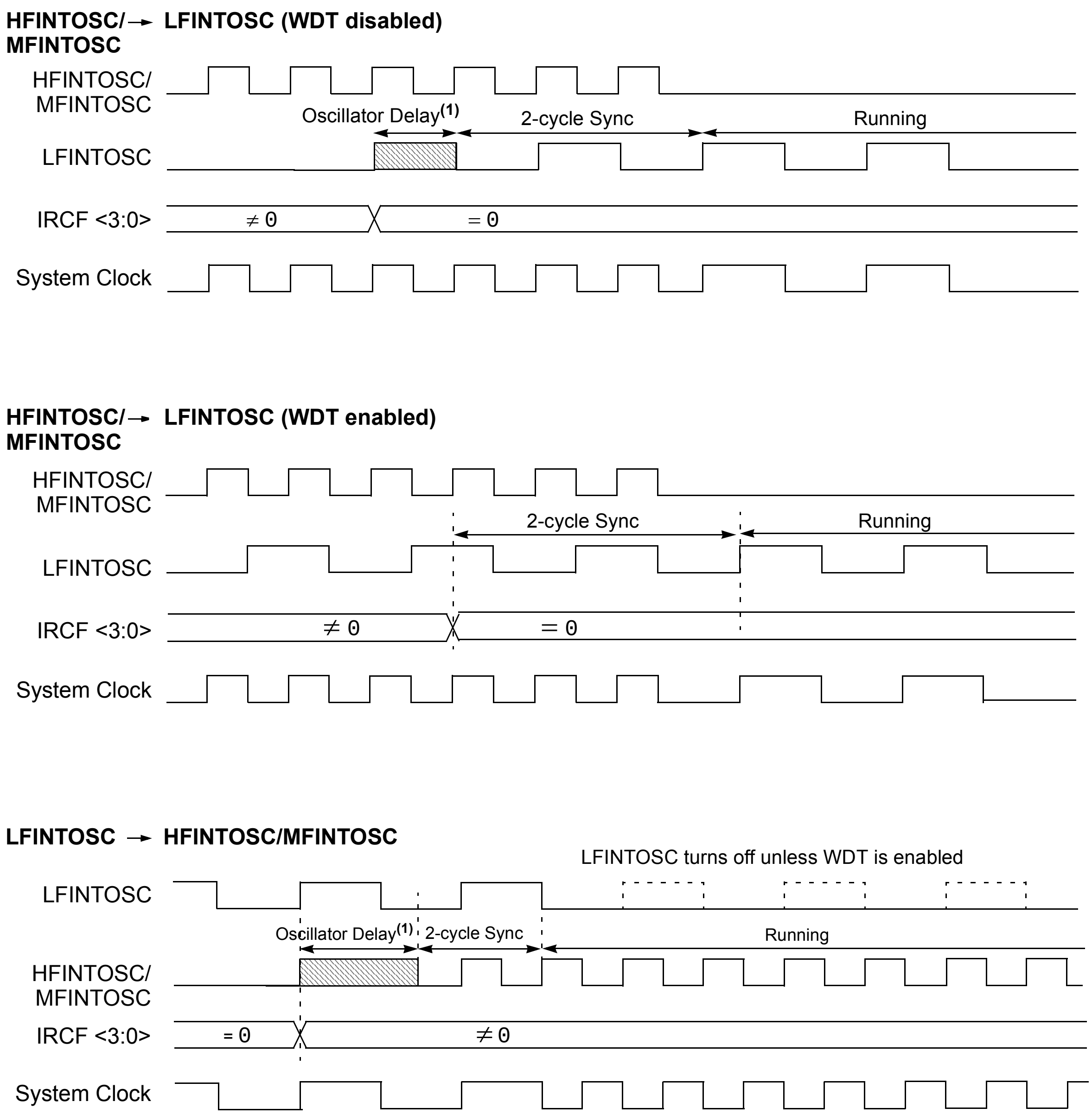
1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
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 **Section 27.0 “Electrical Specifications”**.

FIGURE 5-3: INTERNAL OSCILLATOR SWITCH TIMING



Note 1: See Table 5-1, “Oscillator Switching Delays” for more information.

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 = `01`, the system clock source is the Timer1 oscillator.
- When the SCS bits of the OSCCON register = `1x`, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

Note: Any automatic clock switch does not update the SCS bits of the OSCCON register. The user can monitor the OSTS bit of the OSCSTAT register to determine the current system clock source.

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.

TABLE 5-1: OSCILLATOR SWITCHING DELAYS

Switch From	Switch To	Frequency	Oscillator Delay
Sleep/POR	LFINTOSC(1) MFINTOSC(1) HFINTOSC(1)	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	Oscillator Warm-up Delay (T_{WARM})
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.)

Note 1: PLL inactive.

5.4 Register Definitions: Oscillator Control

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER

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 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 **SPLLEN:** Software PLL Enable bit
If PLLEN in Configuration Words = 1:
SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements)
If PLLEN in Configuration Words = 0:
1 = 4x PLL Is enabled
0 = 4x PLL is disabled

bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits
1111 = 16 MHz HF
1110 = 8 MHz or 32 MHz HF (see **Section 5.2.2.1 “HFINTOSC”**)
1101 = 4 MHz HF
1100 = 2 MHz HF
1011 = 1 MHz HF
1010 = 500 kHz HF(1)
1001 = 250 kHz HF(1)
1000 = 125 kHz HF(1)
0111 = 500 kHz MF (default upon Reset)
0110 = 250 kHz MF
0101 = 125 kHz MF
0100 = 62.5 kHz MF
0011 = 31.25 kHz HF(1)
0010 = 31.25 kHz MF
000x = 31 kHz LF

bit 2 **Unimplemented:** Read as ‘0’

bit 1-0 **SCS<1:0>:** System Clock Select bits
1x = Internal oscillator block
01 = Reserved
00 = Clock determined by FOSC<1:0> in Configuration Words.

Note 1: Duplicate frequency derived from HFINTOSC.

REGISTER 5-2: OSCSTAT: OSCILLATOR STATUS REGISTER

U-0	R-0/q	R-q/q	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 = 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 = Conditional

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **PLLR** 4x PLL Ready bit
1 = 4x PLL is ready
0 = 4x PLL is not ready

bit 5 **OSTS:** Oscillator Start-up Timer Status bit
1 = Running from the clock defined by the FOSC<1:0> bits of the Configuration Words
0 = Running from an internal oscillator (FOSC<1:0> = 00)

bit 4 **HFIOFR:** High-Frequency Internal Oscillator Ready bit
1 = HFINTOSC is ready
0 = HFINTOSC is not ready

bit 3 **HFIOFL:** High-Frequency Internal Oscillator Locked bit
1 = HFINTOSC is at least 2% accurate
0 = HFINTOSC is not 2% accurate

bit 2 **MFIOFR:** Medium-Frequency Internal Oscillator Ready bit
1 = MFINTOSC is ready
0 = MFINTOSC is not ready

bit 1 **LFIOFR:** Low-Frequency Internal Oscillator Ready bit
1 = LFINTOSC is ready
0 = LFINTOSC is not ready

bit 0 **HFIOFS:** High-Frequency Internal Oscillator Stable bit
1 = HFINTOSC is at least 0.5% accurate
0 = HFINTOSC is not 0.5% accurate

REGISTER 5-3: OSCTUNE: OSCILLATOR TUNING 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 = 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 **TUN<5:0>:** Frequency Tuning bits
100000 = Minimum frequency
•
•
•
111111 =
000000 = Oscillator module is running at the factory-calibrated frequency.
000001 =
•
•
•
011110 =
011111 = Maximum frequency

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		70
OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	71
OSCTUNE	—	—	TUN<5:0>						72

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

TABLE 5-3: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		

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

6.0 RESETS

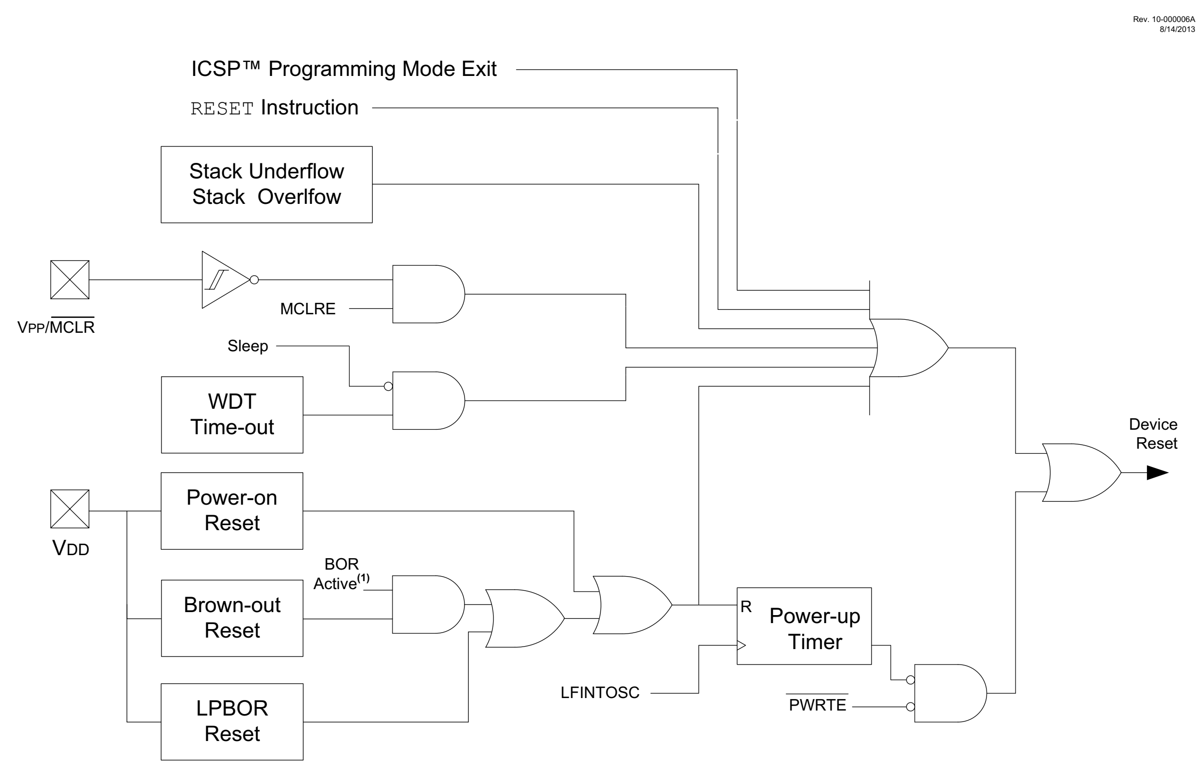
There are multiple ways to reset this device:

- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- Low-Power Brown-Out Reset (LPBOR)
- $\overline{\text{MCLR}}$ 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.

FIGURE 6-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: See Table 6-1 for BOR active conditions.

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 $\overline{\text{MCLR}}$ 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 time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is enabled by clearing the $\overline{\text{PWRTE}}$ 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.

TABLE 6-1: BOR OPERATING MODES

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

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.

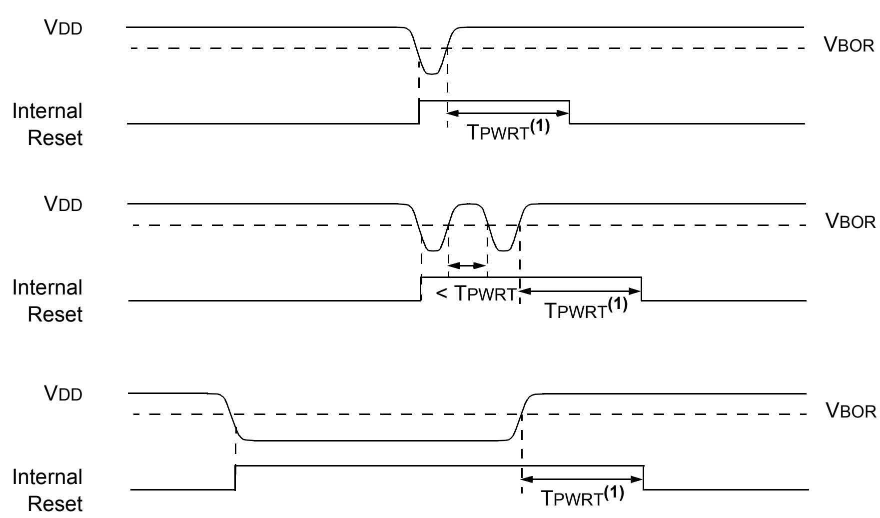
6.2.3 BOR CONTROLLED BY SOFTWARE

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.

FIGURE 6-2: BROWN-OUT SITUATIONS



Note 1: TPWRT delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

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-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS	—	—	—	—	—	BORRDY
bit 7							bit 0

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

bit 7 **SBOREN:** Software Brown-Out Reset Enable bit
If BOREN <1:0> in Configuration Words = 01:
1 = BOR Enabled
0 = BOR Disabled
If BOREN <1:0> in Configuration Words ≠ 01:
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: 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 MCLR is an optional external input that can reset the device. The 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 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 **Section 11.1 "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. The TO and PD bits in the STATUS register are changed to indicate the WDT Reset. See **Section 9.0 "Watchdog Timer (WDT)"** for more information.

6.7 RESET Instruction

A `RESET` instruction will cause a device Reset. The 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 **Section 3.5.2 "Overflow/Underflow 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 PWRTE bit of Configuration Words.

6.11 Start-up Sequence

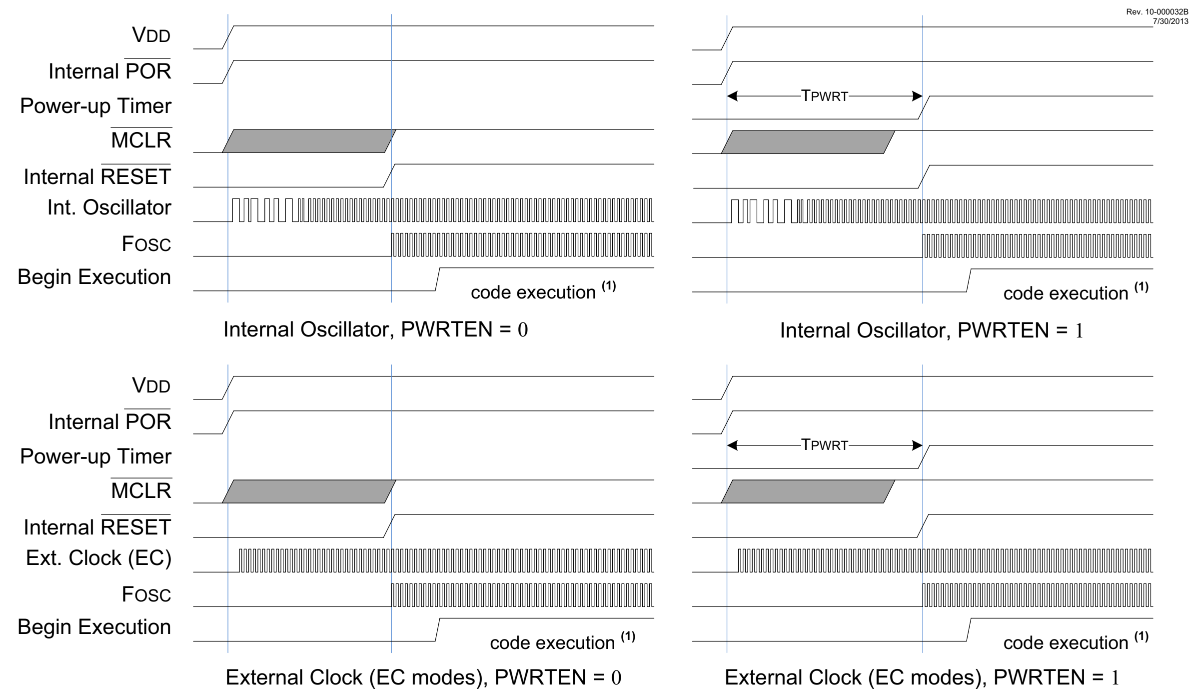
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

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 **Section 5.0 "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.

FIGURE 6-3: RESET START-UP SEQUENCE



Note 1: Code execution begins 10 FOSC cycles after the FOSC clock is released.

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.

TABLE 6-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

STKOVF	STKUNF	$\overline{RWDT}$	$\overline{RMCLR}$	$\overline{RI}$	$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	Condition
0	0	1	1	1	0	x	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	0	1	1	1	0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
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	$\overline{MCLR}$ Reset during normal operation
u	u	u	0	u	u	u	1	0	$\overline{MCLR}$ 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 6-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	---1 1000	00-- 110x
$\overline{MCLR}$ Reset during normal operation	0000h	---u uuuu	uu-- 0uuu
$\overline{MCLR}$ Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-- uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-- uuuu
Brown-out Reset	0000h	---1 1uuu	00-- 11u0
Interrupt Wake-up from Sleep	PC + 1[(1)]	---1 0uuu	uu-- uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-- u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-- 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 ($\overline{\text{POR}}$)
- Brown-Out Reset ($\overline{\text{BOR}}$)
- Reset Instruction Reset ($\overline{\text{RI}}$)
- $\overline{\text{MCLR}}$ Reset ($\overline{\text{RMCLR}}$)
- Watchdog Timer Reset ($\overline{\text{RWDT}}$)
- 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	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is 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 **Unimplemented:** Read as ‘0’

bit 4 **$\overline{\text{RWDT}}$:** 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 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit
1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set by firmware
0 = A $\overline{\text{MCLR}}$ Reset has occurred (cleared by hardware)

bit 2 **$\overline{\text{RI}}$:** 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 **$\overline{\text{POR}}$:** 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 **$\overline{\text{BOR}}$:** 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: SUMMARY 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	76
PCON	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	80
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	23
WDTCON	—	—	WDTPS<4:0>					SWDTEN	100

Legend: — = unimplemented bit, reads as '0'. Shaded cells are not used by Resets.
Note 1: Other (non Power-up) Resets include $\overline{\text{MCLR}}$ 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	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOREN}}$	BORV	STVREN	PLLEN	57
	7:0	—	—	—	—	—	PPS1WAY	WRT<1:0>		

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.

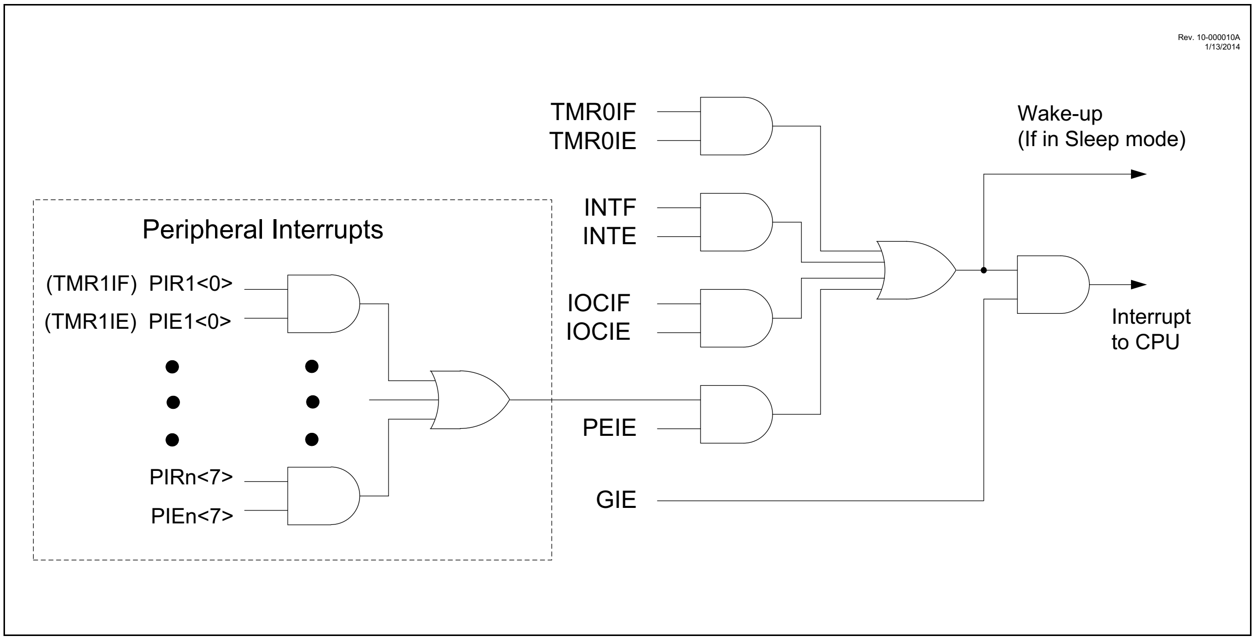
This chapter contains the following information for Interrupts:

- 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 **"Section 7.5 "Automatic Context Saving"."**)
- PC is loaded with the interrupt vector 0004h

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

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

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

Note 1:	Individual interrupt flag bits 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.

FIGURE 7-2: INTERRUPT LATENCY

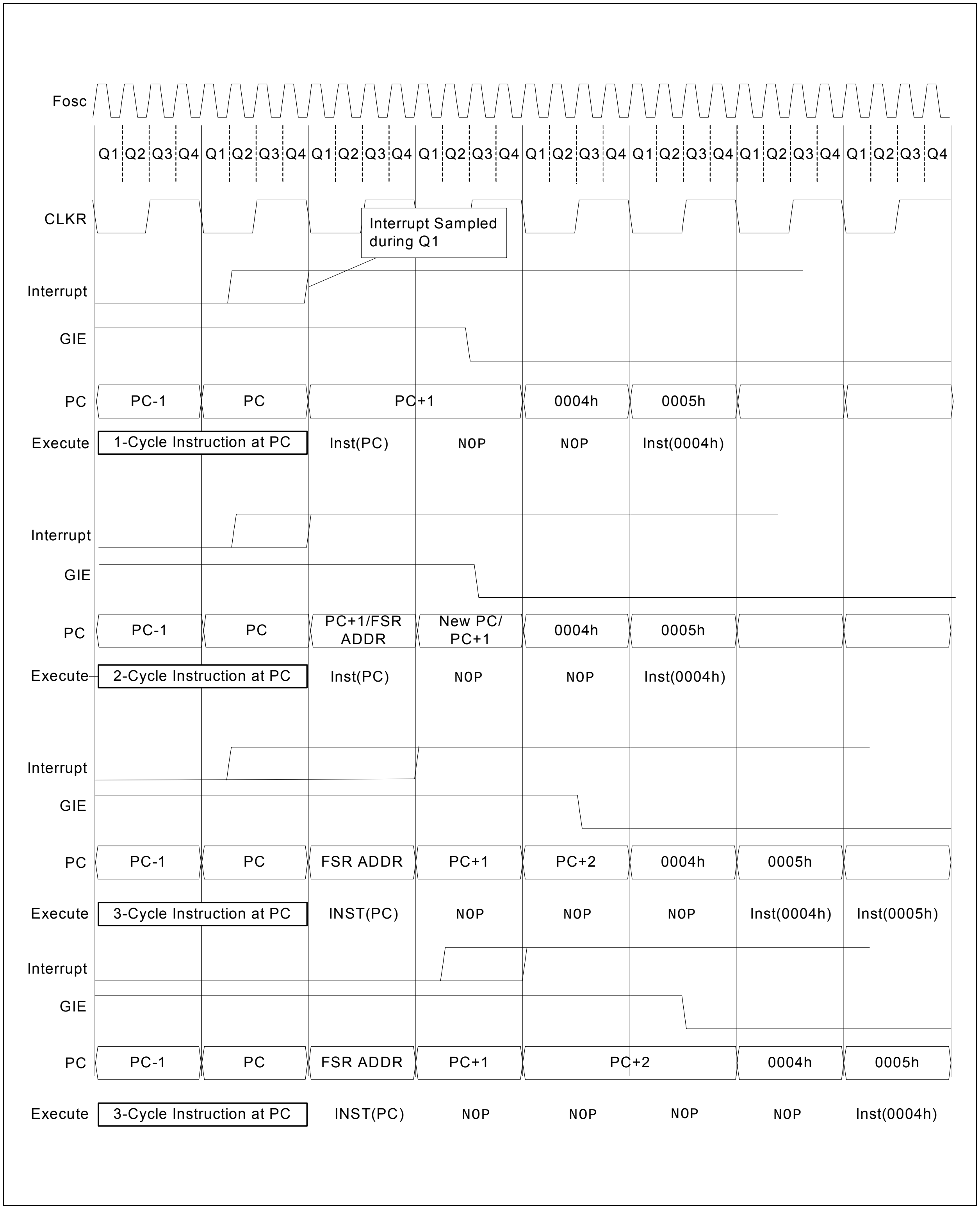
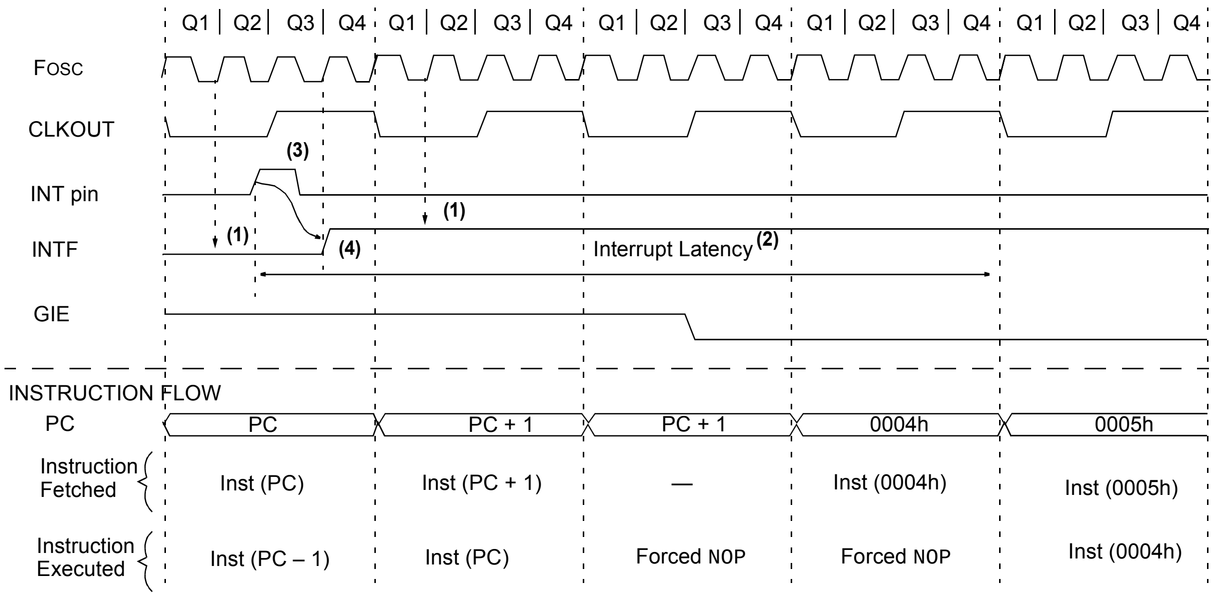


FIGURE 7-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: For minimum width of INT pulse, refer to AC specifications in **Section 27.0 "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 **Section 8.0 "Power-Down Mode (Sleep)"** for more details.

7.4 INT Pin

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

7.5 Automatic Context Saving

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

- W register
- STATUS register (except for $\overline{\text{TO}}$ and $\overline{\text{PD}}$)
- 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

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-0/0
GIE(1)	PEIE(2)	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF(3)
bit 7							bit 0

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

bit 7 **GIE:** Global Interrupt Enable bit(1)
1 = Enables all active interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit(2)
1 = Enables all active peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **TMR0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** INT External Interrupt Enable bit
1 = Enables the INT external interrupt
0 = Disables the INT external interrupt

bit 3 **IOCIE:** Interrupt-on-Change Enable bit
1 = Enables the interrupt-on-change
0 = Disables the interrupt-on-change

bit 2 **TMR0IF:** Timer0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed
0 = TMR0 register did not overflow

bit 1 **INTF:** INT External Interrupt Flag bit
1 = The INT external interrupt occurred
0 = The INT external interrupt did not occur

bit 0 **IOCIF:** Interrupt-on-Change Interrupt Flag bit(3)
1 = When at least one of the interrupt-on-change pins changed state
0 = None of the interrupt-on-change pins have changed state

Note 1: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

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

3: The IOCIF Flag bit is read-only and cleared when all the interrupt-on-change flags in the IOCxF registers have been cleared by software.

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

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE
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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3-2 **Unimplemented:** Read as '0'

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

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

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

U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
—	C2IE	C1IE	—	—	—	—	—
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **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
1 = Enables the Comparator C1 interrupt
0 = Disables the Comparator C1 interrupt

bit 4-0 **Unimplemented:** Read as ‘0’

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

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	U-0	U-0	U-0	U-0
PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—
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 **PWM4IE:** PWM4 Interrupt Enable bit
1 = Enables the PWM4 interrupt
0 = Disables the PWM4 interrupt

bit 6 **PWM3IE:** PWM3 Interrupt Enable bit
1 = Enables the PWM3 interrupt
0 = Disables the PWM3 interrupt

bit 5 **PWM2IE:** PWM2 Interrupt Enable bit
1 = Enables the PWM2 interrupt
0 = Disables the PWM2 interrupt

bit 4 **PWM1IE:** PWM1 Interrupt Enable bit
1 = Enables the PWM1 interrupt
0 = Disables the PWM1 interrupt

bit 3-0 **Unimplemented:** Read as ‘0’

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

REGISTER 7-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	U-0	U-0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF
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 **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 6 **ADIF:** ADC Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5 **RCIF:** USART Receive Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 4 **TXIF:** USART Transmit Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

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

REGISTER 7-6: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
—	C2IF	C1IF	—	—	—	—	—
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **C2IF:** Comparator C2 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5 **C1IF:** Comparator C1 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 4-0 **Unimplemented:** Read as ‘0’

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

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

R-0/0	R-0/0	R-0/0	R-0/0	U-0	U-0	U-0	U-0
PWM4IF(1)	PWM3IF(1)	PWM2IF(1)	PWM1IF(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 **PWM4IF:** PWM4 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

bit 6 **PWM3IF:** PWM3 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5 **PWM2IF:** PWM2 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

bit 4 **PWM1IF:** PWM1 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

bit 3-0 **Unimplemented:** Read as ‘0’

Note 1: These bits are read-only. They must be cleared by addressing the Flag registers inside the module.

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

TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

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	87
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			180
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIE2	—	C2IE	C1IE	—	—	—	—	—	89
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	90
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
PIR2	—	C2IF	C1IF	—	—	—	—	—	92
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	93

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. $\overline{\text{PD}}$ bit of the STATUS register is cleared.
3. $\overline{\text{TO}}$ bit of the STATUS register is set.
4. CPU clock is disabled.
5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
   - LFINTOSC
   - T1CKI
7. ADC is unaffected, if the dedicated FRC oscillator is selected.
8. I/O ports maintain the status they had before `SLEEP` was executed (driving high, low or high-impedance).
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 module using HFINTOSC

I/O pins that are high-impedance inputs should be pulled to V_{DD} or V_{SS} externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include the FVR module. See **Section 14.0 "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 $\overline{\text{MCLR}}$ 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.

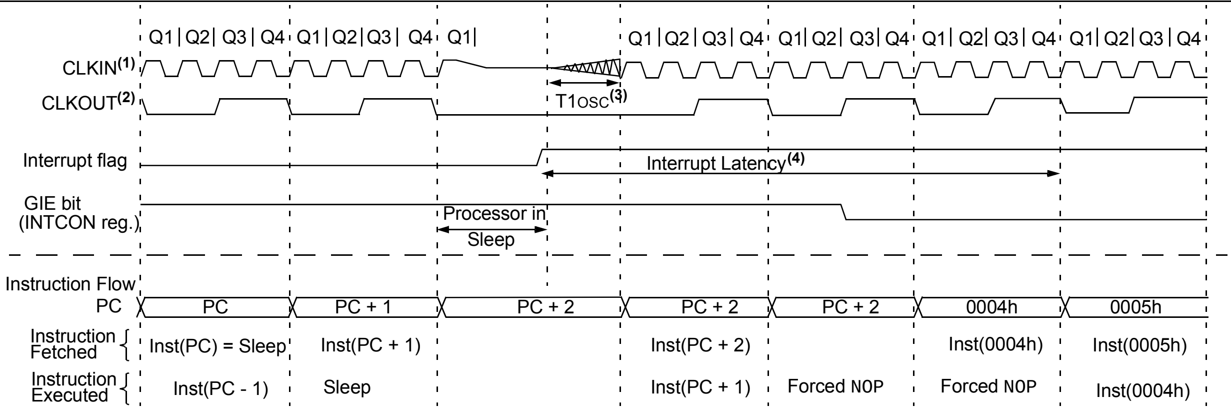
8.1.1 WAKE-UP USING INTERRUPTS

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
  - $\overline{\text{TO}}$ bit of the STATUS register will not be set
  - $\overline{\text{PD}}$ 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
  - $\overline{\text{TO}}$ bit of the STATUS register will be set
  - $\overline{\text{PD}}$ 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 $\overline{\text{PD}}$ bit. If the $\overline{\text{PD}}$ bit is set, the `SLEEP` instruction was executed as a `NOP`.

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



Note 1: External clock. High, Medium, Low mode assumed.
2: CLKOUT is shown here for timing reference.
3: T1OSC is not available on all devices.
4: GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

8.2 Low-Power Sleep Mode

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

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

8.2.1 SLEEP CURRENT VS. WAKE-UP TIME

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

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

8.2.2 PERIPHERAL USAGE IN SLEEP

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

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

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

Please refer to section **24.10 "Operation During Sleep"** for more information.

Note:	The PIC16LF1574/5/8/9 devices do not have a configurable Low-Power Sleep mode. PIC16LF1574/5/8/9 are unregulated devices and are always in the lowest power state when in Sleep, with no wake-up time penalty. These devices have a lower maximum VDD and I/O voltage than the PIC16F1574/5/8/9 devices. See **Section 27.0 "Electrical Specifications"** for more information.

8.3 Register Definitions: Voltage Regulator Control

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

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

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **VREGPM:** Voltage Regulator Power Mode Selection bit
1 = Low-Power Sleep mode enabled in Sleep(2)
Draws lowest current in Sleep, slower wake-up
0 = Normal-Power mode enabled in Sleep(1)
Draws higher current in Sleep, faster wake-up

bit 0 **Reserved:** Read as '1'. Maintain this bit set.

Note 1: PIC16F1574/5/8/9 only.
2: See **Section 27.0 "Electrical Specifications"**.

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

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	145
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	145
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	145
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIE2	—	C2IE	C1IE	—	—	—	—	—	89
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	90
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
PIR2	—	C2IF	C1IF	—	—	—	—	—	92
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	93
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	23
WDTCON	—	—	WDTPS<4:0>					SWDTEN	100

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

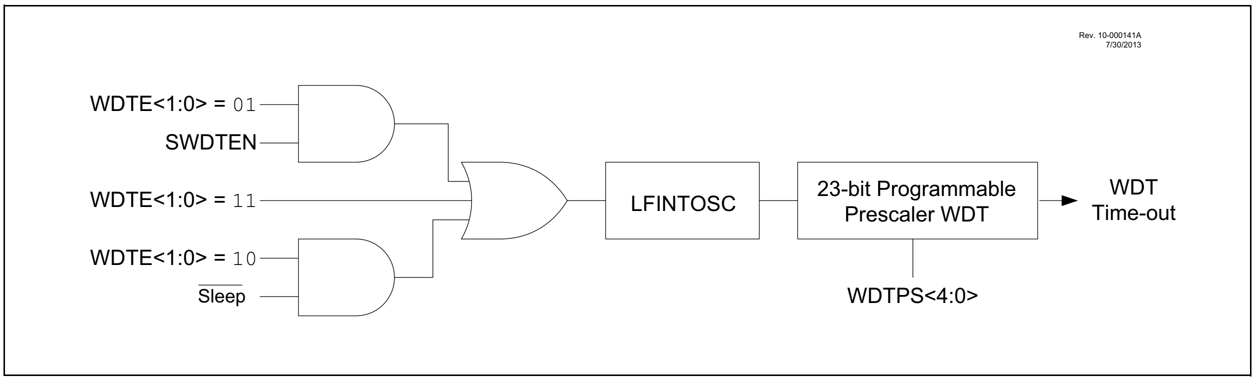
9.0 WATCHDOG TIMER (WDT)

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

The WDT has the following features:

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

FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM



9.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1 ms. See **Section 27.0 "Electrical Specifications"** for the LFINTOSC tolerances.

9.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration 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 SWDTEN bit of the WDTCON register.

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

TABLE 9-1: WDT OPERATING MODES

WDTE<1:0>	SWDTEN	Device Mode	WDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	X	X	Disabled

9.3 Time-Out Period

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

9.4 Clearing the WDT

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

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

See Table 9-2 for more information.

9.5 Operation During Sleep

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

The WDT remains clear until the OST, if enabled, completes. See **Section 5.0 "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 $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the event. The $\overline{RWDT}$ bit in the PCON register can also be used. See **Section 3.0 "Memory Organization"** for more information.

TABLE 9-2: WDT CLEARING CONDITIONS

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

9.6 Register Definitions: Watchdog Control

REGISTER 9-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>					SWDTEN
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 5-1 **WDTPS<4:0>:** Watchdog Timer Period Select bits(1)

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)
•
•
•
10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 (2^{23}) (Interval 256s nominal)
10001 = 1:4194304 (2^{22}) (Interval 128s nominal)
10000 = 1:2097152 (2^{21}) (Interval 64s nominal)
01111 = 1:1048576 (2^{20}) (Interval 32s nominal)
01110 = 1:524288 (2^{19}) (Interval 16s nominal)
01101 = 1:262144 (2^{18}) (Interval 8s nominal)
01100 = 1:131072 (2^{17}) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

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

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

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		70
PCON	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	80
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	23
WDTCON	—	—	WDTPS<4:0>					SWDTEN	100

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

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

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		

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 V_{DD} 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.

The Flash program memory can be protected in two ways; by code protection ($\overline{\text{CP}}$ bit in Configuration Words) and write protection (WRT<1:0> bits in Configuration Words).

Code protection ($\overline{\text{CP}}$ = 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 $\overline{\text{CP}}$ 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. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations.

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)F1574	32	32
PIC16(L)F1575		
PIC16(L)F1578		
PIC16(L)F1579		

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 `NOP`s. 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

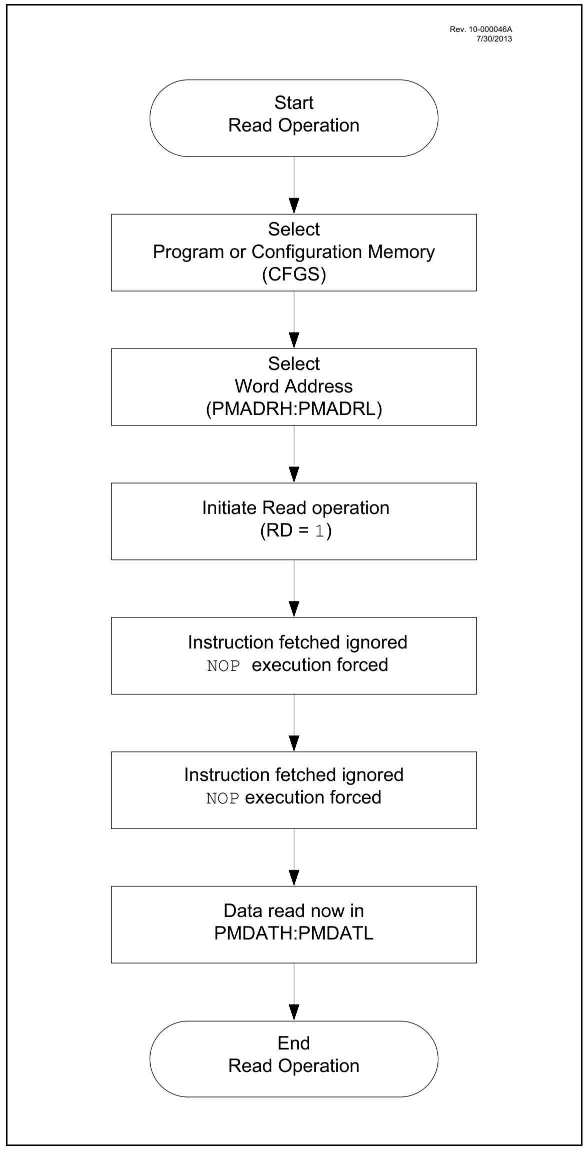
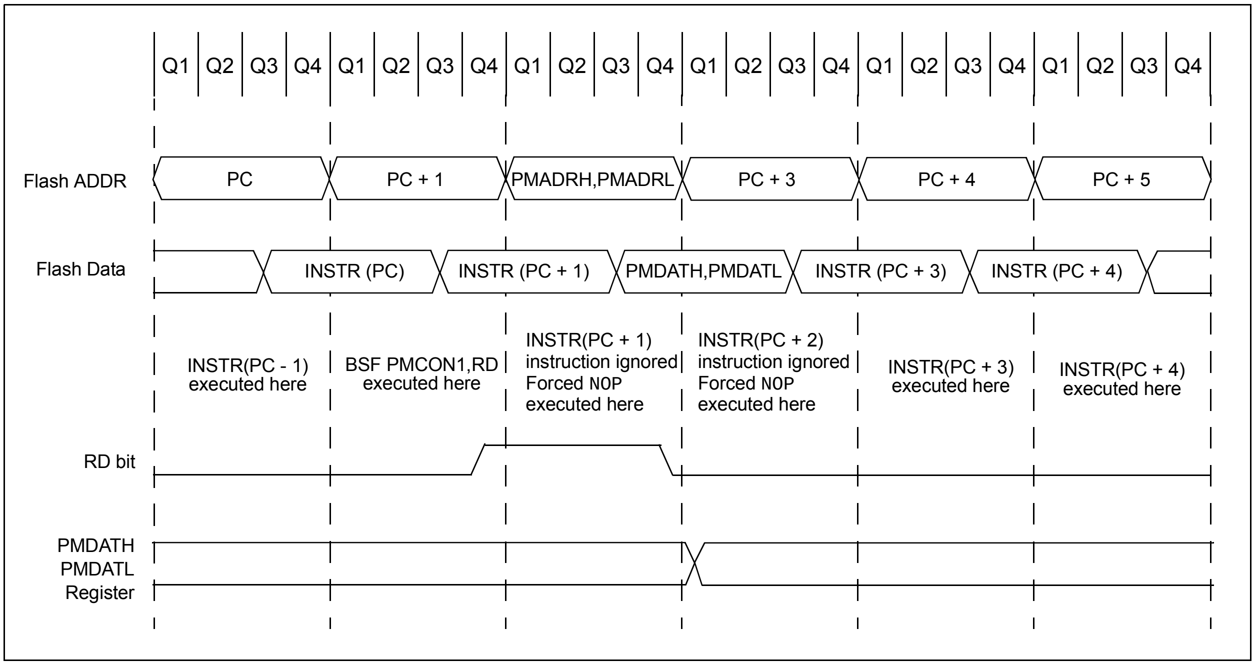


FIGURE 10-2: FLASH PROGRAM MEMORY READ CYCLE EXECUTION



EXAMPLE 10-1: FLASH PROGRAM MEMORY READ

```
* This code block will read 1 word of program
* memory at the memory address:
   PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

   BANKSEL  PMADRL          ; Select Bank for PMCON registers
   MOVLW    PROG_ADDR_LO    ;
   MOVWF    PMADRL          ; Store LSB of address
   MOVLW    PROG_ADDR_HI    ;
   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
   MOVF     PMDATH,W        ; Get MSB of word
   MOVWF    PROG_DATA_HI    ; Store in user location
```


10.2.2 FLASH MEMORY UNLOCK SEQUENCE

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

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

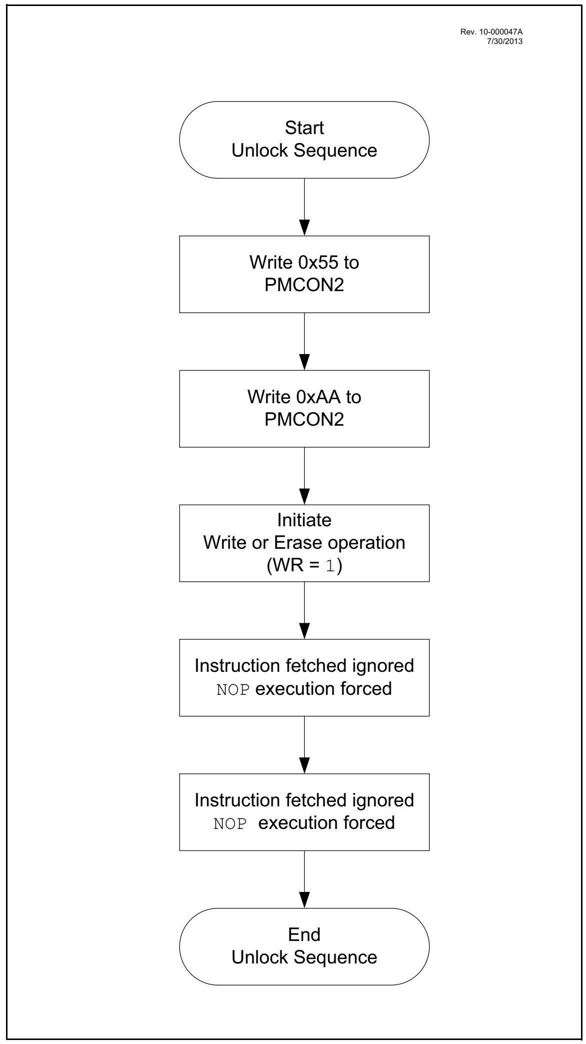
The unlock sequence consists of the following steps:

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

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

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

FIGURE 10-3: FLASH PROGRAM MEMORY UNLOCK SEQUENCE FLOWCHART



10.2.3 ERASING FLASH PROGRAM MEMORY

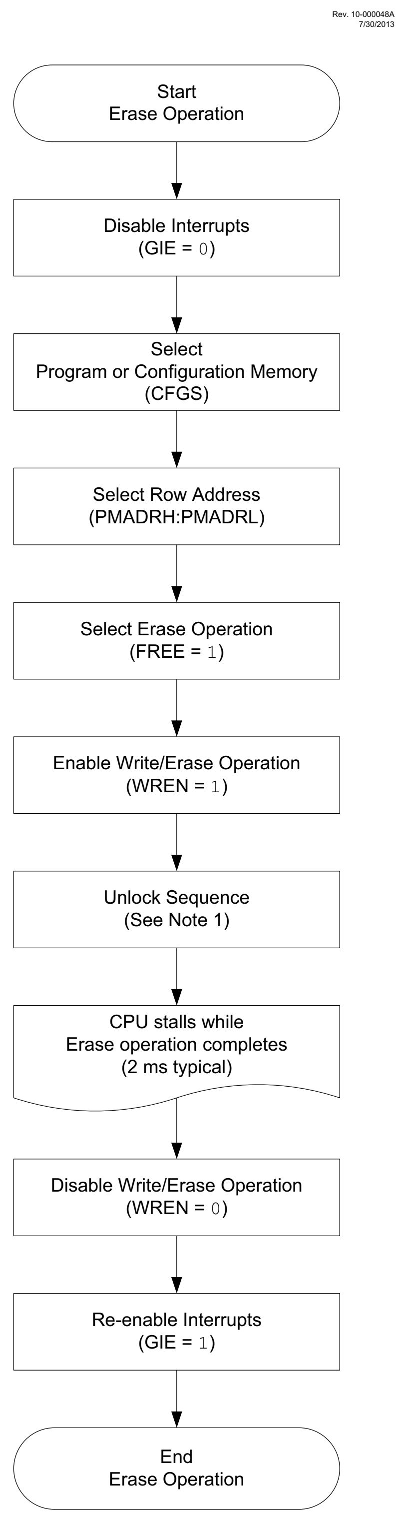
While executing code, program memory can only be erased by rows. To erase a row:

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

See Example 10-2.

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

FIGURE 10-4: FLASH PROGRAM MEMORY ERASE FLOWCHART



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         INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRL
        MOVF        ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF       PMADRL
        MOVF        ADDRH,W         ; Load upper 6 bits of erase address boundary
        MOVWF       PMADRH
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,FREE     ; Specify an erase operation
        BSF         PMCON1,WREN     ; Enable writes

Required Sequence:
        MOVLW       55h             ; Start of required sequence to initiate erase
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin erase
        NOP                         ; NOP instructions are forced as processor starts
        NOP                         ; row erase of program memory.
                                    ;
                                    ; The processor stalls until the erase process is complete
                                    ; after erase processor continues with 3rd instruction

        BCF         PMCON1,WREN     ; Disable writes
        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 16 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 4-bits of PMADRL, (PMADRL<3:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the PMDATH:PMDATL using the unlock sequence with LWLO = `1`. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

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

1. Set the WREN bit of the PMCON1 register.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '`1`', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the PMADRH:PMADRL register pair with the address of the location to be written.
5. Load the PMDATH:PMDATL register pair with the program memory data to be written.
6. 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.
9. Clear the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '`0`', the write sequence will initiate the write to Flash program memory.
10. Load the PMDATH:PMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 10.2.2 "Flash Memory Unlock Sequence"**). The entire program memory latch content is now written to Flash program memory.

Note:	The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

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

FIGURE 10-5: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 32 WRITE LATCHES

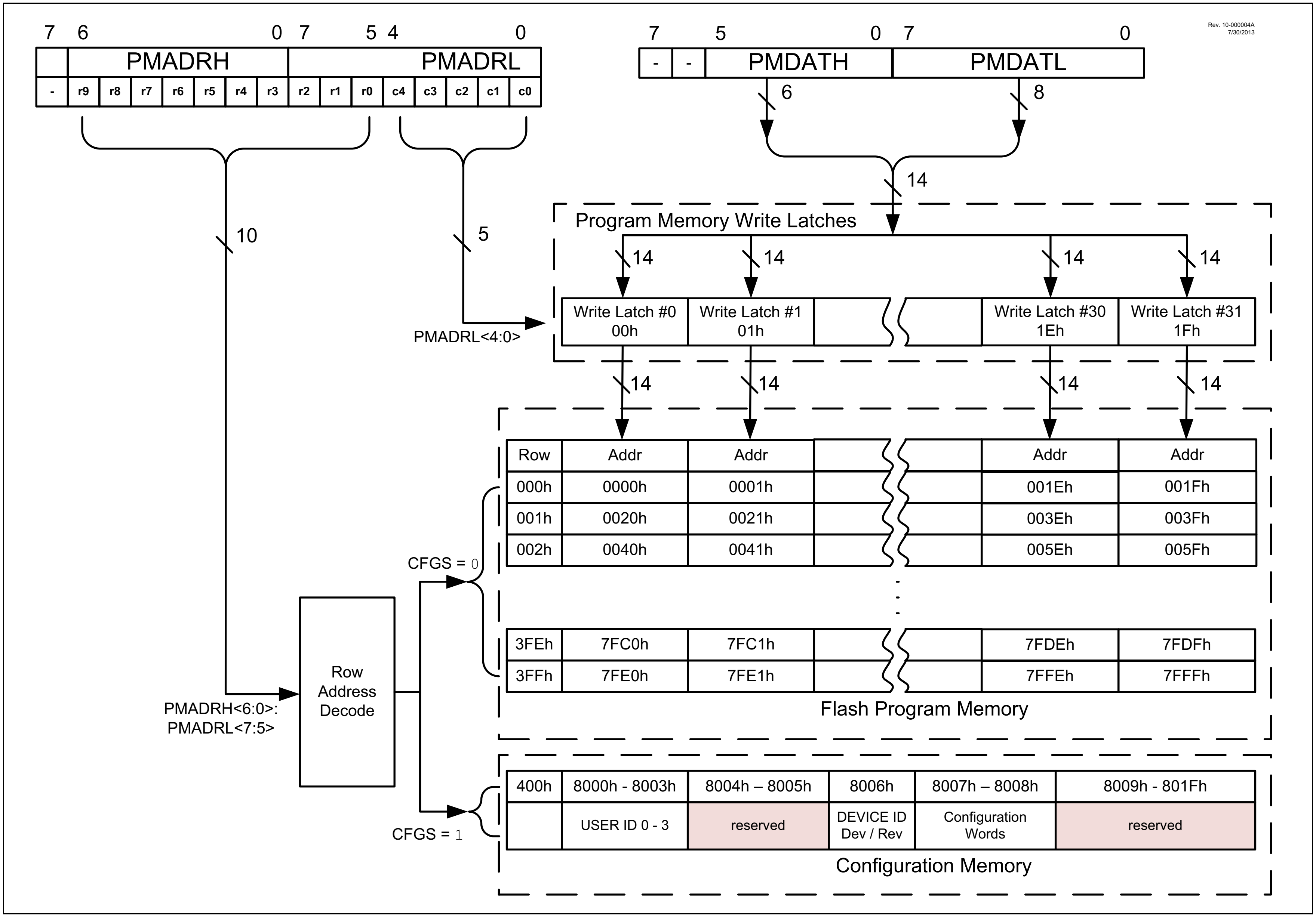
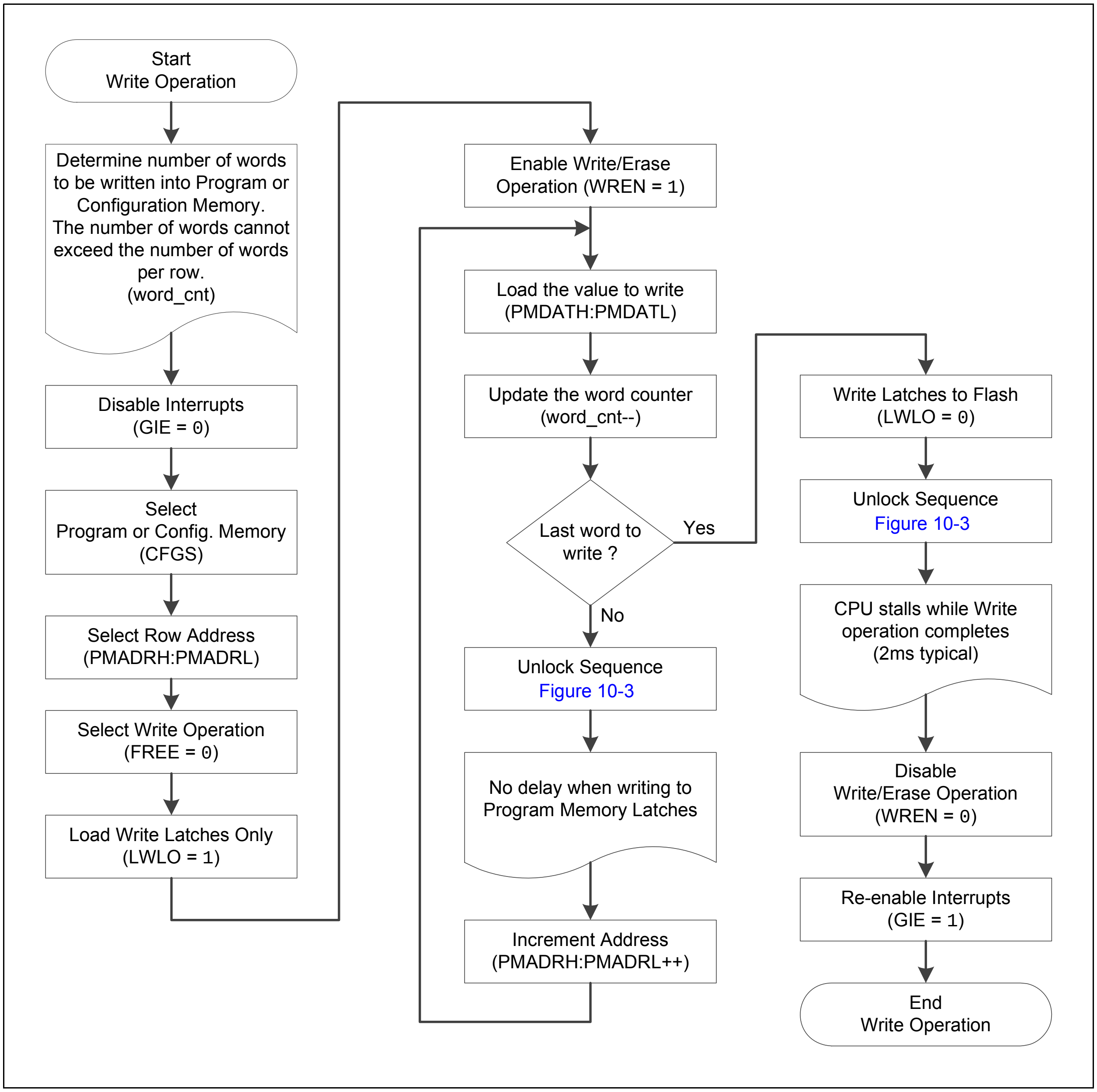


FIGURE 10-6: FLASH PROGRAM MEMORY WRITE FLOWCHART



EXAMPLE 10-3: WRITING TO FLASH PROGRAM MEMORY

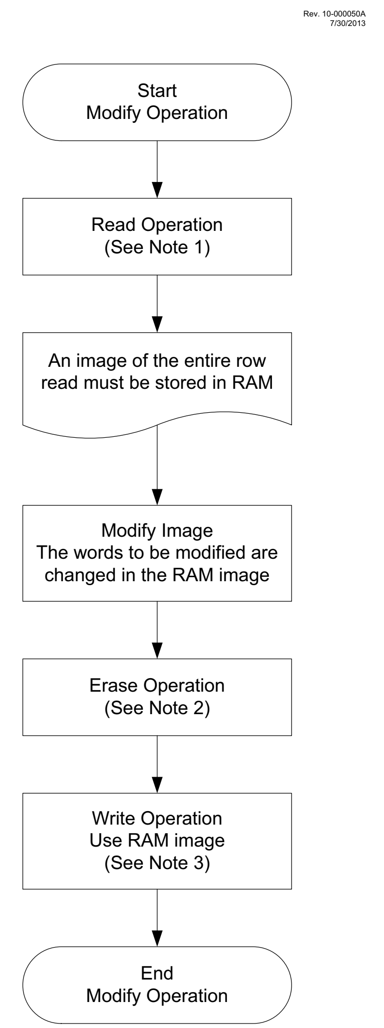
```
; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the Least Significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
        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       FSR0L           ;
        MOVLW       HIGH DATA_ADDR  ; Load initial data address
        MOVWF       FSR0H           ;
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,WREN     ; 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       0x1F            ;
        BTFSC       STATUS,Z        ; Exit if last of 32 words,
        GOTO        START_WRITE     ;

  Required Sequence
        MOVLW       55h             ; Start of required write sequence:
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin write
        NOP                         ; NOP instructions are forced as processor
                                    ; loads program memory write latches
        NOP                         ;

        INCF        PMADRL,F        ; Still loading latches Increment address
        GOTO        LOOP            ; Write next latches

START_WRITE
        BCF         PMCON1,LWLO     ; No more loading latches - Actually start Flash program
                                    ; memory write

  Required Sequence
        MOVLW       55h             ; Start of required write sequence:
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin write
        NOP                         ; NOP instructions are forced as processor writes
                                    ; all the program memory write latches simultaneously
        NOP                         ; to program memory.
                                    ; After NOPs, the processor
                                    ; stalls until the self-write process in complete
                                    ; after write processor continues with 3rd instruction
        BCF         PMCON1,WREN     ; Disable writes
        BSF         INTCON,GIE      ; Enable interrupts
```

10.3 Modifying Flash Program Memory

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

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

FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



Note 1: See Figure 10-2.
2: See Figure 10-4.
3: See Figure 10-6.

10.4 User ID, Device ID and Configuration Word Access

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

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

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

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8006h/8005h	Device ID/Revision ID	Yes	No
8007h-8008h	Configuration Words 1 and 2	Yes	No

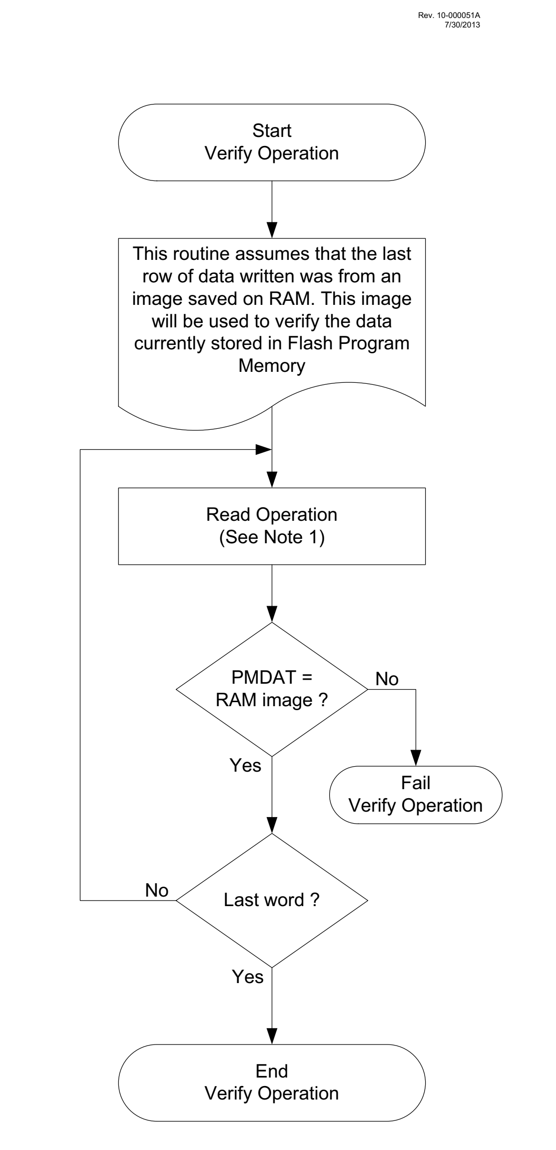
EXAMPLE 10-4: CONFIGURATION WORD AND DEVICE ID ACCESS

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  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



Note 1: See Figure 10-1.

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
PMDAT<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 **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	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PMADR<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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’.

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

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(3)	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **Unimplemented:** Read as ‘1’

bit 6 **CFGS:** Configuration Select bit
1 = Access Configuration, User ID and Device ID Registers
0 = Access Flash program memory

bit 5 **LWLO:** Load Write Latches Only bit(3)
1 = Only the addressed program memory write latch is loaded/updated on the next WR command
0 = The addressed program memory write latch is loaded/updated and a write of all program memory write latches will be initiated on the next WR command

bit 4 **FREE:** Program Flash Erase Enable bit
1 = Performs an erase operation on the next WR command (hardware cleared upon completion)
0 = Performs an write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit(2)
1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write ‘1’) of the WR bit).
0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
1 = Allows program/erase cycles
0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit
1 = Initiates a program Flash program/erase operation.
The operation is self-timed and the bit is cleared by hardware once operation is complete.
The WR bit can only be set (not cleared) in software.
0 = Program/erase operation to the Flash is complete and inactive.

bit 0 **RD:** Read Control bit
1 = Initiates a program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate a program Flash read.

Note 1: Unimplemented bit, read as ‘1’.
2: The WRERR bit is automatically set by hardware when a program memory write or erase operation is started (WR = 1).
3: The LWLO bit is ignored during a program memory erase operation (FREE = 1).

REGISTER 10-6: PMCON2: PROGRAM MEMORY CONTROL 2 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 bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

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	87
PMCON1	—[1]	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	116
PMCON2	Program Memory Control Register 2								117
PMADRL	PMADRL<7:0>								115
PMADRH	—[1]	PMADRH<6:0>							115
PMDATL	PMDATL<7:0>								115
PMDATH	—	—	PMDATH<5:0>						115

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
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOR}}$	BORV	STVREN	PLLEN	57
	7:0	—	—	—	—	—	PPS1WAY	WRT<1:0>		

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

11.0 I/O PORTS

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

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- 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 11-1: PORT AVAILABILITY PER DEVICE

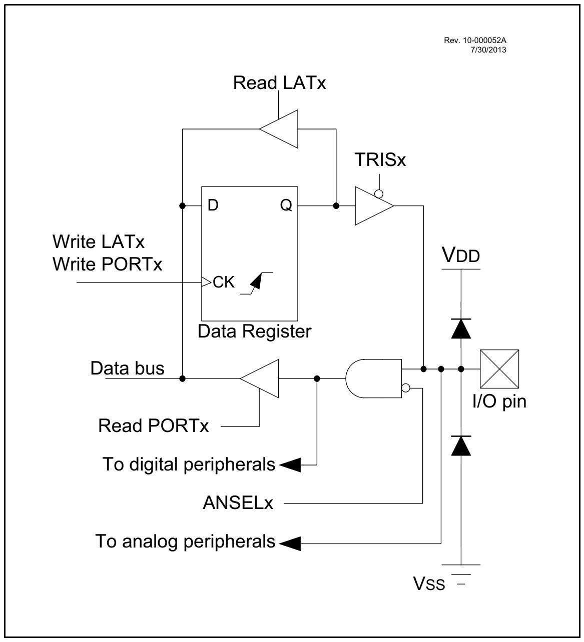
Device	PORTA	PORTB	PORTC
PIC16(L)F1574	●		●
PIC16(L)F1575	●		●
PIC16(L)F1578	●	●	●
PIC16(L)F1579	●	●	●

The Data Latch (LATx registers) is useful for read-modify-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 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



11.1 PORTA Registers

11.1.1 DATA REGISTER

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 11-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). The exception is RA3, which is input-only and its TRIS bit will always read as '1'. Example 11-1 shows how to initialize an I/O port.

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

11.1.2 DIRECTION CONTROL

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

11.1.3 OPEN-DRAIN CONTROL

The ODCONA register (Register 11-6) 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.

11.1.4 SLEW RATE CONTROL

The SLRCONA register (Register 11-7) 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.

11.1.5 INPUT THRESHOLD CONTROL

The INLVLA register (Register 11-8) 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 Table 27-4 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.

11.1.6 ANALOG CONTROL

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

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

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

EXAMPLE 11-1: INITIALIZING PORTA

```
; This code example illustrates
; initializig the PORTA register. The
; other ports are initialized in the same
; manner.

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

11.1.7 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 **Section 12.0 "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 it is in Analog mode.

11.2 Register Definitions: PORTA

REGISTER 11-1: PORTA: PORTA REGISTER

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 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 **RA<5:0>**: PORTA I/O Value bits(1)
1 = Port pin is ≥ V_{IH}
0 = Port pin is ≤ V_{IL}

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 11-2: 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’.

REGISTER 11-3: LATA: PORTA DATA LATCH REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **LATA<5:4>**: RA<5:4> Output Latch Value bits(1)

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **LATA<2:0>**: RA<2:0> 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 11-4: 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.

REGISTER 11-5: WPUA: WEAK PULL-UP PORTA 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
—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
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 **WPUA<5:0>**: Weak Pull-up Register bits(3)
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{\text{WPUEN}}$ 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 11-6: 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)

REGISTER 11-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

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 = 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 **SLRA<5:4>:** PORTA Slew Rate Enable bits
For RA<5:4> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SLRA<2:0>:** PORTA Slew Rate Enable bits
For RA<2:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 11-8: 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(1)	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

Note 1: The INLVLA3 bit selects the input type on this pin only when the $\overline{MCLR}$ function is not selected. When the $\overline{MCLR}$ function is selected, the input type for this pin will be ST.

TABLE 11-2: 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	122
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	124
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	122
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	123
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			180
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	121
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	124
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	123

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

TABLE 11-3: SUMMARY OF CONFIGURATION WORD WITH PORTA

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		FOSC<2:0>			

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

11.3 PORTB Registers (PIC16(L)F1578/9 only)

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 11-10). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 11-9) 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).

11.3.1 DIRECTION CONTROL

The TRISB register (Register 11-10) 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 inputs always read '0'.

11.3.2 OPEN-DRAIN CONTROL

The ODCONB register (Register 11-14) 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.

11.3.3 SLEW RATE CONTROL

The SLRCONB register (Register 11-15) 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.

11.3.4 INPUT THRESHOLD CONTROL

The INLVLB register (Register 11-16) 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 Table 27-4 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.

11.3.5 ANALOG CONTROL

The ANSELB register (Register 11-12) 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 ANSELB set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

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

11.3.6 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 **Section 12.0 "Peripheral Pin Select (PPS) Module"** for more information. Analog input functions, such as ADC and op amp 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 may continue to control the pin when it is in Analog mode.

11.4 Register Definitions: PORTB

REGISTER 11-9: PORTB: PORTB REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
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 **RB<7:4>**: PORTB General Purpose I/O Pin bits(1)
1 = Port pin is $\geq V_{IH}$
0 = Port pin is $\leq V_{IL}$

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 11-10: 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’

REGISTER 11-11: LATB: PORTB DATA LATCH REGISTER

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>**: PORTB 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 11-12: 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
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

bit 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.

REGISTER 11-13: WPUB: WEAK PULL-UP PORTB 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
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
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 **WPUB<7:4>**: Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

bit 3-0 **Unimplemented:** Read as ‘0’

Note 1: Global $\overline{\text{WPUEN}}$ 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 11-14: 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’

REGISTER 11-15: SLRCONB: PORTB SLEW RATE 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
SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—
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 **SLRB<7:4>:** PORTB Slew Rate Enable bits
For RB<7:4> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 11-16: 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 11-4: 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	—	—	—	—	128
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	129
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	127
ODCONB	ODB7	ODB6	ODB5	ODB4	—	—	—	—	129
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	127
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	129
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	129
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	128

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

11.5 PORTC Registers

11.5.1 DATA REGISTER

PORTC is a 6-bit wide bidirectional port in the PIC16(L)F1574/5 device and 8-bit wide bidirectional port in the PIC16(L)F1578/9 device. The corresponding data direction register is TRISC (Register 11-18). Setting a TRISC bit (= `1`) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= `0`) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 11-17) 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).

11.5.2 DIRECTION CONTROL

The TRISC register (Register 11-18) 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 inputs always read '0'.

11.5.3 INPUT THRESHOLD CONTROL

The INLVLC register (Register 11-24) 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 Table 27-4 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.

11.5.4 OPEN DRAIN CONTROL

The ODCONC register (Register 11-22) 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.

11.5.5 SLEW RATE CONTROL

The SLRCONC register (Register 11-23) 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.

11.5.6 ANALOG CONTROL

The ANSELC register (Register 11-20) 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.

11.5.7 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 **Section 12.0 "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 may continue to control the pin when it is in Analog mode.

11.6 Register Definitions: PORTC

REGISTER 11-17: 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[2]	RC6[2]	RC5	RC4	RC3	RC2	RC1	RC0
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 **RC<7:0>**: PORTC General Purpose I/O Pin bits[1, 2]
`1` = Port pin is $\geq V_{IH}$
`0` = Port pin is $\leq V_{IL}$

Note 1: Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.
2: RC<7:6> are available on PIC16(L)F1578/9 only.

REGISTER 11-18: 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> are available on PIC16(L)F1578/9 only.

REGISTER 11-19: 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>**: PORTC Output Latch Value bits[1]

Note 1: LATC<7:6> are available on PIC16(L)F1578/9 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 11-20: 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(2)	ANSC6(2)	—	—	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, 2)
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

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

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.
2: ANSC<7:6> are available on PIC16(L)F1578/9 only.

REGISTER 11-21: WPUC: WEAK PULL-UP PORTC 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
WPUC7(3)	WPUC6(3)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
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 **WPUC<7:0>**: Weak Pull-up Register bits(3)
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{\text{WPUEN}}$ 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: WPUC<7:6> are available on PIC16(L)F1578/9 only.

REGISTER 11-22: 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> are available on PIC16(L)F1578/9 only.

REGISTER 11-23: SLRCONC: PORTC SLEW RATE 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
SLRC7[(1)]	SLRC6[(1)]	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
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 **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> are available on PIC16(L)F1578/9 only.

REGISTER 11-24: 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> are available on PIC16(L)F1578/9 only.

TABLE 11-5: 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	133
INLVLC	INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	134
LATC	LATC7(1)	LATC6(1)	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	132
ODCONC	ODC7(1)	ODC6(1)	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	134
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			180
PORTC	RC7(1)	RC6(1)	RC5	RC4	RC3	RC2	RC1	RC0	132
SLRCONC	SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	134
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	132
WPUC	WPUC7(1)	WPUC6(1)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	133

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTC.
Note 1: PIC16(L)F1578/9 only.

NOTES:

12.0 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 12-1.

12.1 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 12-1.

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

12.2 PPS Outputs

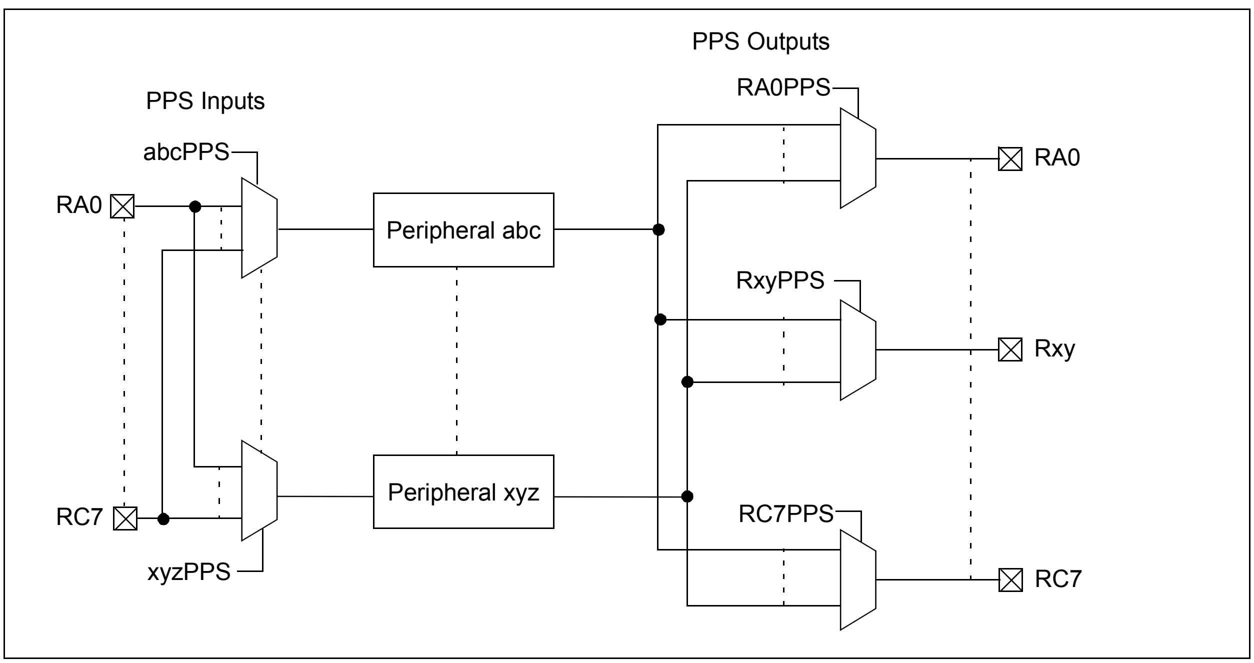
Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control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

- EUSART (synchronous operation)
- MSSP (I^2C)
- CWG (auto-shutdown)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 12-2.

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

FIGURE 12-1: SIMPLIFIED PPS BLOCK DIAGRAM



12.3 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 (I^2C)

> **Note:** The I^2C default input pins are I^2C and SMBus compatible and are the only pins on the device with this compatibility.

12.4 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 12-1.

EXAMPLE 12-1: 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
```

12.5 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.

12.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

12.7 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 12-1.

12.8 Register Definitions: PPS Input Selection

REGISTER 12-1: 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	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 peripheral

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-3 **xxxPPS<4:3>:** Peripheral xxx Input PORT Selection bits
11 = Reserved. Do not use.
10 = Peripheral input is PORTC
01 = Peripheral input is PORTB(2)
00 = Peripheral input is PORTA

bit 2-0 **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)

Note 1: See Table 12-1 for xxxPPS register list and Reset values.
2: PIC16(L)F1578/9 only.

REGISTER 12-2: 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 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 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits
Selection code determines the output signal on the port pin.
See Table 12-2 for the selection codes

REGISTER 12-3: 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	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 **Unimplemented:** Read as '0'

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.

TABLE 12-1: PPS INPUT REGISTER RESET VALUES

Peripheral	xxxPPS Register	Default Pin Selection		Reset Value (xxxPPS<4:0>)	
		PIC16(L)F1578/9	PIC16(L)F1574/5	PIC16(L)F1578/9	PIC16(L)F1574/5
Interrupt-on-change	INTPPS	RA2	RA2	00010	00010
Timer 0clock	T0CKIPPS	RA2	RA2	00010	00010
Timer 1clock	T1CKIPPS	RA5	RA5	00101	00101
Timer 1 gate	T1GPPS	RA4	RA4	00100	00100
CWG1	CWG1INPPS	RA2	RA2	00010	00010
EUSART RX	RXPPS	RB5	RC5	01101	10101
EUSART CK	CKPPS	RB7	RC4	01111	10100
ADC Auto-Conversion Trigger	ADCACTPPS	RC4	RC4	10100	10100

Example: ADCACTPPS = 0x14 selects RC4 as the ADC Auto-Conversion Trigger input.

TABLE 12-2: AVAILABLE PORTS FOR OUTPUT BY PERIPHERAL(2)

RxyPPS<3:0>	Output Signal	PIC16(L)F1578/9			PIC16(L)F1574/5	
		PORTA	PORTB	PORTC	PORTA	PORTC
1111	Reserved	—	—	—	—	—
1110	Reserved	—	—	—	—	—
1101	Reserved	—	—	—	—	—
1100	Reserved	—	—	—	—	—
1011	Reserved	—	—	—	—	—
1010	DT(1)	●	●	●	●	●
1001	TX/CK(1)	●	●	●	●	●
1000	CWG1OUTB(1)	●	●	●	●	●
0111	CWG1OUTA(1)	●	●	●	●	●
0110	PWM4_out	●	●	●	●	●
0101	PWM3_out	●	●	●	●	●
0100	PWM2_out	●	●	●	●	●
0011	PWM1_out	●	●	●	●	●
0010	sync_C2OUT	●	●	●	●	●
0001	sync_C1OUT	●	●	●	●	●
0000	LATxy	●	●	●	●	●

Note 1: TRIS control is overridden by the peripheral as required.
2: Unsupported peripherals will output a ‘0’.

TABLE 12-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	140
INTPPS	—	—	—	INTPPS<4:0>					139
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					139
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					139
T1GPPS	—	—	—	T1GPPS<4:0>					139
CWG1INPPS	—	—	—	CWG1INPPS<4:0>					139
RXPPS	—	—	—	RXPPS<4:0>					139
CKPPS	—	—	—	CKPPS<4:0>					139
ADCACTPPS	—	—	—	ADCACTPPS<4:0>					139
RA0PPS	—	—	—	—	RA0PPS<3:0>				139
RA1PPS	—	—	—	—	RA1PPS<3:0>				139
RA2PPS	—	—	—	—	RA2PPS<3:0>				139
RA4PPS	—	—	—	—	RA4PPS<3:0>				139
RA5PPS	—	—	—	—	RA5PPS<3:0>				139
RB4PPS(1)	—	—	—	—	RB4PPS<3:0>				139
RB5PPS(1)	—	—	—	—	RB5PPS<3:0>				139
RB6PPS(1)	—	—	—	—	RB6PPS<3:0>				139
RB7PPS(1)	—	—	—	—	RB7PPS<3:0>				139
RC0PPS	—	—	—	—	RC0PPS<3:0>				139
RC1PPS	—	—	—	—	RC1PPS<3:0>				139
RC2PPS	—	—	—	—	RC2PPS<3:0>				139
RC3PPS	—	—	—	—	RC3PPS<3:0>				139
RC4PPS	—	—	—	—	RC4PPS<3:0>				139
RC5PPS	—	—	—	—	RC5PPS<3:0>				139
RC6PPS(1)	—	—	—	—	RC6PPS<3:0>				139
RC7PPS(1)	—	—	—	—	RC7PPS<3:0>				139

Note 1: PIC16(L)F1578/9 only.

13.0 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 13-1 is a block diagram of the IOC module.

Note 1: PORTB available on PIC16(L)F1578/9 only.

13.1 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.

13.2 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.

13.3 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.

13.4 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 13-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

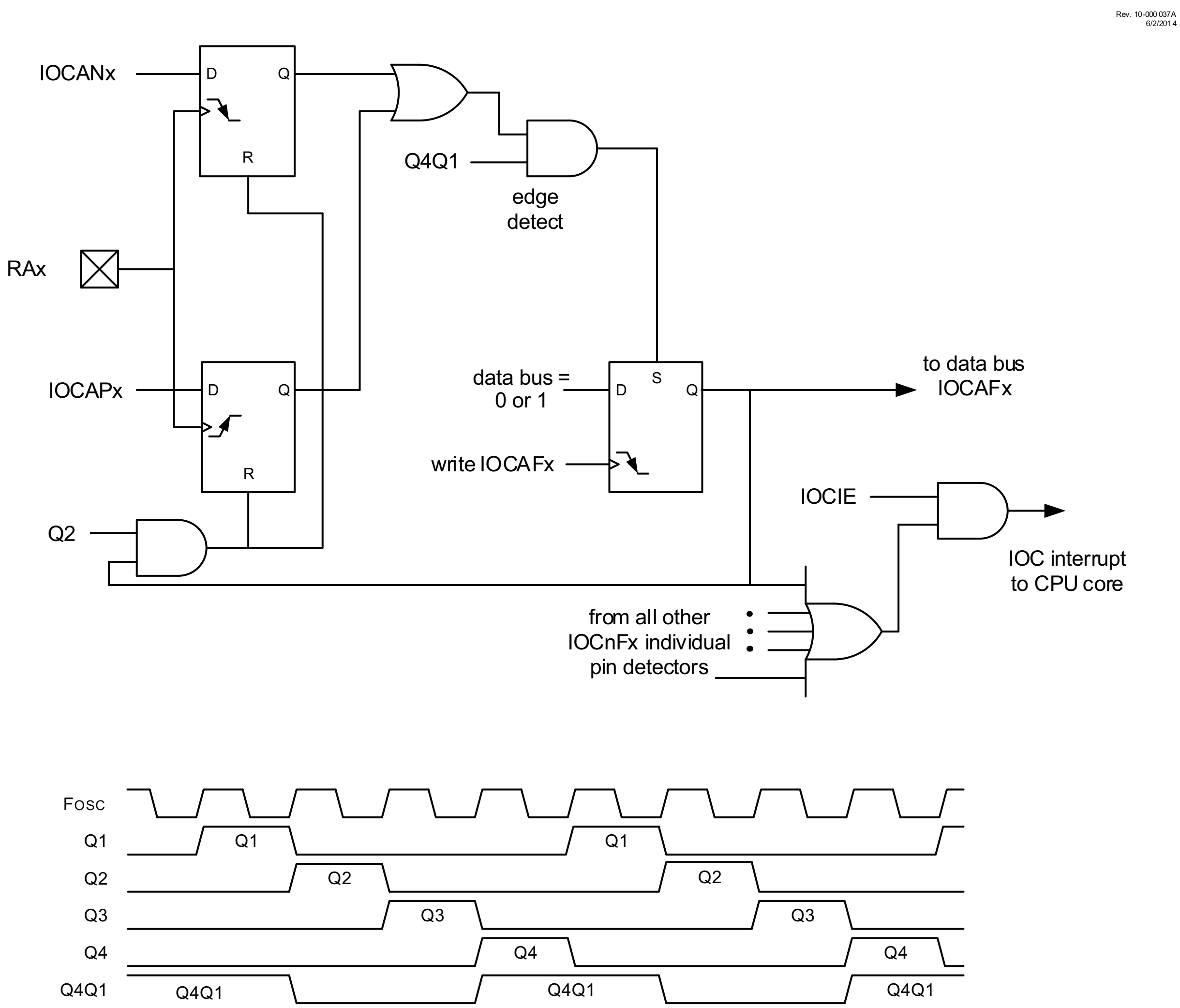
```
MOVLW   0xff
XORWF   IOCAF, W
ANDWF   IOCAF, F
```

13.5 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 13-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)



13.6 Register Definitions: Interrupt-on-Change Control

REGISTER 13-1: 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 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 **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 13-2: 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’

bit 5-0 **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 13-3: 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’

bit 5-0 **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 13-4: 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 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 **IOCBP<7:4>**: Interrupt-on-Change PORTB Positive Edge Enable bits
`1` = Interrupt-on-Change enabled on the pin for a positive going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.
`0` = Interrupt-on-Change disabled for the associated pin.

bit 3-0 **Unimplemented:** Read as ‘`0`’

Note 1: PORTB functions available on PIC16(L)F1578/9 devices only.

REGISTER 13-5: 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: PORTB functions available on PIC16(L)F1578/9 devices only.

REGISTER 13-6: 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: PORTB functions available on PIC16(L)F1578/9 devices only.

REGISTER 13-7: IOCCP: INTERRUPT-ON-CHANGE PORTC POSITIVE 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
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	

bit 7-0 **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)F1578/9 devices only.

REGISTER 13-8: 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 **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)F1578/9 devices only.

REGISTER 13-9: 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)F1578/9 devices only.

TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE

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	122
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	145
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	145
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	145
IOCBP(2)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	146
IOCBN(2)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	146
IOCBF(2)	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	146
IOCCP	IOCCP7(2)	IOCCP6(2)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	147
IOCCN	IOCCN7(2)	IOCCN6(2)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	147
IOCCF	IOCCF7(2)	IOCCF6(2)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	147
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121
TRISC	TRISC7(2)	TRISC7(2)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	132

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.
Note 1: Unimplemented, read as '1'.
2: PIC16(L)F1578/9 only.

14.0 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.

14.1 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 **Section 16.0 “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 **Section 18.0 “Comparator Module”** for additional information.

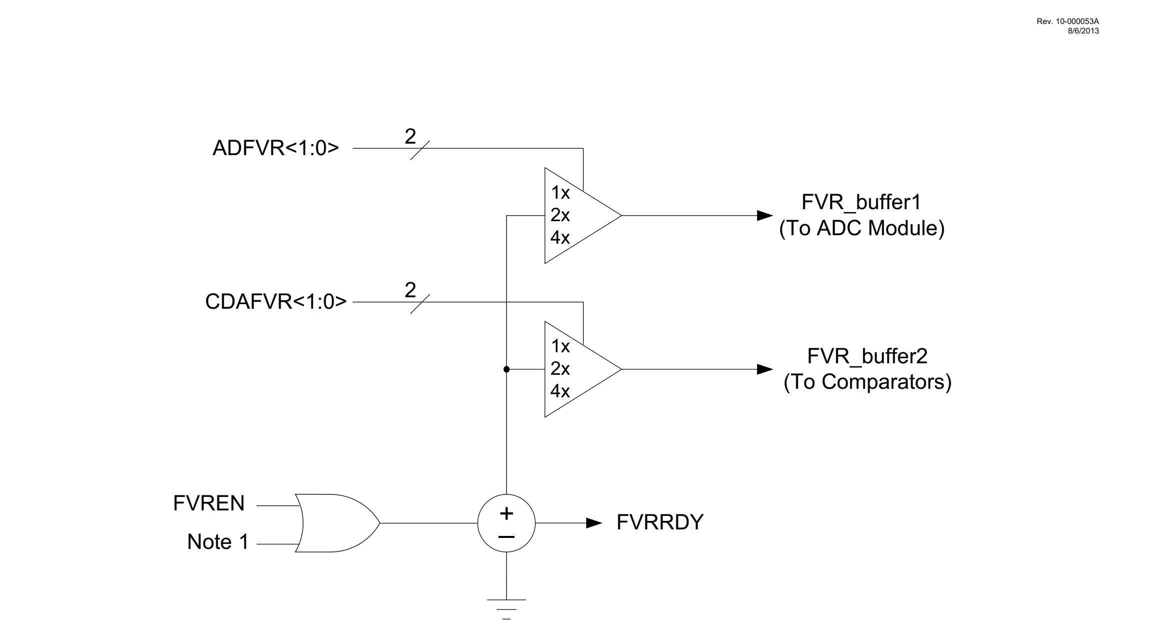
To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the Buffer Gain Selection bits.

14.2 FVR Stabilization Period

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

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.

FIGURE 14-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note 1: Any peripheral requiring the Fixed Reference (See Table 14-1)

TABLE 14-1: PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

Peripheral	Conditions	Description
HFINTOSC	FOSC<2:0> = 010 and IRCF<3:0> = 000x	INTOSC is active and device is not in Sleep.
BOR	BOREN<1:0> = 11	BOR always enabled.
	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 PIC16F1574/5/8/9 devices, when VREGPM = 1 and not in Sleep	The device runs off of the Low-Power Regulator when in Sleep mode.

14.3 Register Definitions: FVR Control

REGISTER 14-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

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]	CDAFVR<1:0>[1]		ADFVR<1:0>[1]	
bit 7							bit 0

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

bit 7 **FVREN:** Fixed Voltage Reference Enable bit[1]
- 1 = Fixed Voltage Reference is enabled
- 0 = Fixed Voltage Reference is disabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit[2]
- 1 = Fixed Voltage Reference output is ready for use
- 0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit[3]
- 1 = Temperature Indicator is enabled
- 0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit[3]
- 1 = $V_{OUT} = V_{DD} - 4V_T$ (High Range)
- 0 = $V_{OUT} = V_{DD} - 2V_T$ (Low Range)

bit 3-2 **CDAFVR<1:0>:** Comparator FVR Buffer Gain Selection bits[1]
- 11 = Comparator FVR Buffer Gain is 4x, with output $V_{CDAFVR} = 4x\ V_{FVR}$[4]
- 10 = Comparator FVR Buffer Gain is 2x, with output $V_{CDAFVR} = 2x\ V_{FVR}$[4]
- 01 = Comparator FVR Buffer Gain is 1x, with output $V_{CDAFVR} = 1x\ V_{FVR}$
- 00 = Comparator FVR Buffer is off

bit 1-0 **ADFVR<1:0>:** ADC FVR Buffer Gain Selection bit[1]
- 11 = ADC FVR Buffer Gain is 4x, with output $V_{ADFVR} = 4x\ V_{FVR}$[4]
- 10 = ADC FVR Buffer Gain is 2x, with output $V_{ADFVR} = 2x\ V_{FVR}$[4]
- 01 = ADC FVR Buffer Gain is 1x, with output $V_{ADFVR} = 1x\ V_{FVR}$
- 00 = ADC FVR Buffer is off

Note 1: To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the Buffer Gain Selection bits.

2: FVRRDY is always ‘1’ for the PIC16F1574/5/8/9 devices.

3: See **Section 15.0 “Temperature Indicator Module”** for additional information.

4: Fixed Voltage Reference output cannot exceed V_{DD}.

TABLE 14-2: 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	CDAFVR>1:0>		ADFVR<1:0>		151

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

15.0 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.

15.1 Circuit Operation

Figure 15-1 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 15-1 describes the output characteristics of the temperature indicator.

EQUATION 15-1: VOUT RANGES

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

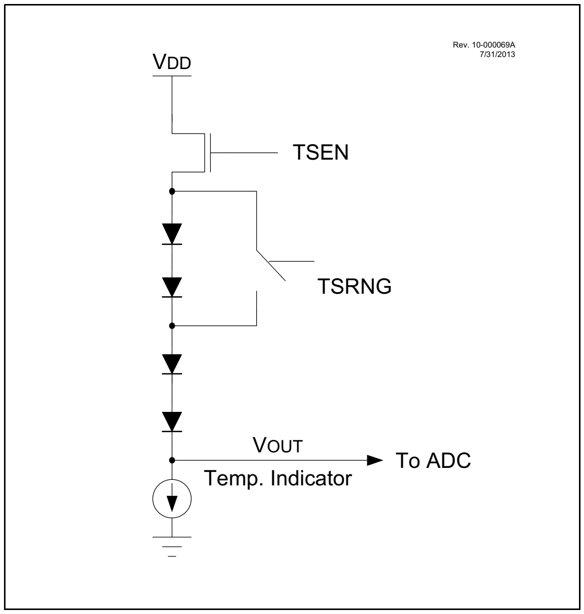
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.

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 15-1: TEMPERATURE CIRCUIT DIAGRAM



15.2 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 15-1 shows the recommended minimum VDD vs. range setting.

TABLE 15-1: RECOMMENDED VDD VS. RANGE

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

15.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

15.4 ADC Acquisition Time

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

TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

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

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

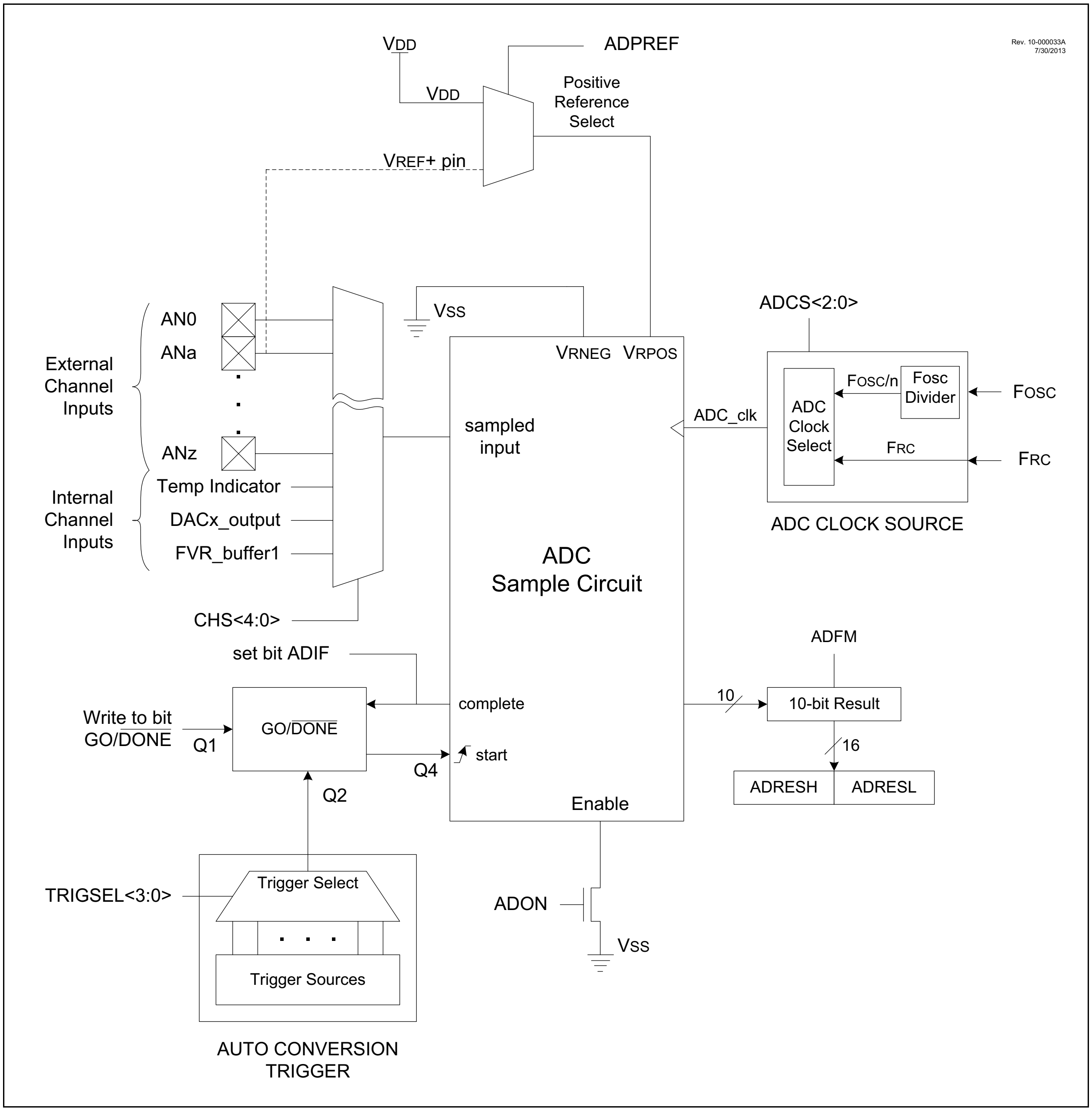
16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

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 16-1 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.

FIGURE 16-1: ADC BLOCK DIAGRAM



16.1 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

16.1.1 PORT CONFIGURATION

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

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

16.1.2 CHANNEL SELECTION

There are up to 15 channel selections available:

- AN<7:0> pins (PIC16(L)F1574/5 only)
- AN<11:0> pins (PIC16(L)F1578/9 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 **Section 16.2.6 "ADC Conversion Procedure"** for more information.

16.1.3 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

16.1.4 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 16-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the ADC conversion requirements in **Section 27.0 "Electrical Specifications"** for more information. Table 16-1 gives examples of appropriate ADC clock selections.

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

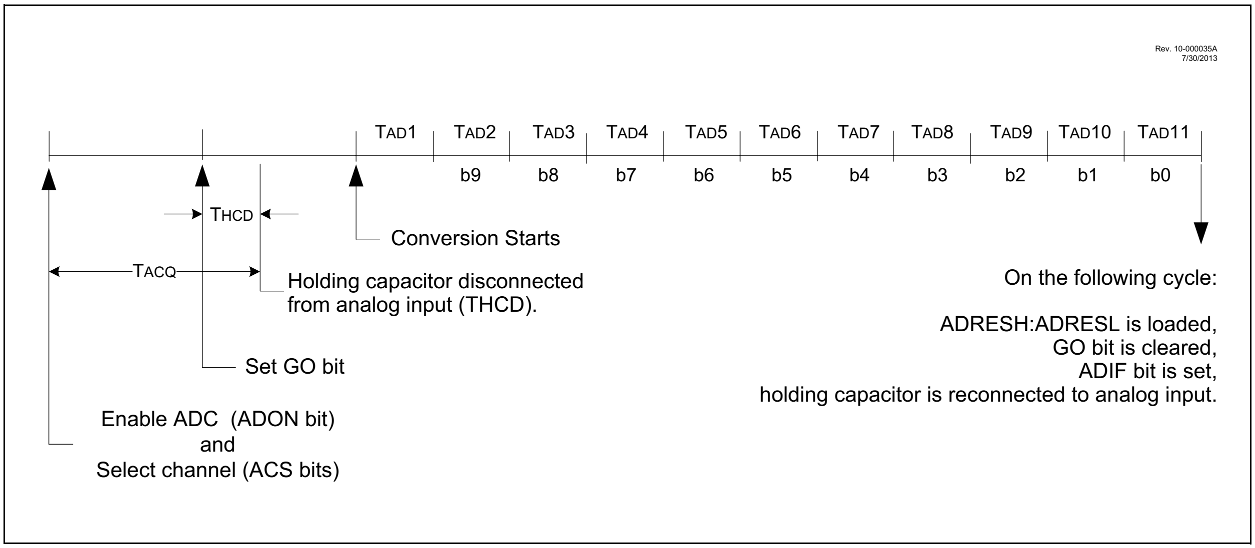
TABLE 16-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES

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

Legend: Shaded cells are outside of recommended range.

Note: 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 16-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES



16.1.5 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.

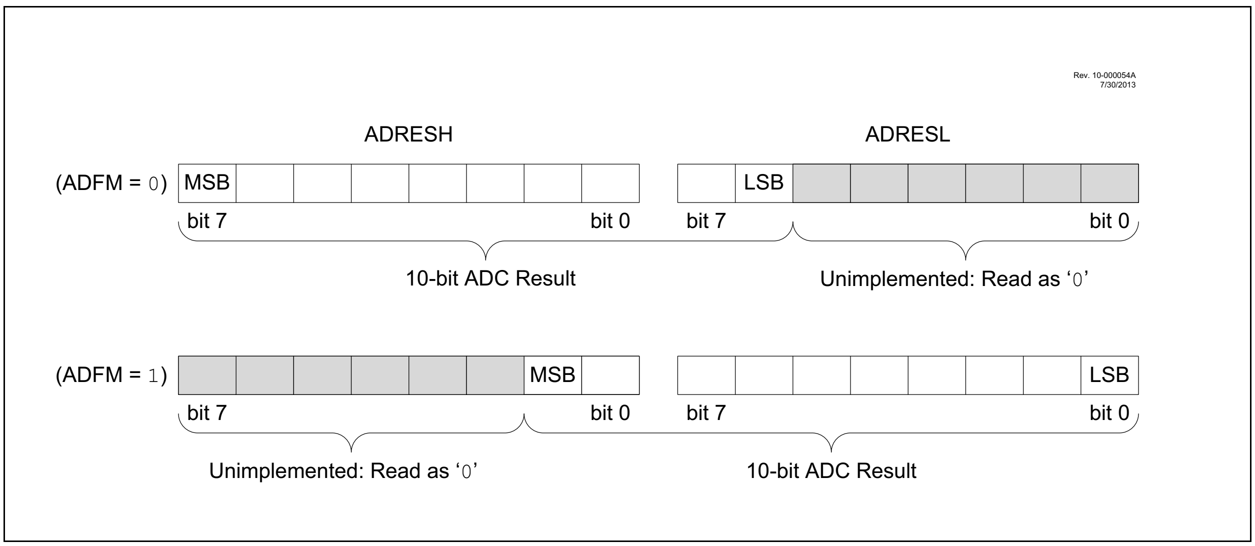
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the ADIE bit of the PIE1 register and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

16.1.6 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 16-3 shows the two output formats.

FIGURE 16-3: 10-BIT ADC CONVERSION RESULT FORMAT



16.2 ADC Operation

16.2.1 STARTING A CONVERSION

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

Note:	The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 16.2.6 "ADC Conversion Procedure"**.

16.2.2 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

16.2.3 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.

16.2.4 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.

16.2.5 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<3:0> 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.

The PWM module can trigger the ADC in two ways, directly through the PWMx_OF_match or through the interrupts generated by all four match signals. See **Section 23.0 "16-bit Pulse-Width Modulation (PWM) Module"**. If the interrupts are chosen, each enabled interrupt in PWMxINTE will trigger a conversion. Refer to Figure 16-4 for more information.

See Table 16-2 for auto-conversion sources.

FIGURE 16-4: 16-BIT PWM INTERRUPT BLOCK DIAGRAM

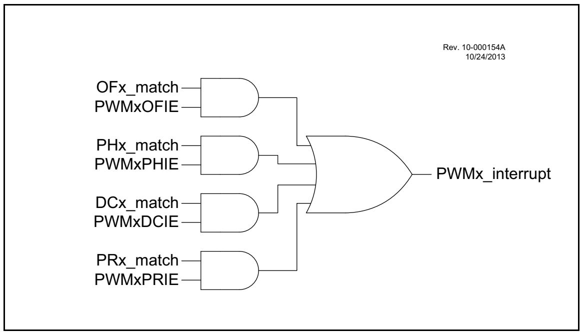


TABLE 16-2: AUTO-CONVERSION SOURCES

Source Peripheral	Signal Name
Timer0	T0_overflow
Timer1	T1_overflow
Timer2	T2_match
Comparator C1	C1OUT_sync
Comparator C2	C2OUT_sync
PWM1	PWM1_OF_match
PWM1	PWM1_interrupt
PWM2	PWM2_OF_match
PWM2	PWM2_interrupt
PWM3	PWM3_OF_match
PWM3	PWM3_interrupt
PWM4	PWM4_OF_match
PWM4	PWM4_interrupt
ADC Trigger	ADCACT
CWG Input Pin	CWGIN

16.2.6 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)
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/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

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

2: Refer to **Section 16.4 "ADC Acquisition Requirements"**.

EXAMPLE 16-1: ADC CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
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   ANSEL       ;
BSF       ANSEL,0     ;Set RA0 to analog
BANKSEL   ADCON0      ;
MOVLW     B'00000001' ;Select channel AN0
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   ADRESL      ;
MOVF      ADRESL,W    ;Read lower 8 bits
MOVWF     RESULTLO    ;Store in GPR space
```

16.3 Register Definitions: ADC Control

REGISTER 16-1: ADCON0: ADC CONTROL REGISTER 0

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS<4:0>					GO/$\overline{DONE}$	ADON
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as '0'

bit 6-2 **CHS<4:0>:** Analog Channel Select bits

00000 = AN0
00001 = AN1
00010 = AN2
00011 = AN3
00100 = AN4
00101 = AN5
00110 = AN6
00111 = AN7
01000 = AN8(4)
01001 = AN9(4)
01010 = AN10(4)
01011 = AN11(4)
01100 = Reserved. No channel connected.
•
•
•
11100 = Reserved. No channel connected.
11101 = Temperature Indicator(1)
11110 = DAC (Digital-to-Analog Converter)(2)
11111 = FVR (Fixed Voltage Reference) Buffer 1 Output(3)

bit 1 **GO/$\overline{DONE}$:** ADC Conversion Status bit

1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle.
This bit is automatically cleared by hardware when the ADC conversion has completed.
0 = ADC conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit

1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Note 1: See **Section 15.0 "Temperature Indicator Module"** for more information.
2: See **Section 17.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** for more information.
3: See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.
4: Available on PIC16(L)F1578/9 devices only.

REGISTER 16-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	—	ADPREF<1:0>	
bit 7							bit 0

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

bit 7 **ADFM:** ADC Result Format Select bit
1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.
0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC (clock supplied from an internal RC oscillator)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64
111 = FRC (clock supplied from an internal RC oscillator)

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
00 = VRPOS is connected to VDD
01 = Reserved
10 = VRPOS is connected to external VREF+ pin(1)
11 = VRPOS is connected to internal Fixed Voltage Reference (FVR)

Note 1: When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 27.0 "Electrical Specifications"** for details.

REGISTER 16-3: ADCON2: ADC CONTROL REGISTER 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
TRIGSEL<3:0>[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 7-4 **TRIGSEL<3:0>:** Auto-Conversion Trigger Selection bits[1]

- `0000` = No auto-conversion trigger selected
- `0001` = PWM1 – PWM1_interrupt
- `0010` = PWM2 – PWM2_interrupt
- `0011` = Timer0 – T0_overflow[2]
- `0100` = Timer1 – T1_overflow[2]
- `0101` = Timer2 – T2_match
- `0110` = Comparator C1 – C1OUT_sync
- `0111` = Comparator C2 – C2OUT_sync
- `1000` = PWM1 – PWM1_OF_match
- `1001` = PWM2 – PWM2_OF_match
- `1010` = PWM3 – PWM3_OF_match
- `1011` = PWM3 – PWM3_interrupt
- `1100` = PWM4 – PWM4_OF_match
- `1101` = PWM4 – PWM4_interrupt
- `1110` = ADC Auto-Conversion Trigger input pin
- `1111` = CWG input pin

bit 3-0 **Unimplemented:** Read as ‘0’

Note 1: This is a rising edge sensitive input for all sources.

2: Signal also sets its corresponding interrupt flag.

REGISTER 16-4: 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
ADRES<9:2>							
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 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

REGISTER 16-5: 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 0

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

bit 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

REGISTER 16-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADRES<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	

bit 7-2 **Reserved**: Do not use.

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

REGISTER 16-7: 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 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

16.4 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 16-5. 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 16-5. **The maximum recommended impedance for analog sources is 10 kΩ.** 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 16-1 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 16-1: ACQUISITION TIME EXAMPLE

Assumptions: *Temperature* = *50°C and external impedance of 10kΩ 5.0V VDD*

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\mu s + T_C + [(Temperature - 25°C)(0.05\mu s/°C)]$$

The value for TC can be approximated with the following equations:

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

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \quad ;\text{combining [1] and [2]}$$

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

Solving for TC:

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ \ln(1/2047)$$
$$= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega)\ \ln(0.0004885)$$
$$= 1.715\mu s$$

Therefore:

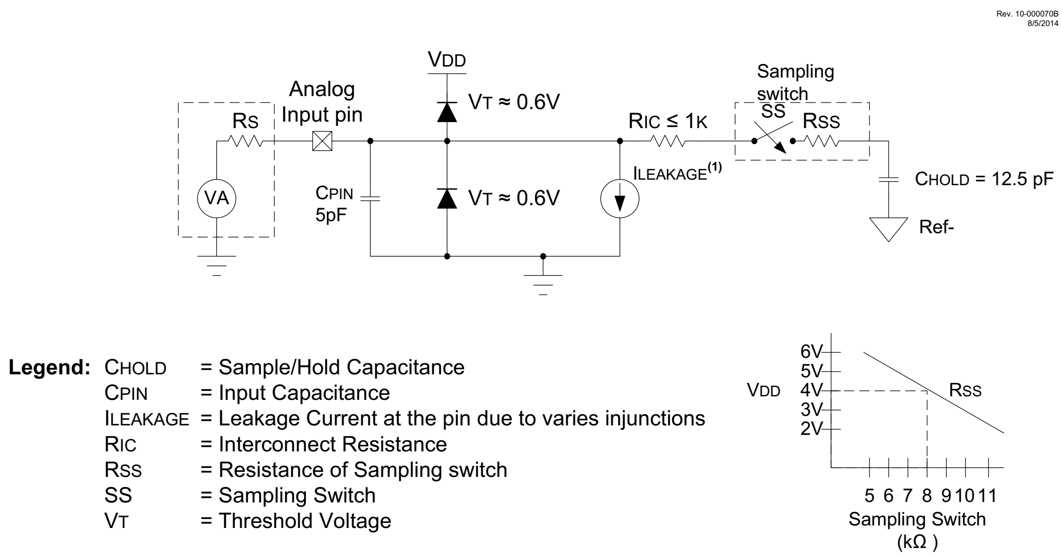
$$T_{ACQ} = 2\mu s + 1.715\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.96\mu s$$

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 kΩ. This is required to meet the pin leakage specification.

FIGURE 16-5: ANALOG INPUT MODEL



Note 1: Refer to **Section 27.0 "Electrical Specifications"**.

FIGURE 16-6: ADC TRANSFER FUNCTION

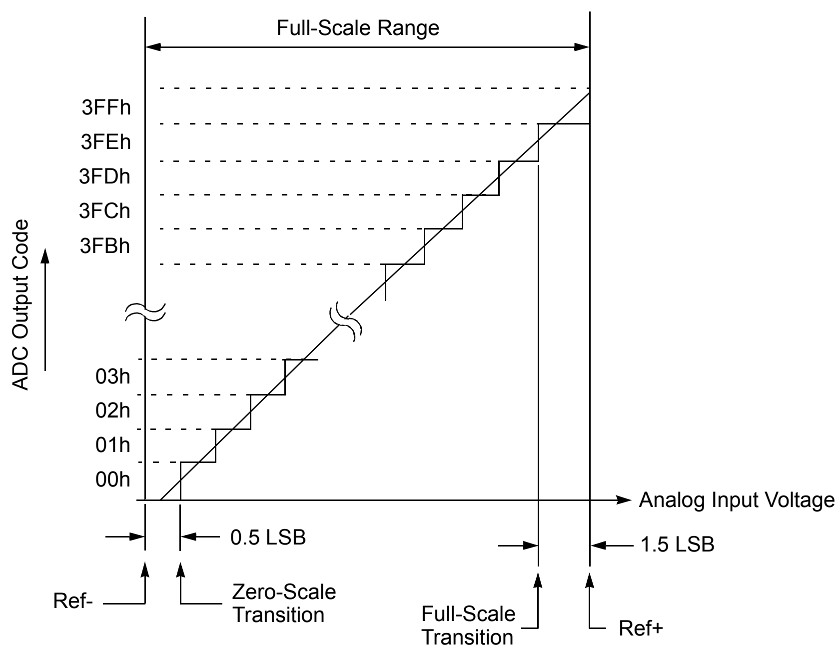


TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

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/$\overline{DONE}$	ADON	160
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		161
ADCON2	TRIGSEL<3:0>				—	—	—	—	162
ADRESH	ADC Result Register High								163, 164
ADRESL	ADC Result Register Low								163, 164
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	122
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		151

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’.

17.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The positive input source (VSOURCE+) of the DAC can be connected to:

- External VREF+ pin
- VDD supply voltage
- FVR_buffer1

The negative input source (VSOURCE-) of the DAC can be connected to:

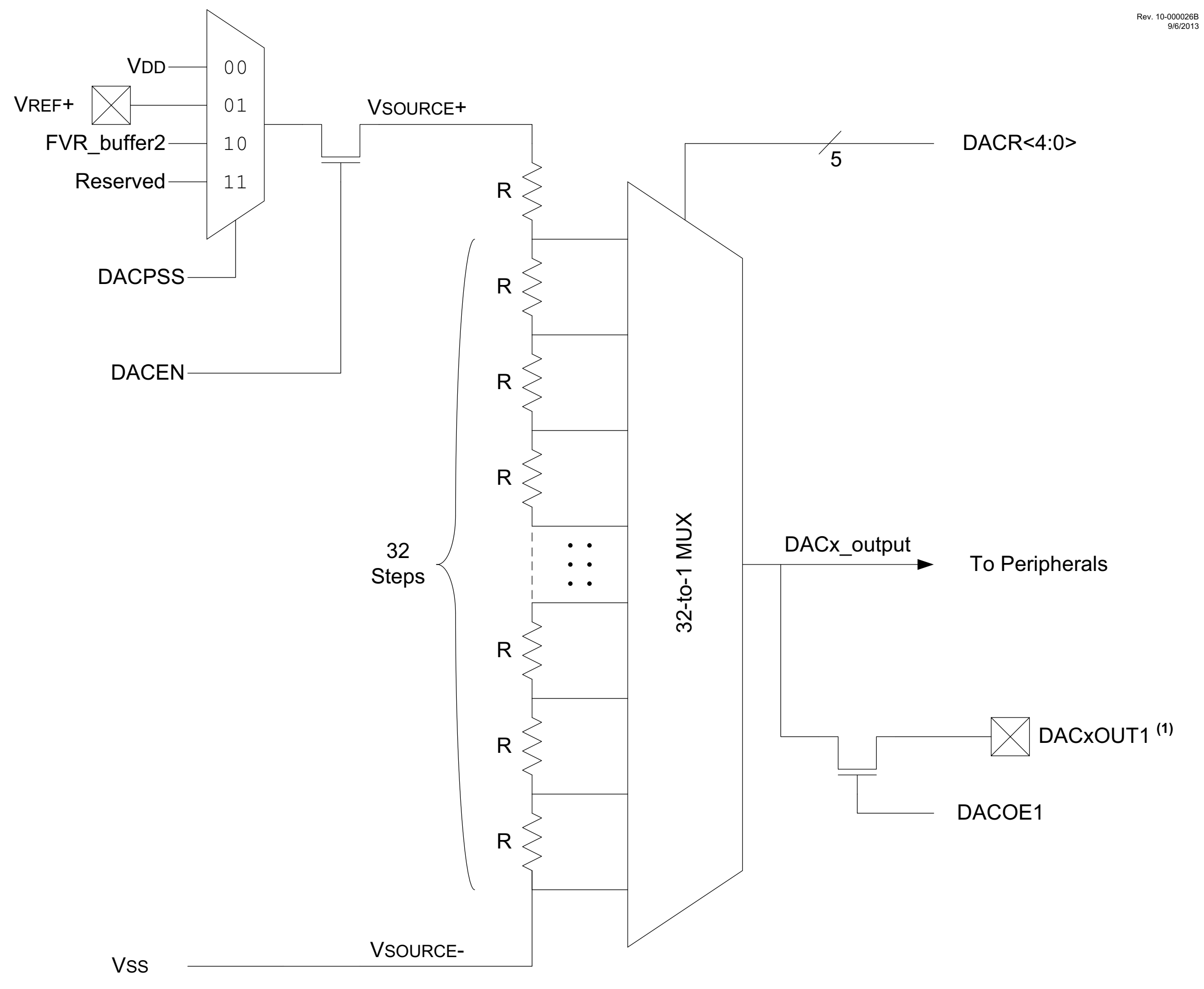
- Vss

The output of the DAC (DACx_output) can be selected as a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACxOUT1 pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the DACEN bit of the DACxCON0 register.

FIGURE 17-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



Note 1: The unbuffered DACx_output is provided on the DACxOUT pin(s).

17.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DACR<4:0> bits of the DACxCON1 register.

The DAC output voltage can be determined by using Equation 17-1.

17.2 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 Table 27-16.

17.3 DAC Voltage Reference Output

The unbuffered DAC voltage can be output to the DACxOUTn pin(s) by setting the respective DACOEn bit(s) of the DACxCON0 register. Selecting the DAC reference voltage for output on either DACxOUTn pin automatically overrides the digital output buffer, the weak pull-up and digital input threshold detector functions of that pin.

Reading the DACxOUTn pin when it has been configured for DAC reference voltage output will always return a '0'.

Note:	The unbuffered DAC output (DACxOUTn) is not intended to drive an external load.

17.4 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DACxCON0 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

17.5 Effects of a Reset

A device Reset affects the following:

- DACx is disabled.
- DACx output voltage is removed from the DACxOUTn pin(s).
- The DACR<4:0> range select bits are cleared.

EQUATION 17-1: DAC OUTPUT VOLTAGE

IF DACEN = 1

$$DACx\_output = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[4:0]}{2^5}\right) + V_{SOURCE-}$$

Note: See the DACxCON0 register for the available $V_{SOURCE+}$ and $V_{SOURCE-}$ selections.

17.6 Register Definitions: DAC Control

REGISTER 17-1: DACCON0: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
DACEN	—	DACOE	—	DACPSS<1:0>		—	—
bit 7							bit 0

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

bit 7 **DACEN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DAC voltage level is output on the DACOUT1 pin
0 = DAC voltage level is disconnected from the DACOUT1 pin

bit 4 **Unimplemented:** Read as ‘0’

bit 3-2 **DACPSS<1:0>:** DAC Positive Source Select bits
11 = Reserved
10 = FVR_buffer2
01 = VREF+ pin
00 = VDD

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 17-2: DACCON1: VOLTAGE REFERENCE CONTROL REGISTER 1

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
—	—	—	DACR<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **DACR<4:0>:** DAC Voltage Output Select bits

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>		—	—	170
DACCON1	—	—	—	DACR<4:0>					170

Legend: — = Unimplemented location, read as ‘0’. Shaded cells are not used with the DAC module.

18.0 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

18.1 Comparator Overview

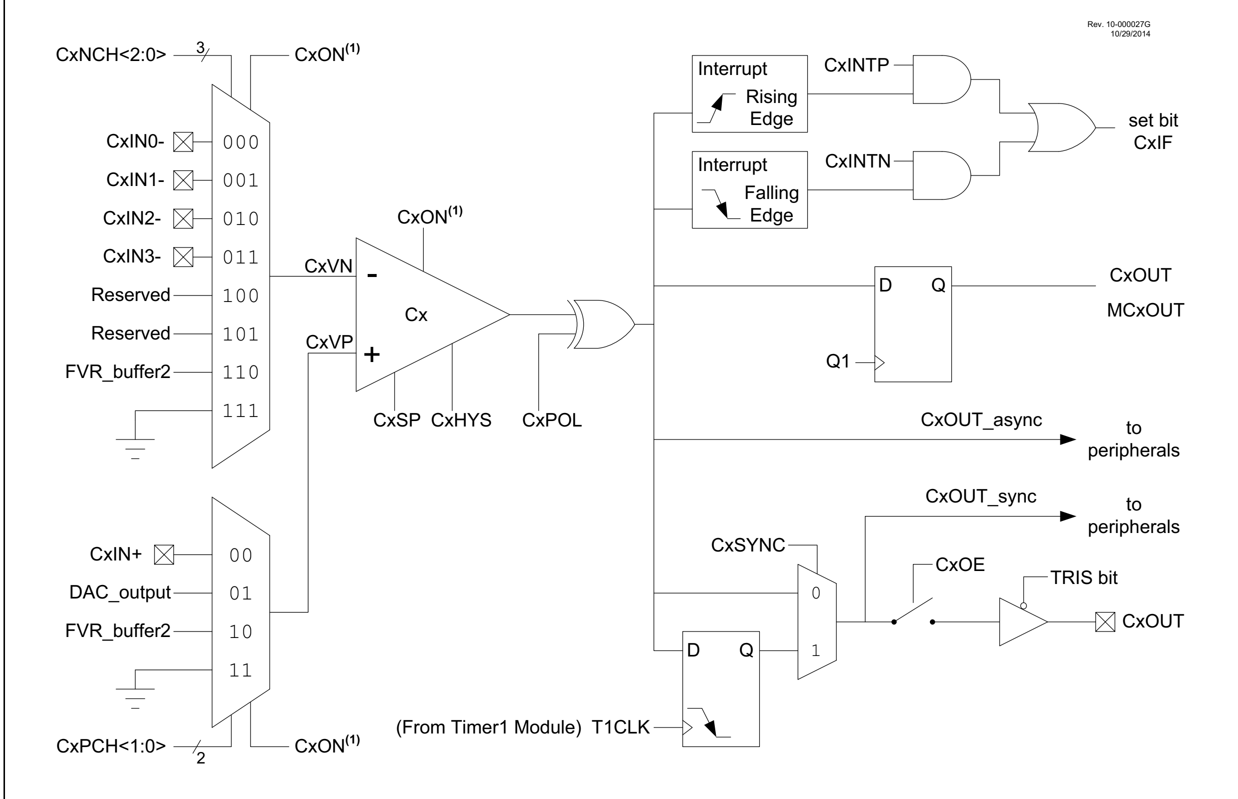
A single comparator is shown in Figure 18-2 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

The comparators available for this device are listed in Table 18-1.

TABLE 18-1: AVAILABLE COMPARATORS

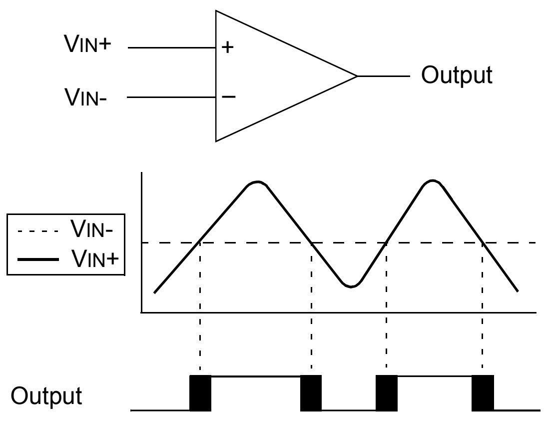
Device	C1	C2
PIC16(L)F1574	●	●
PIC16(L)F1575	●	●
PIC16(L)F1578	●	●
PIC16(L)F1579	●	●

FIGURE 18-1: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM



Note 1: When CxON = 0, all multiplexer inputs are disconnected and the Comparator will produce a '0' at the output.

FIGURE 18-2: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

18.2 Comparator Control

The comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 18-1) contains Control and Status bits for the following:

- Enable
- Output selection
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 register (see Register 18-2) contains Control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

18.2.1 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.

18.2.2 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
- DAC1_output
- FVR_buffer2
- VSS

See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information on the Fixed Voltage Reference module.

See **Section 17.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = `0`), all comparator inputs are disabled.

18.2.3 COMPARATOR NEGATIVE INPUT SELECTION

The CxNCH<2:0> bits of the CMxCON0 register direct one of the input sources to the comparator inverting input.

Note: To use CxIN+ and CxINx- 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.

18.2.4 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:

- Corresponding TRIS bit must be cleared
- CxON bit of the CMxCON0 register must be set

The synchronous comparator output signal (CxOUT_sync) is available to the following peripheral(s):

- Analog-to-Digital Converter (ADC)
- Timer1

The asynchronous comparator output signal (CxOUT_async) is available to the following peripheral(s):

- Complementary Waveform Generator (CWG)

Note: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

18.2.5 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 18-2 shows the output state versus input conditions, including polarity control.

TABLE 18-2: 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

18.2.6 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’.

18.3 Analog Input Connection Considerations

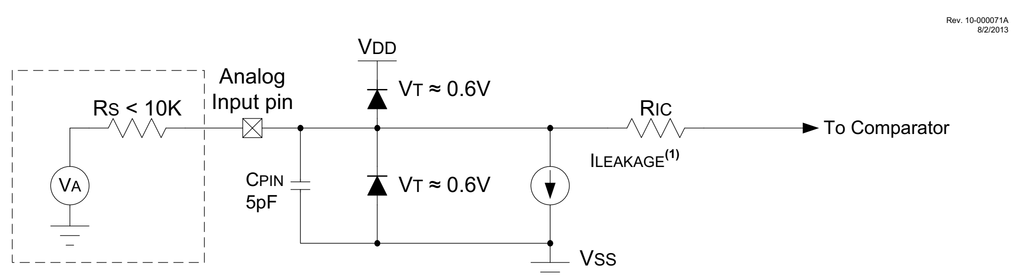
A simplified circuit for an analog input is shown in Figure 18-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward-biased and a latch-up may occur.

A maximum source impedance of 10 kΩ 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.

FIGURE 18-3: ANALOG INPUT MODEL



Legend:
CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage
VT = Threshold Voltage

Note 1: See **Section 27.0 “Electrical Specifications”**.

18.4 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 **Section 27.0 "Electrical Specifications"** for more information.

18.5 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 20.5 "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.

18.5.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from the Cx 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 18-2) and the Timer1 Block Diagram (Figure 20-1) for more information.

18.6 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, you must set the following bits:

- CxON and CxPOL 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.

18.7 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 **Section 27.0 "Electrical Specifications"** for more details.

18.8 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

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 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 **CxON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **CxOUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **CxPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **CxSP:** Comparator Speed/Power Select bit
1 = Comparator mode in normal-power, higher speed
0 = Comparator mode in low-power, low-speed

bit 1 **CxHYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

bit 0 **CxSYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous

REGISTER 18-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	CxNCH<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	

bit 7 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit

bit 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit

bit 5-4 **CxPCH<1:0>:** Comparator Positive Input Channel Select bits
11 = CxVP connects to VSS
10 = CxVP connects to FVR Voltage Reference
01 = CxVP connects to DAC Voltage Reference
00 = CxVP connects to CxIN+ pin

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **CxNCH<1:0>:** Comparator Negative Input Channel Select bits
111 = CxVN connects to GND
110 = CxVN connects to FVR Voltage Reference
101 = Reserved
100 = Reserved
011 = CxVN connects to CxIN3- pin
010 = CxVN connects to CxIN2- pin
001 = CxVN connects to CxIN1- pin
000 = CxVN connects to CxIN0- pin

REGISTER 18-3: 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 = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

TABLE 18-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE

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	122
CM1CON0	C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	175
CM1CON1	C1NTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			176
CM2CON0	C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	175
CM2CON1	C2NTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			176
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	176
DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>			—	170
DACCON1	—	—	—	DACR<4:0>					170
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE2	—	C2IE	C1IE	—	—	—	—	—	89
PIR2	—	C2IF	C1IF	—	—	—	—	—	92
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	121
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	122
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121

Legend: — = unimplemented location, read as ‘0’. Shaded cells are unused by the comparator module.

Note 1: Unimplemented, read as ‘1’.

19.0 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 19-1 is a block diagram of the Timer0 module.

19.1 Timer0 Operation

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

19.1.1 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.

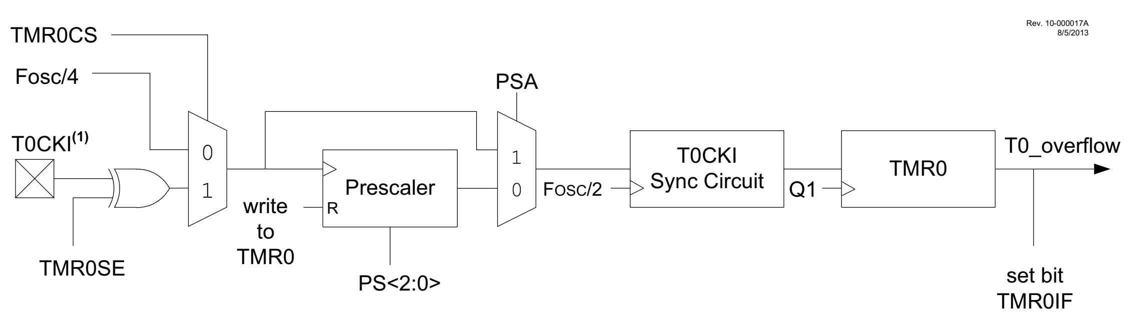
19.1.2 8-BIT COUNTER MODE

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.

FIGURE 19-1: TIMER0 BLOCK DIAGRAM



Note 1: The T0CKI prescale output frequency should not exceed FOSC/8.

19.1.3 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.

19.1.4 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.

19.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 27.0 "Electrical Specifications"**.

19.1.6 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.

19.2 Register Definitions: Option Register

REGISTER 19-1: OPTION_REG: 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
$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<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	

bit 7 **$\overline{\text{WPUEN}}$:** Weak Pull-Up Enable bit
1 = All weak pull-ups are disabled (except $\overline{\text{MCLR}}$, if it is enabled)
0 = Weak pull-ups are enabled by individual WPUx latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **TMR0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **TMR0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is not assigned to the Timer0 module
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	Timer0 Rate
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON2	TRIGSEL<3:0>				—	—	—	—	162
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			180
TMR0	Holding Register for the 8-bit Timer0 Count								178*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121

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'.

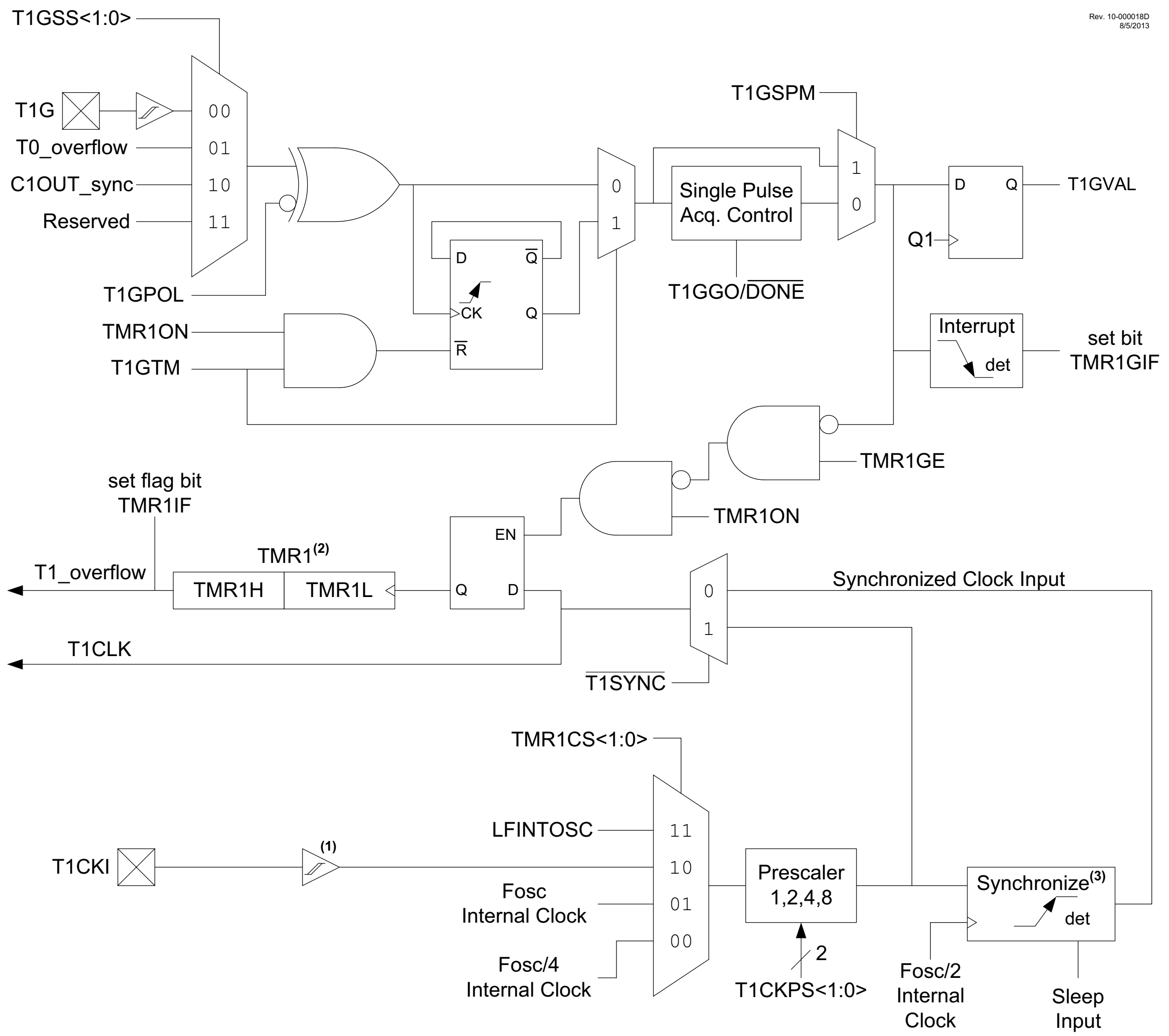
20.0 TIMER1 MODULE WITH GATE CONTROL

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

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- 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 20-1 is a block diagram of the Timer1 module.

FIGURE 20-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using T1CKI.
2: Timer1 register increments on rising edge.
3: Synchronize does not operate while in Sleep.

20.1 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 20-1 displays the Timer1 enable selections.

TABLE 20-1: TIMER1 ENABLE SELECTIONS

TMR1ON	TMR1GE	Timer1 Operation
0	0	Off
0	1	Off
1	0	Always On
1	1	Count Enabled

20.2 Clock Source Selection

The TMR1CS<1:0> bits of the T1CON register are used to select the clock source for Timer1. Table 20-2 displays the clock source selections.

20.2.1 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

20.2.2 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 20-2: CLOCK SOURCE SELECTIONS

TMR1CS<1:0>	T1OSCEN(1)	Clock Source
11	x	LFINTOSC
10	x	External Clocking on T1CKI Pin
01	x	System Clock (FOSC)
00	x	Instruction Clock (FOSC/4)

Note 1: T1OSC is not available on all devices.

20.3 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.

20.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 20.4.1 "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.

20.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

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

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

20.5 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.

20.5.1 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 20-3 for timing details.

TABLE 20-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

20.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 20-4. Source selection is controlled by the T1GSS<1:0> bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

TABLE 20-4: TIMER1 GATE SOURCES

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 (C1OUT_sync)(1)
11	Comparator 2 Output (C2OUT_sync)(1)

Note 1: Optionally synchronized comparator output.

20.5.2.1 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 gate circuitry.

20.5.2.2 Timer0 Overflow Gate Operation

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

20.5.3 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 20-4 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.

20.5.4 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/$\overline{\text{DONE}}$ 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/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/$\overline{\text{DONE}}$ bit is once again set in software. See Figure 20-5 for timing details.

If the Single Pulse Gate mode is disabled by clearing the T1GSPM bit in the T1GCON register, the T1GGO/$\overline{\text{DONE}}$ 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 20-6 for timing details.

20.5.5 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).

20.5.6 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).

20.6 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, you must set these bits:

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

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

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

20.7 Timer1 Operation During Sleep

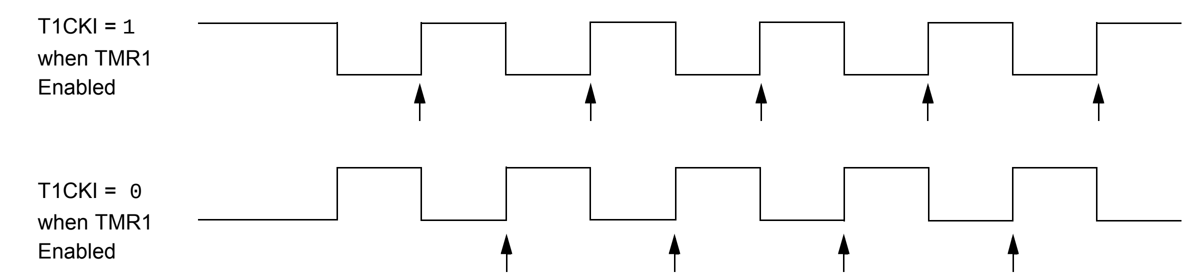
Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- 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
- $\overline{\text{T1SYNC}}$ 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.

Timer1 oscillator will continue to operate in Sleep regardless of the $\overline{\text{T1SYNC}}$ bit setting.

FIGURE 20-2: TIMER1 INCREMENTING EDGE



Note 1: Arrows indicate counter increments.

2: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

FIGURE 20-3: TIMER1 GATE ENABLE MODE

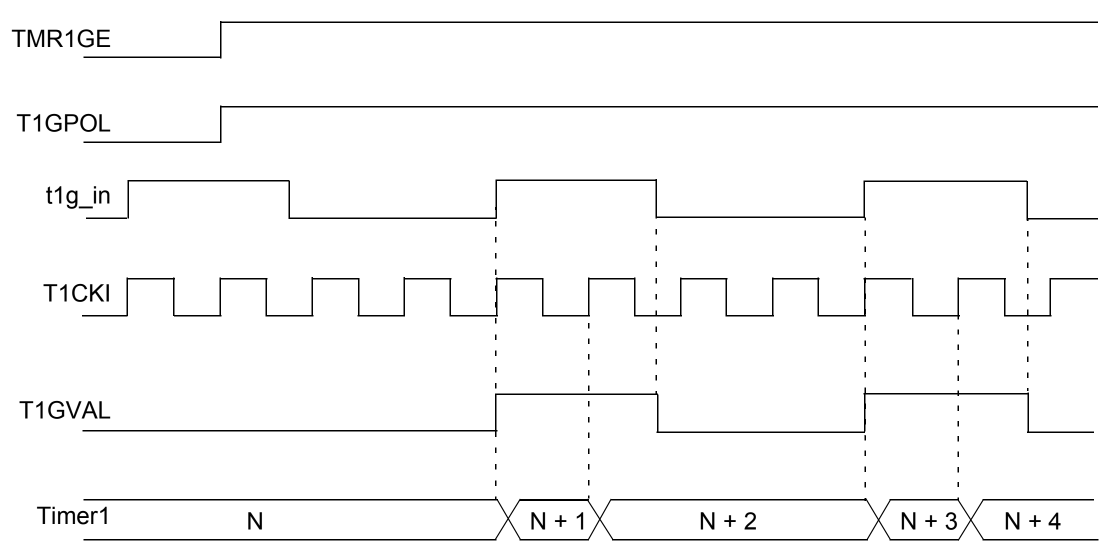


FIGURE 20-4: TIMER1 GATE TOGGLE MODE

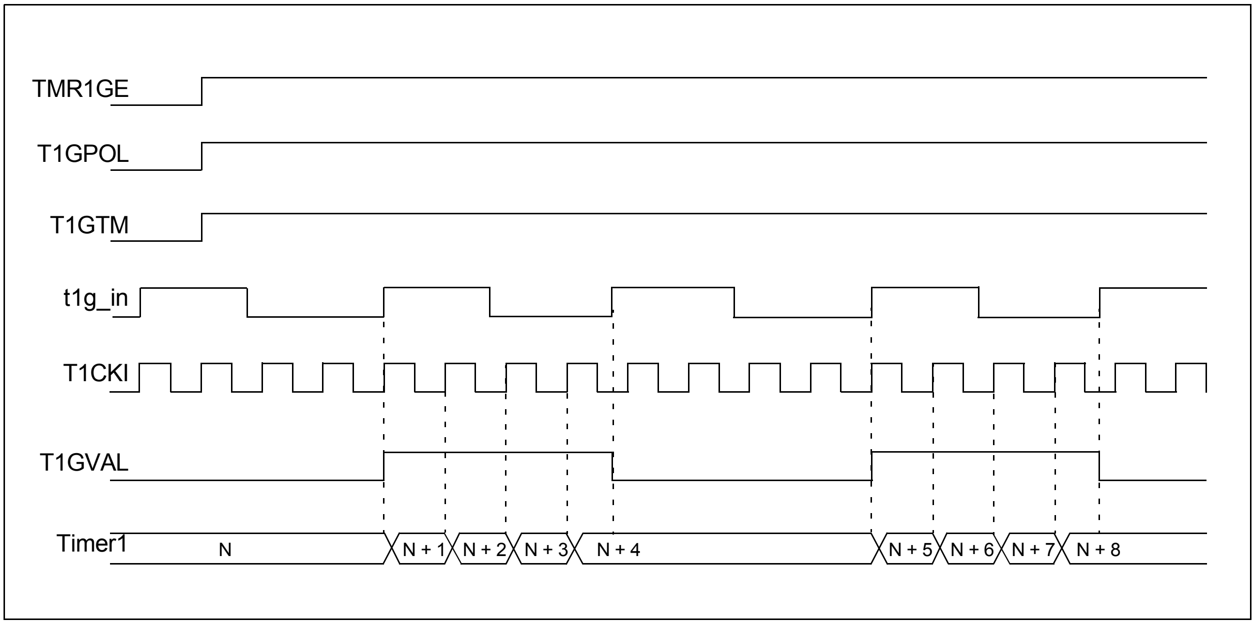


FIGURE 20-5: TIMER1 GATE SINGLE-PULSE MODE

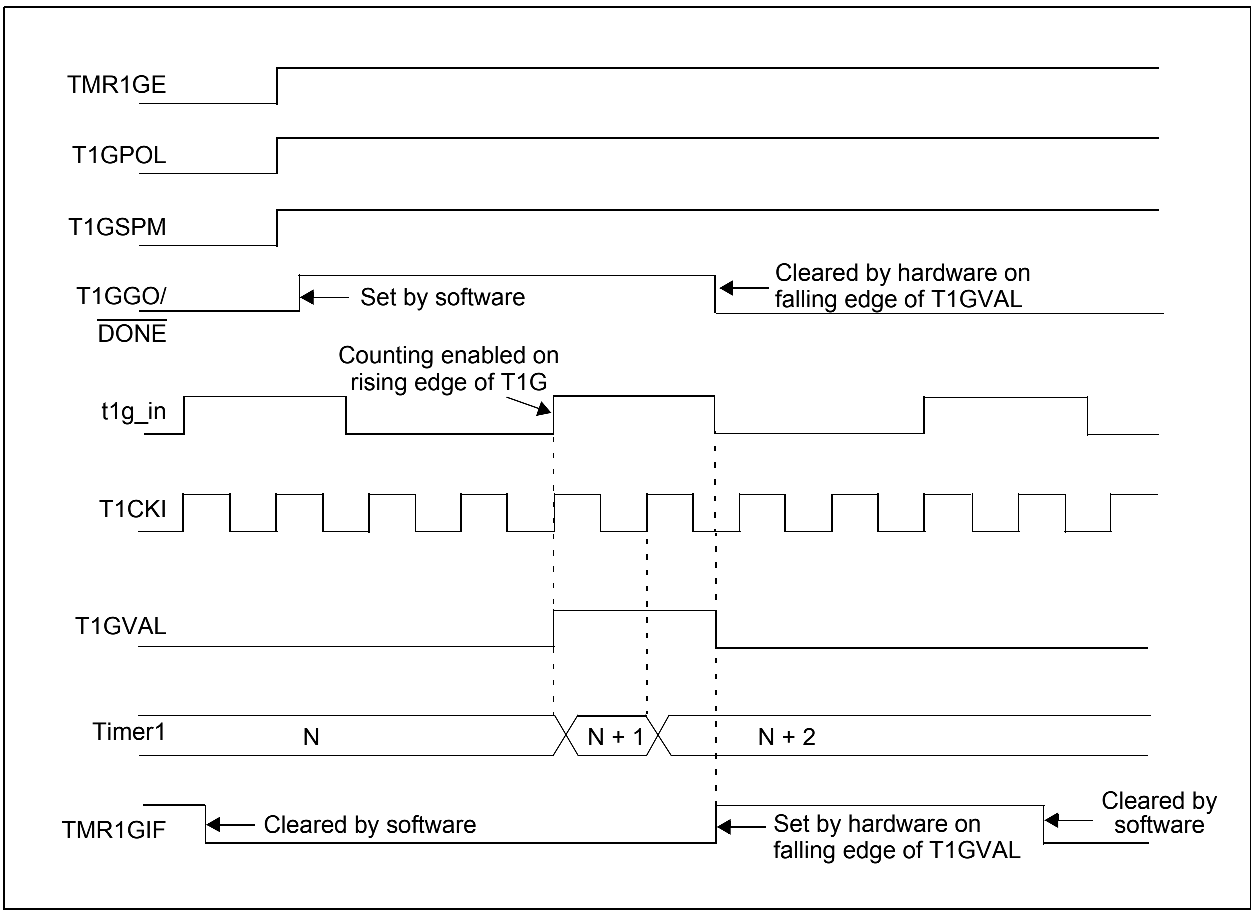
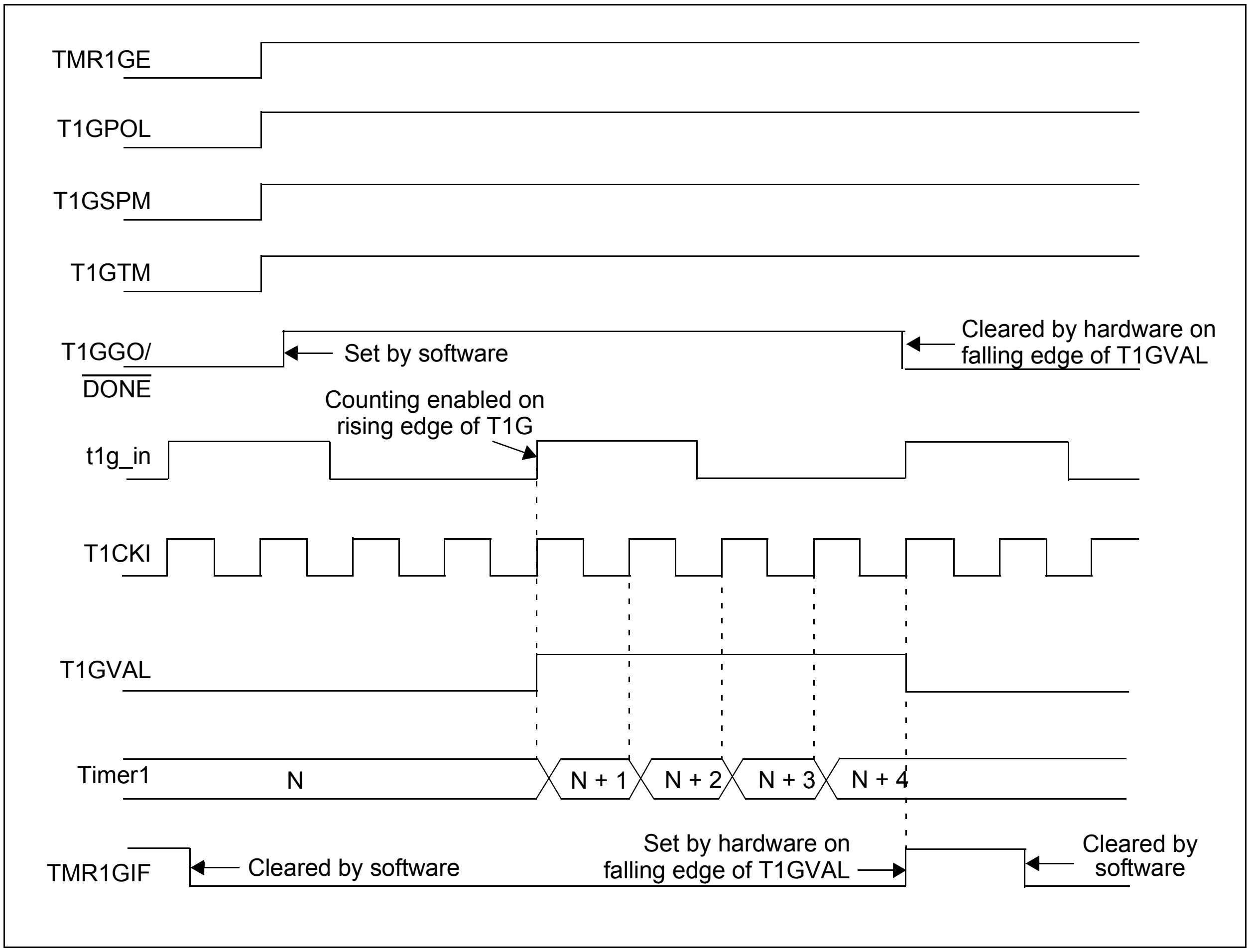


FIGURE 20-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



20.8 Register Definitions: Timer1 Control

REGISTER 20-1: 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>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON
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 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
- `11` = Timer1 clock source is LFINTOSC
- `10` = Timer1 clock source is T1CKI pin (on the rising edge)
- `01` = Timer1 clock source is system clock (FOSC)
- `00` = Timer1 clock source is instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
- `11` = 1:8 Prescale value
- `10` = 1:4 Prescale value
- `01` = 1:2 Prescale value
- `00` = 1:1 Prescale value

bit 3 **Unimplemented:** Read as ‘`0`’

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 Synchronization Control bit
- `1` = Do not synchronize asynchronous clock input
- `0` = Synchronize asynchronous clock input with system clock (FOSC)

bit 1 **Unimplemented:** Read as ‘`0`’

bit 0 **TMR1ON:** Timer1 On bit
- `1` = Enables Timer1
- `0` = Stops Timer1 and clears Timer1 gate flip-flop

REGISTER 20-2: T1GCON: TIMER1 GATE CONTROL REGISTER

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/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>	
bit 7							bit 0

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

bit 7 **TMR1GE:** Timer1 Gate Enable bit
If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 counting is controlled by the Timer1 gate function
0 = Timer1 counts regardless of Timer1 gate function

bit 6 **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
1 = Timer1 gate Single-Pulse mode is enabled and is controlling Timer1 gate
0 = Timer1 gate Single-Pulse mode is disabled

bit 3 **T1GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit
1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge
0 = Timer1 gate single-pulse acquisition has completed 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 1-0 **T1GSS<1:0>:** Timer1 Gate Source Select bits
11 = Comparator 2 optionally synchronized output (C2OUT_sync)
10 = Comparator 1 optionally synchronized output (C1OUT_sync)
01 = Timer0 overflow output (T0_overflow)
00 = Timer1 gate pin (T1G)

TABLE 20-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

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	122
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	71
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	92
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								185*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count								185*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON	188
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		189

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'.
2: PIC16(L)F1575 only.

21.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16, and 1:64)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2

See Figure 21-1 for a block diagram of Timer2.

FIGURE 21-1: TIMER2 BLOCK DIAGRAM

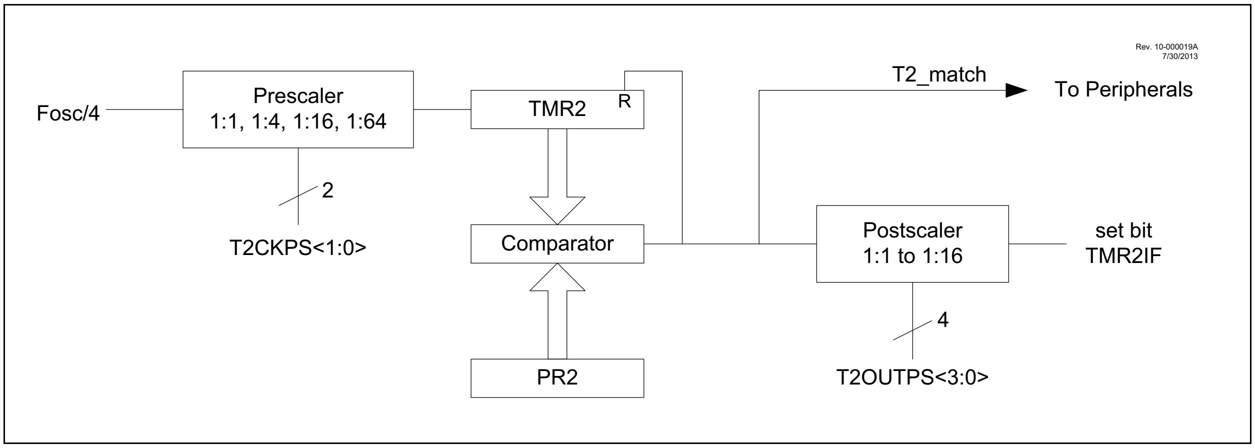
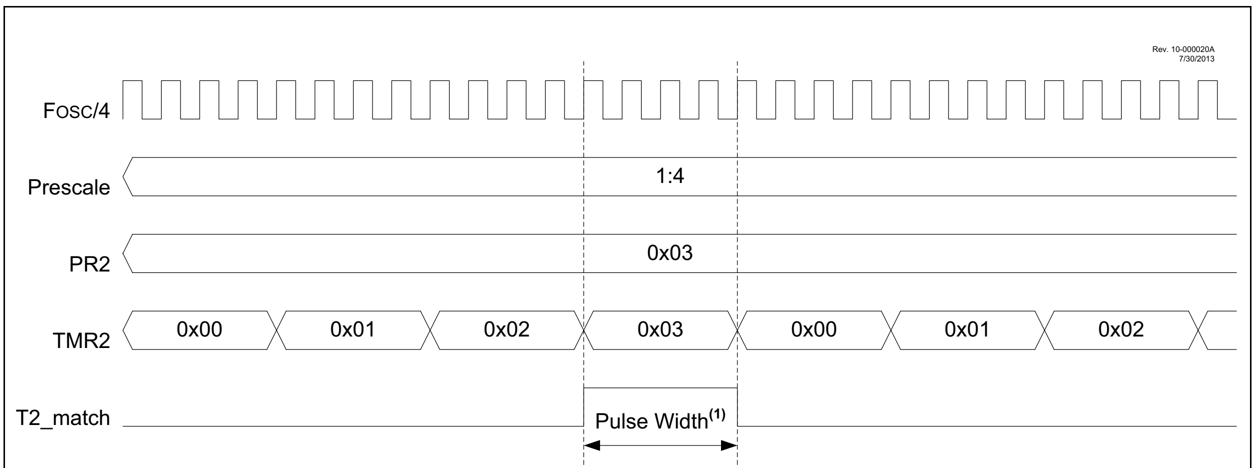


FIGURE 21-2: TIMER2 TIMING DIAGRAM



Note 1: The Pulse Width of T2_match is equal to the scaled input of TMR2.

21.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4).

TMR2 increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 21.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- MCLR Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note:	TMR2 is not cleared when T2CON is written.

21.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (T2_match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE of the PIE1 register.

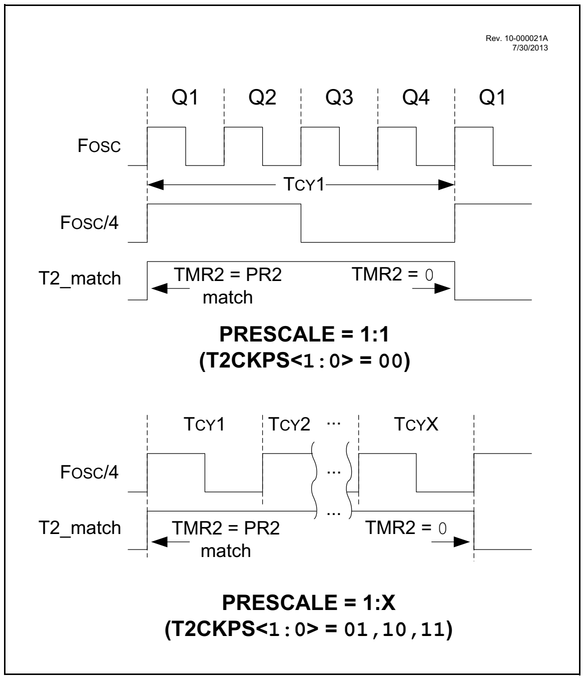
A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS<3:0>, of the T2CON register.

21.3 Timer2 Output

The output of TMR2 is T2_match.

The T2_match signal is synchronous with the system clock. Figure 21-3 shows two examples of the timing of the T2_match signal relative to FOSC and prescale value, T2CKPS<1:0>. The upper diagram illustrates 1:1 prescale timing and the lower diagram, 1:X prescale timing.

FIGURE 21-3: T2_MATCH TIMING DIAGRAM



21.4 Timer2 Operation During Sleep

Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

21.5 Register Definitions: Timer2 Control

REGISTER 21-1: T2CON: TIMER2 CONTROL REGISTER

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
—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscaler Select bits
- 0000 = 1:1 Postscaler
- 0001 = 1:2 Postscaler
- 0010 = 1:3 Postscaler
- 0011 = 1:4 Postscaler
- 0100 = 1:5 Postscaler
- 0101 = 1:6 Postscaler
- 0110 = 1:7 Postscaler
- 0111 = 1:8 Postscaler
- 1000 = 1:9 Postscaler
- 1001 = 1:10 Postscaler
- 1010 = 1:11 Postscaler
- 1011 = 1:12 Postscaler
- 1100 = 1:13 Postscaler
- 1101 = 1:14 Postscaler
- 1110 = 1:15 Postscaler
- 1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit
- 1 = Timer2 is on
- 0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
- 00 = Prescaler is 1
- 01 = Prescaler is 4
- 10 = Prescaler is 16
- 11 = Prescaler is 64

TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

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	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
PR2	Timer2 Module Period Register								191*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		193
TMR2	Holding Register for the 8-bit TMR2 Count								191*

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for Timer2 module.
* Page provides register information.

Note 1: PIC16(L)F1575 only.

22.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

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

The EUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock 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 22-1 and Figure 22-2.

The EUSART transmit output (TX_out) is available to the TX/CK pin and internally to the following peripherals:

- Configurable Logic Cell (CLC)

FIGURE 22-1: EUSART TRANSMIT BLOCK DIAGRAM

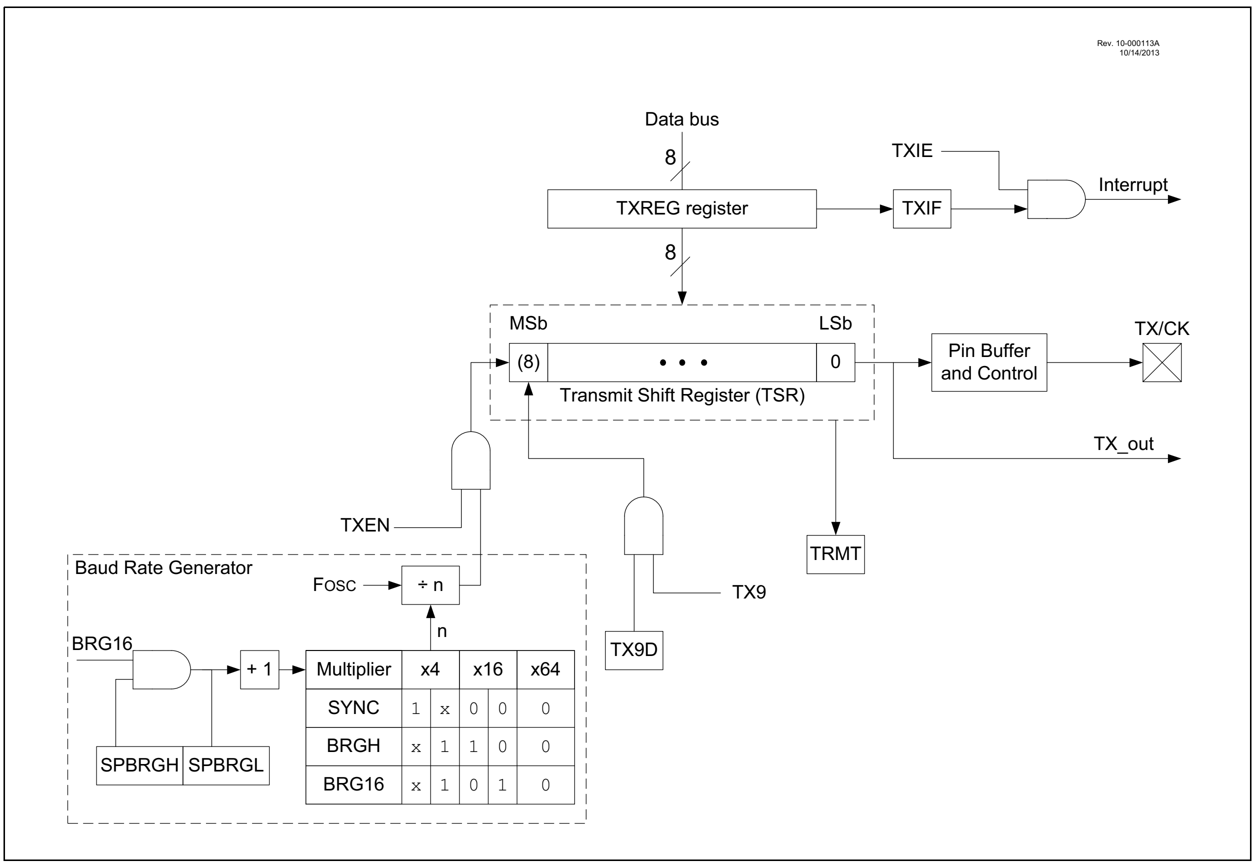
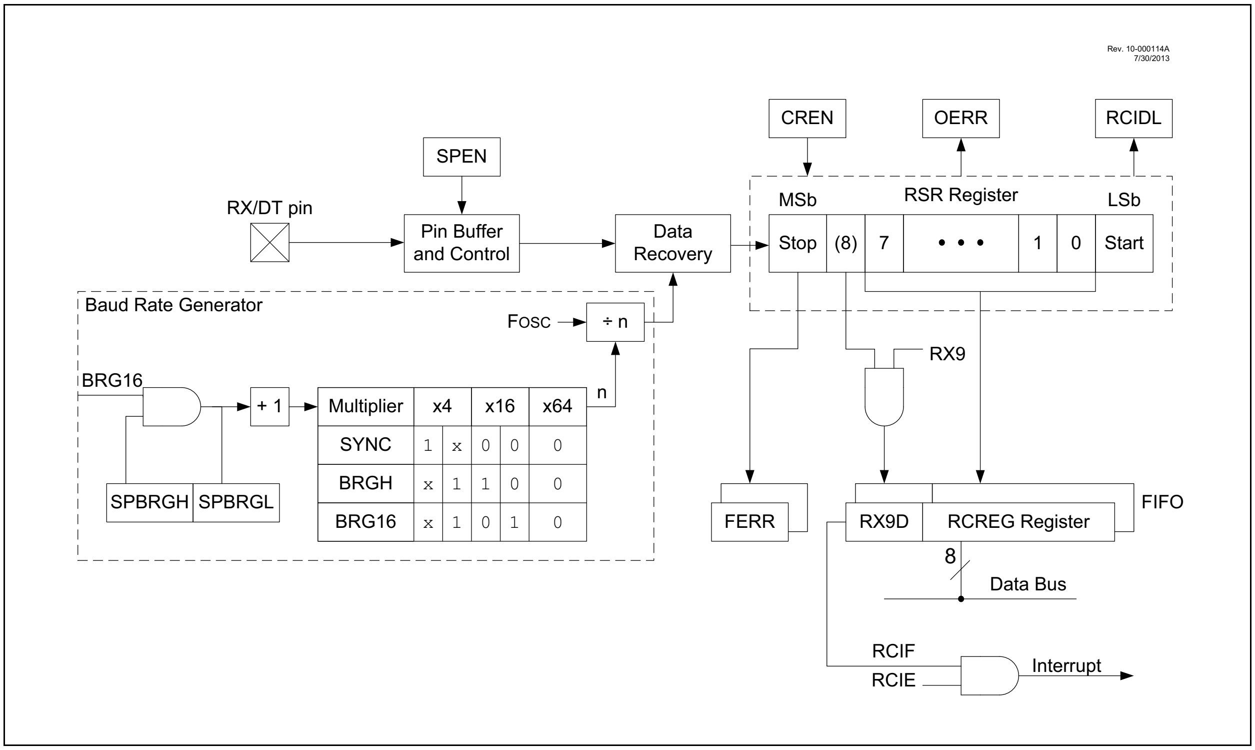


FIGURE 22-2: EUSART RECEIVE BLOCK DIAGRAM



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

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

These registers are detailed in Register 22-1, Register 22-2 and Register 22-3, respectively.

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

22.1 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/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 22-5 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.

22.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 22-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

22.1.1.1 Enabling the Transmitter

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

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

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

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART 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.

22.1.1.2 Transmitting Data

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

22.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 22.5.1.2 "Clock Polarity"**.

22.1.1.4 Transmit Interrupt Flag

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

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

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

22.1.1.5 TSR Status

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

22.1.1.6 Transmitting 9-Bit Characters

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

A special 9-bit Address mode is available for use with multiple receivers. See **Section 22.1.2.7 “Address Detection”** for more information on the address mode.

22.1.1.7 Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 22.4 “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.
6. 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 TXREG register. This will start the transmission.

FIGURE 22-3: ASYNCHRONOUS TRANSMISSION

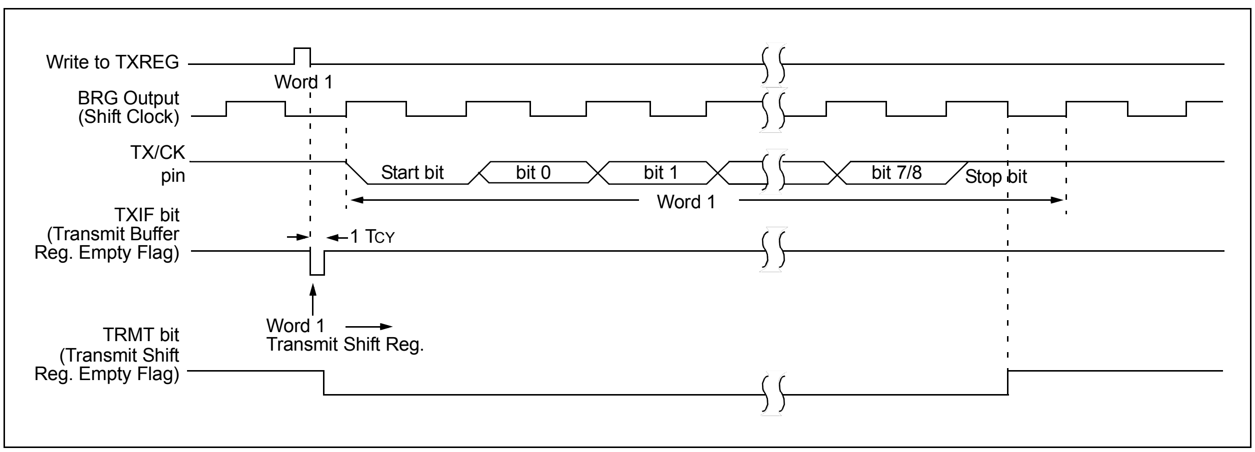
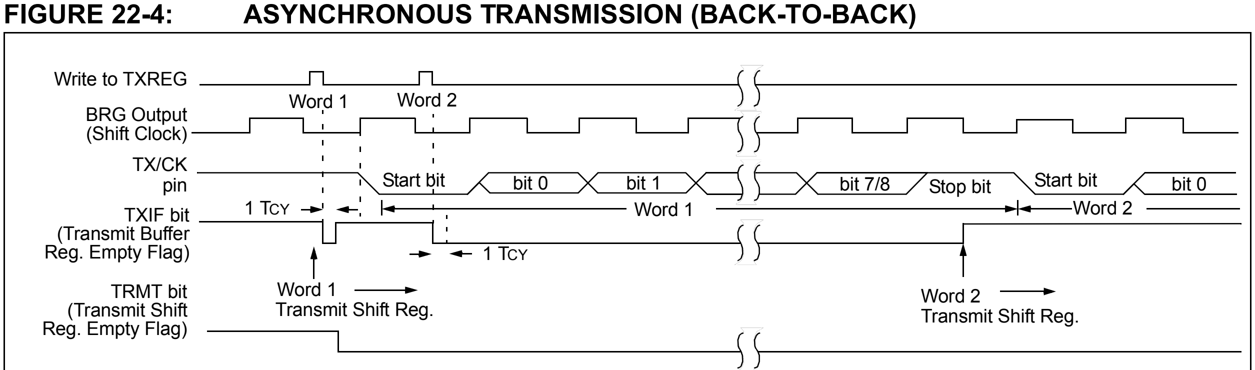


FIGURE 22-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 22-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205*
SPBRGL	BRG<7:0>								207*
SPBRGH	BRG<15:8>								207*
TXREG	EUSART Transmit Data Register								196
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for asynchronous transmission.
* Page provides register information.

22.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 22-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

22.1.2.1 Enabling the Receiver

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

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

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

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

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

22.1.2.2 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 22.1.2.4 "Receive Framing Error"** for more information on framing errors.

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

> **Note:** If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 22.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

22.1.2.3 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.

22.1.2.4 Receive Framing Error

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

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

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

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

22.1.2.5 Receive Overrun Error

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

22.1.2.6 Receiving 9-bit Characters

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

22.1.2.7 Address Detection

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

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

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

22.1.2.8 Asynchronous Reception Set-up:

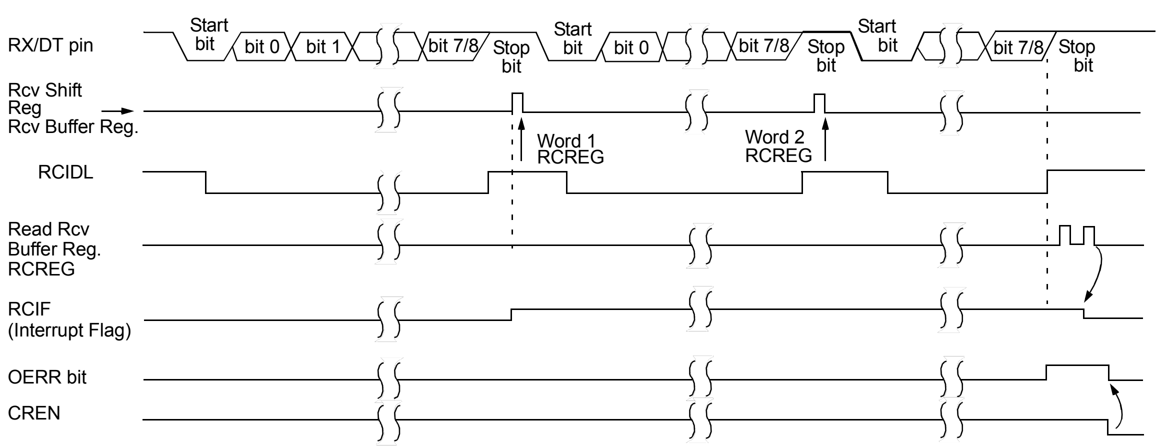
1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 22.4 "EUSART Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for 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 RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

22.1.2.9 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:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 22.4 "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. Enable 9-bit reception by setting the RX9 bit.
6. Enable address detection by setting the ADDEN bit.
7. Enable reception by setting the CREN bit.
8. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
9. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
10. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 22-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

TABLE 22-2: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
RCREG	EUSART Receive Data Register								199*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205*
SPBRGL	BRG<7:0>								207*
SPBRGH	BRG<15:8>								207*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for asynchronous reception.
* Page provides register information.

22.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as V_{DD} or temperature changes, and this directly affects the asynchronous baud rate.

The Auto-Baud Detect feature (see **Section 22.4.1 "Auto-Baud Detect"**) can be used to compensate for changes in the INTOSC frequency.

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

22.3 Register Definitions: EUSART Control

REGISTER 22-1: TXSTA: 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 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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don’t care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
Synchronous mode:
Don’t care

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

REGISTER 22-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

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 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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don’t care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
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 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (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
Asynchronous mode 8-bit (RX9 = 0):
Don’t care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

REGISTER 22-3: BAUDCON: BAUD RATE CONTROL REGISTER

R/W-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 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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don’t care

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don’t care

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **SCKP**: Synchronous Clock Polarity Select bit
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
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge. No character will be received, RCIF bit will be set. WUE will automatically clear after RCIF is set.
0 = Receiver is operating normally
Synchronous mode:
Don’t care

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don’t care

22.4 EUSART Baud Rate Generator (BRG)

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

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

Table 22-3 contains the formulas for determining the baud rate. Example 22-1 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 22-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is idle before changing the system clock.

EXAMPLE 22-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$Desired\ Baud\ Rate = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SPBRGH:SPBRGL:

$$X = \frac{\frac{F_{OSC}}{Desired\ Baud\ Rate}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$Calculated\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$Error = \frac{Calc.\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

TABLE 22-3: BAUD RATE FORMULAS

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

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

TABLE 22-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205
SPBRGL	BRG<7:0>								207*
SPBRGH	BRG<15:8>								207*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

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

TABLE 22-5: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	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	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	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	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5

TABLE 22-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	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	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	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

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	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	—	—	—

TABLE 22-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 16.000 MHz			FOSC = 11.0592 MHz		
	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

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	FOSC = 8.000 MHz			FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 1.000 MHz		
	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	—	—	—

22.4.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCON register starts the auto-baud calibration sequence (Figure 22-6). 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 SPBRG begins counting up using the BRG counter clock as shown in Table 22-6. 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 SPBRGH, SPBRGL register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRGL register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 22-6. During ABD, both the SPBRGH and SPBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH and SPBRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

Note 1: If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see **Section 22.4.3 "Auto-Wake-up on Break"**).

2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.

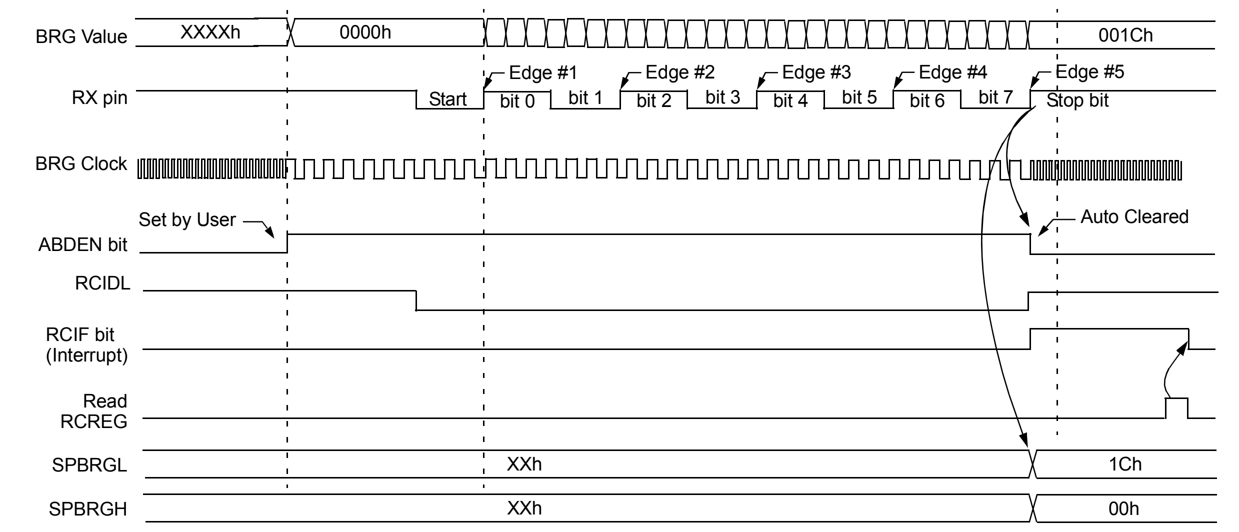
3: During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRGL register pair.

TABLE 22-6: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	FOSC/64	FOSC/512
0	1	FOSC/16	FOSC/128
1	0	FOSC/16	FOSC/128
1	1	FOSC/4	FOSC/32

Note: During the ABD sequence, SPBRGL and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

FIGURE 22-6: AUTOMATIC BAUD RATE CALIBRATION



Note 1: The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode.

22.4.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRGL register pair. 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 RDICL bit will set. If the RCREG is read after the overflow occurs, but before the fifth rising edge, 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.

22.4.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 22-7), and asynchronously if the device is in Sleep mode (Figure 22-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

22.4.3.1 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

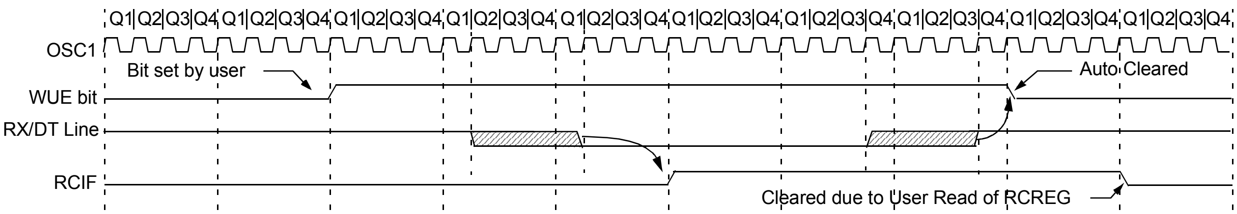
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

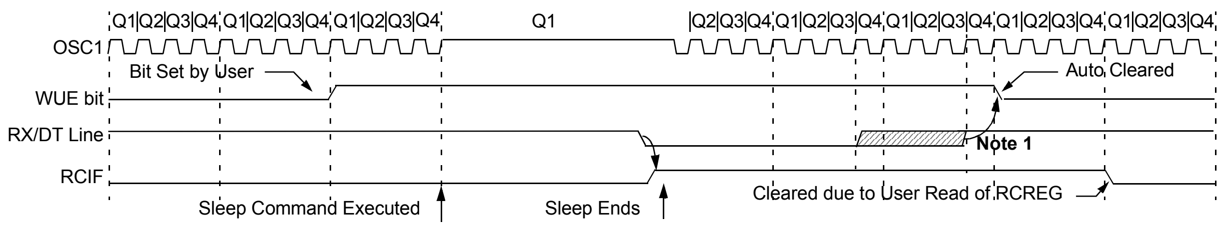
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 22-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION



Note 1: The **EUSART** remains in Idle while the WUE bit is set.

FIGURE 22-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



Note 1: If the wake-up event requires long oscillator warm-up time, the automatic clearing of the WUE bit can occur while the `stposc` 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.

22.4.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 22-9 for the timing of the Break character sequence.

22.4.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

22.4.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

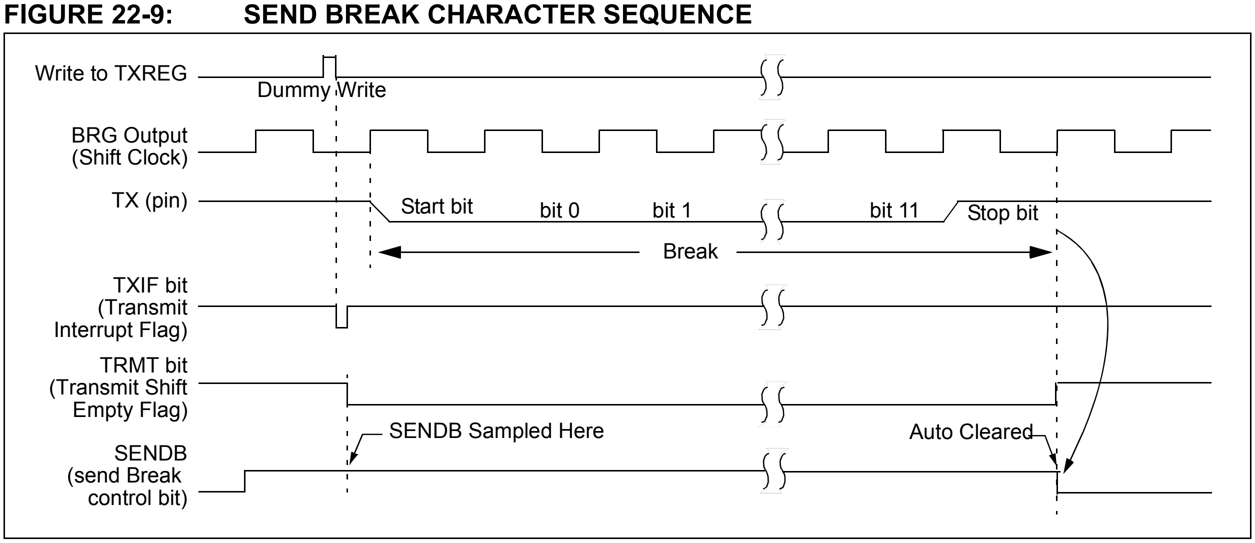
A Break character has been received when;

- RCIF bit is set
- FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 22.4.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCON register before placing the EUSART in Sleep mode.

FIGURE 22-9: SEND BREAK CHARACTER SEQUENCE



22.5 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.

22.5.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for synchronous master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART.

22.5.1.1 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.

22.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. 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.

22.5.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

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

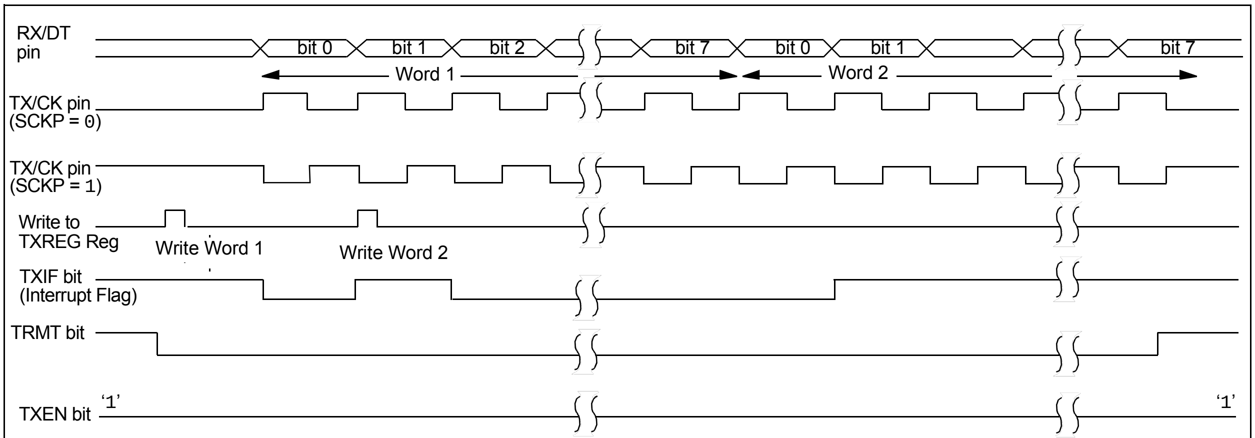
Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

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

22.5.1.4 Synchronous Master Transmission Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 22.4 “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.
6. 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 TXREG register.

FIGURE 22-10: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SPBRGL = 0, continuous transmission of two 8-bit words.

FIGURE 22-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

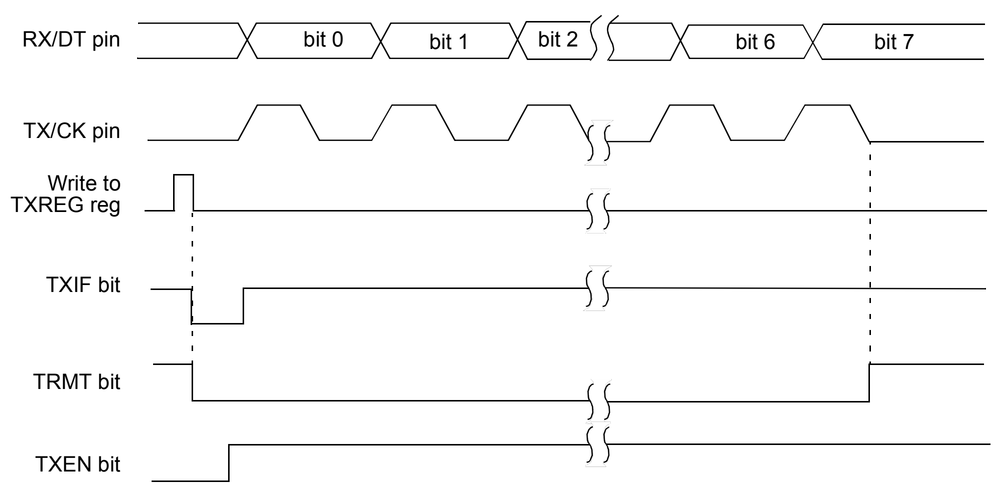


TABLE 22-7: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205
SPBRGL	BRG<7:0>								207*
SPBRGH	BRG<15:8>								207*
TXREG	EUSART Transmit Data Register								196*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous master transmission.
* Page provides register information.

22.5.1.5 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 RCSTA register) or the Continuous Receive Enable bit (CREN of the RCSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREG. The RCIF bit remains set as long as there are 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.

22.5.1.6 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.

22.5.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREG is read to access the FIFO. When this happens the OERR bit of the RCSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

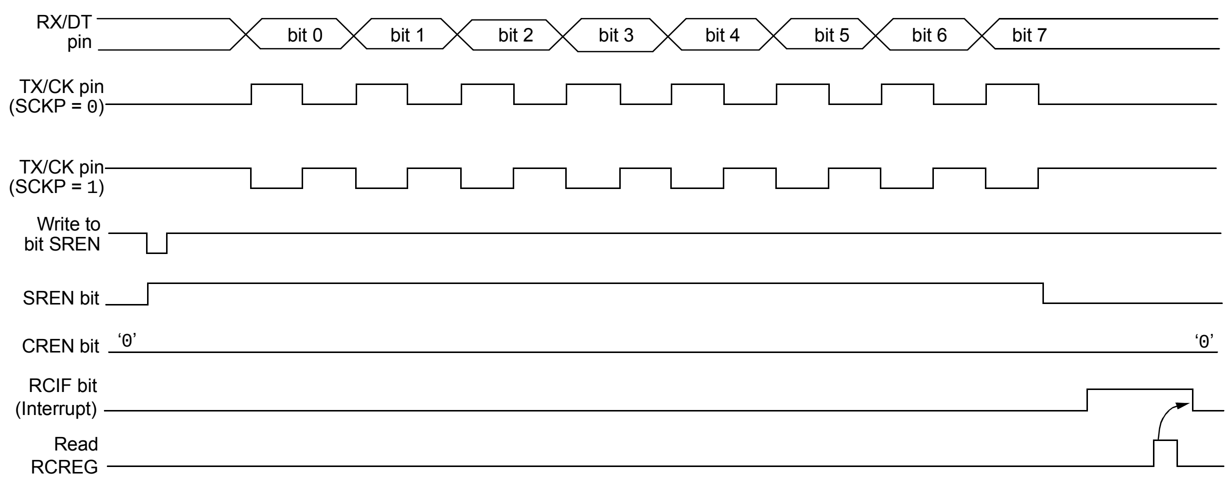
22.5.1.8 Receiving 9-bit Characters

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

22.5.1.9 Synchronous Master Reception Set-up:

1. Initialize the SPBRGH, SPBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. 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 RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

FIGURE 22-12: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRGH = 0.

TABLE 22-8: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
RCREG	EUSART Receive Data Register								199*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205
SPBRGL	BRG<7:0>								207*
SPBRGH	BRG<15:8>								207*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous master reception.
* Page provides register information.

22.5.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Clearing the CSRC bit of the TXSTA register configures the device as a slave. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART.

22.5.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 22.5.1.3 "Synchronous Master Transmission"**), except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in the TXREG register.
3. The TXIF bit will not be set.
4. After the first character has been shifted out of TSR, the TXREG register will transfer the second character to the TSR and the TXIF bit will now be set.
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.

22.5.2.2 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 TXREG register.

TABLE 22-9: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205
TXREG	EUSART Transmit Data Register								196*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave transmission.
* Page provides register information.

22.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 22.5.1.5 "Synchronous Master Reception"**), with the following exceptions:

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

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

22.5.2.4 Synchronous Slave Reception Set-up:

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.
6. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

TABLE 22-10: 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
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	206
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
RCREG	EUSART Receive Data Register								199*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	205
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	204

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave reception.
* Page provides register information.

23.0 16-BIT PULSE-WIDTH MODULATION (PWM) MODULE

The Pulse-Width Modulation (PWM) module generates a pulse width modulated signal determined by the phase, duty cycle, period, and offset event counts that are contained in the following registers:

- PWMxPH register
- PWMxDC register
- PWMxPR register
- PWMxOF register

Figure 23-1 shows a simplified block diagram of the PWM operation.

Each PWM module has four modes of operation:

- Standard
- Set On Match
- Toggle On Match
- Center Aligned

For a more detailed description of each PWM mode, refer to **Section 23.2 "PWM Modes"**.

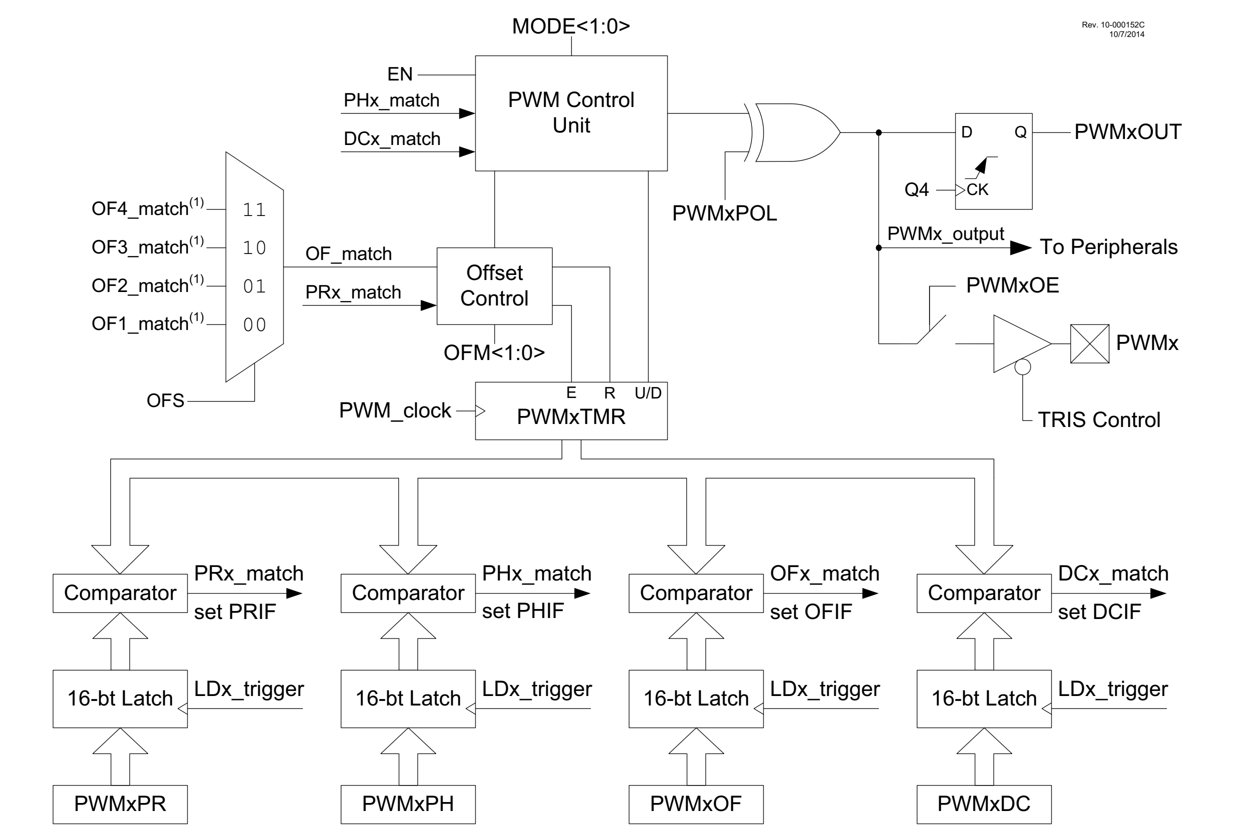
Each PWM module has four offset modes:

- Independent Run
- Slave Run with Synchronous Start
- One-Shot Slave with Synchronous Start
- Continuous Run Slave with Synchronous Start and Timer Reset

Using the offset modes, each PWM module can offset its waveform relative to any other PWM module in the same device. For a more detailed description of the offset modes refer to **Section 23.3 "Offset Modes"**.

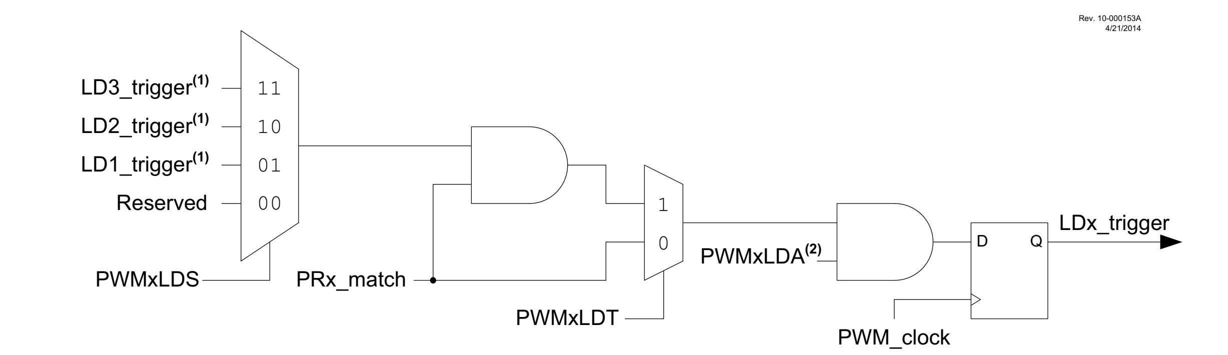
Every PWM module has a configurable reload operation to ensure all event count buffers change at the end of a period thereby avoiding signal glitches. Figure 23-2 shows a simplified block diagram of the reload operation. For a more detailed description of the reload operation, refer to Section **Section 23.4 "Reload Operation"**.

FIGURE 23-1: 16-BIT PWM BLOCK DIAGRAM



Note 1: A PWM module cannot trigger from its own offset match event. The input corresponding to a PWM module's own offset match is reserved.

FIGURE 23-2: LOAD TRIGGER BLOCK DIAGRAM



Note 1. The input corresponding to a PWM module's own load trigger is reserved.
2. PWMxLDA is cleared by hardware upon LDx_trigger.

23.1 Fundamental Operation

The PWM module produces a 16-bit resolution pulse width modulated output.

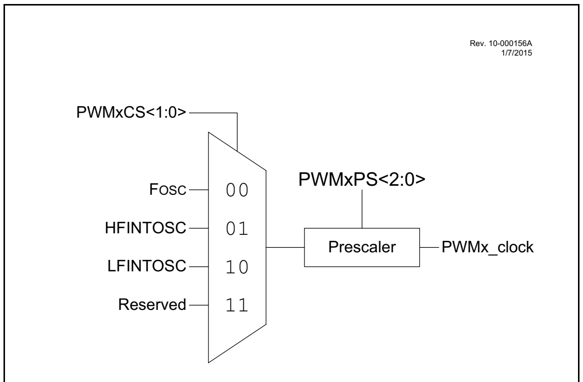
Each PWM module has an independent timer driven by a selection of clock sources determined by the PWMxCLKCON register (Register 23-4). The timer value is compared to event count registers to generate the various events of a the PWM waveform, such as the period and duty cycle. For a block diagram describing the clock sources refer to Figure 23-3.

Each PWM module can be enabled individually using the EN bit of the PWMxCON register, or several PWM modules can be enabled simultaneously using the mirror bits of the PWMEN register.

The current state of the PWM output can be read using the OUT bit of the PWMxCON register. In some modes this bit can be set and cleared by software giving additional software control over the PWM waveform. This bit is synchronized to FOSC/4 and therefore does not change in real time with respect to the PWM_clock.

Note:	If PWM_clock > FOSC/4, the OUT bit may not accurately represent the output state of the PWM.

FIGURE 23-3: PWM CLOCK SOURCE BLOCK DIAGRAM



23.1.1 PWMx PIN CONFIGURATION

All PWM outputs are multiplexed with the PORT data latch, so the pins must also be configured as outputs by clearing the associated PORT TRIS bits.

The slew rate feature may be configured to optimize the rate to be used in conjunction with the PWM outputs. High-speed output switching is attained by clearing the associated PORT SLRCON bits.

The PWM outputs can be configured to be open-drain outputs by setting the associated PORT ODCON bits.

23.1.2 PWMx Output Polarity

The output polarity is inverted by setting the POL bit of the PWMxCON register. The polarity control affects the PWM output even when the module is not enabled.

23.2 PWM Modes

PWM Modes are selected with MODE<1:0> bits of the PWMxCON register (Register 23-1).

In all PWM modes an offset match event can also be used to synchronize the PWMxTMR in three offset modes. See **Section 23.3 "Offset Modes"** for more information.

23.2.1 STANDARD MODE

The Standard mode (MODE = `00`) selects a single phase PWM output. The PWM output in this mode is determined by when the period, duty cycle, and phase counts match the PWMxTMR value. The start of the duty cycle occurs on the phase match and the end of the duty cycle occurs on the duty cycle match. The period match resets the timer. The offset match can also be used to synchronize the PWMxTMR in the offset modes. See **Section 23.3 "Offset Modes"** for more information.

Equation 23-1 is used to calculate the PWM period in Standard mode.

Equation 23-2 is used to calculate the PWM duty-cycle ratio in Standard mode.

EQUATION 23-1: PWM PERIOD IN STANDARD MODE

$$Period = \frac{(PWMxPR + 1) \cdot Prescale}{PWMxCLK}$$

EQUATION 23-2: PWM DUTY CYCLE IN STANDARD MODE

$$Duty\ Cycle = \frac{(PWMxDC - PWMxPH)}{PWMxPR + 1}$$

A detailed timing diagram for Standard mode is shown in Figure 23-4.

23.2.2 SET ON MATCH MODE

The Set On Match mode (MODE = `01`) generates an active output when the phase count matches the PWMxTMR value. The output stays active until the OUT bit of the PWMxCON register is cleared or the PWM module is disabled. The duty cycle count has no effect in this mode. The period count only determines the maximum PWMxTMR value above which no phase matches can occur.

The PWMxOUT bit can be used to set or clear the output of the PWM in this mode. Writes to this bit will take place on the next rising edge of the PWM_clock after the bit is written.

A detailed timing diagram for Set On Match is shown in Figure 23-5.

23.2.3 TOGGLE ON MATCH MODE

The Toggle On Match mode (MODE = `10`) generates a 50% duty cycle PWM with a period twice as long as that computed for the standard PWM mode. Duty cycle count has no effect in this mode. The phase count determines how many PWMxTMR periods after a period event the output will toggle.

Writes to the OUT bit of the PWMxCON register will have no effect in this mode.

A detailed timing diagram for Toggle On Match is shown in Figure 23-6.

23.2.4 CENTER ALIGNED MODE

The Center Aligned mode (MODE = `11`) generates a PWM waveform that is centered in the period. In this mode the period is two times the PWMxPR count. The PWMxTMR counts up to the period value then counts back down to 0. The duty cycle count determines both the start and end of the active PWM output. The start of the duty cycle occurs at the match event when PWMxTMR is incrementing and the duty cycle ends at the match event when PWMxTMR is decrementing. The incrementing match value is the period count minus the duty cycle count. The decrementing match value is the incrementing match value plus 1.

Equation 23-3 is used to calculate the PWM period in Center Aligned mode.

EQUATION 23-3: PWM PERIOD IN CENTER ALIGNED MODE

$$Period = \frac{(PWMxPR + 1) \cdot Prescale \cdot 2}{PWMxCLK}$$

Equation 23-4 is used to calculate the PWM duty cycle ratio in Center Aligned mode

EQUATION 23-4: PWM DUTY CYCLE IN CENTER ALIGNED MODE

$$Duty\ Cycle = \frac{PWMxDC \cdot 2}{(PWMxPR + 1) \cdot 2}$$

Writes to PWMxOUT will have no effect in this mode.

A detailed timing diagram for Center Aligned mode is shown in Figure 23-7.

FIGURE 23-4: STANDARD PWM MODE TIMING DIAGRAM

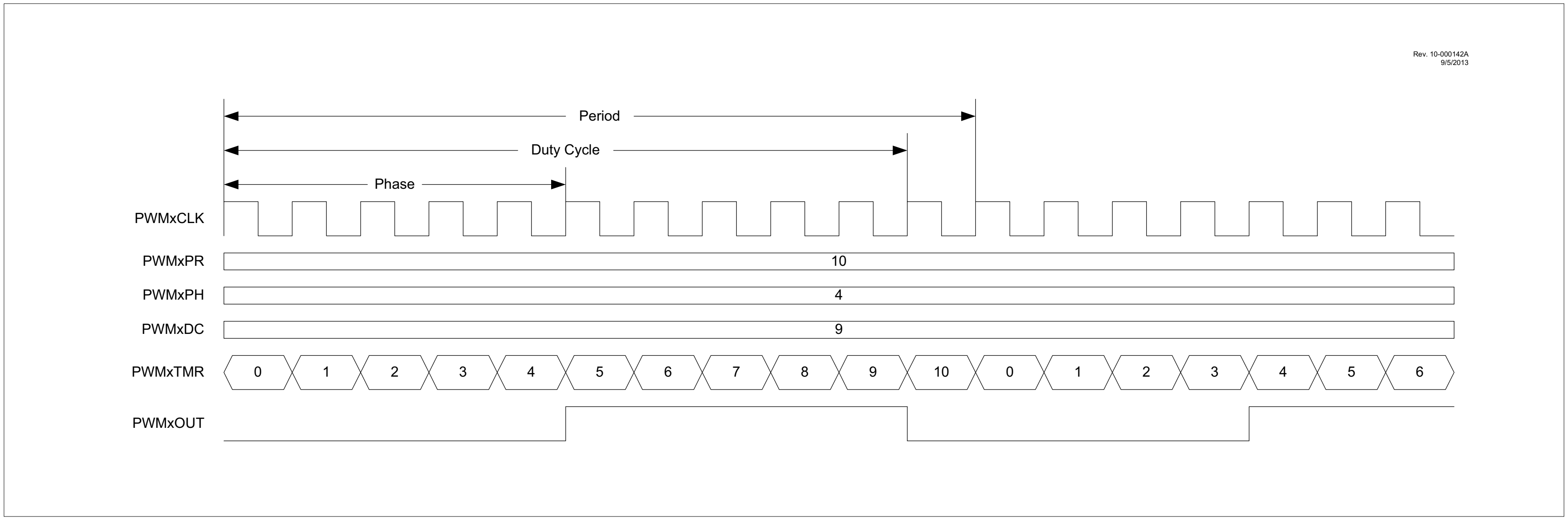


FIGURE 23-5: SET ON MATCH PWM MODE TIMING DIAGRAM

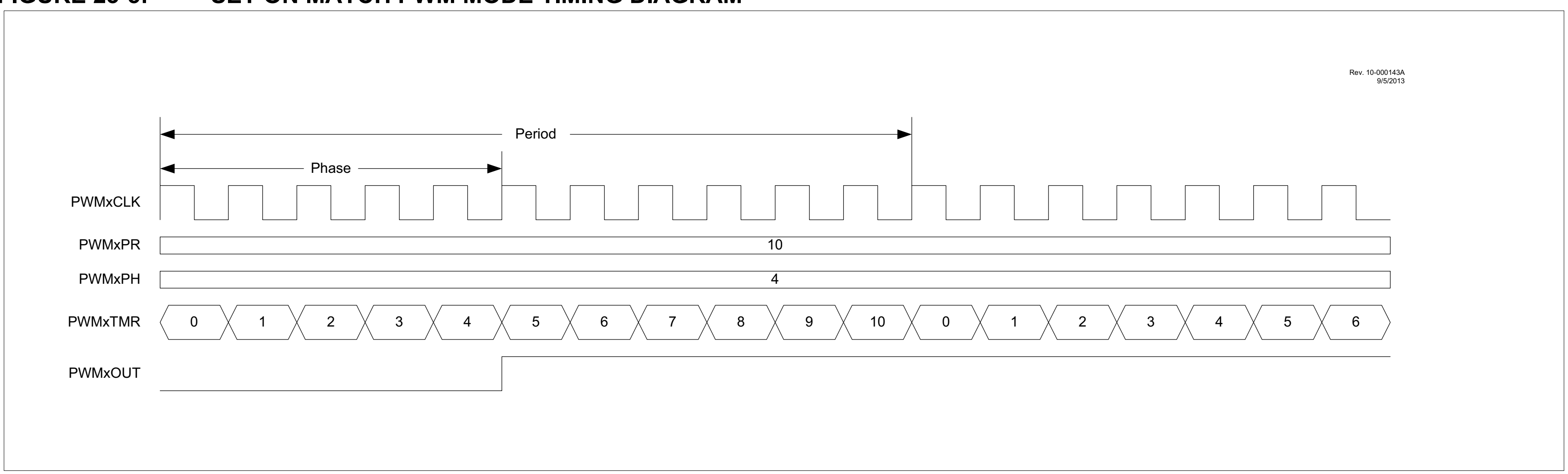


FIGURE 23-6: TOGGLE-ON MATCH PWM MODE TIMING DIAGRAM

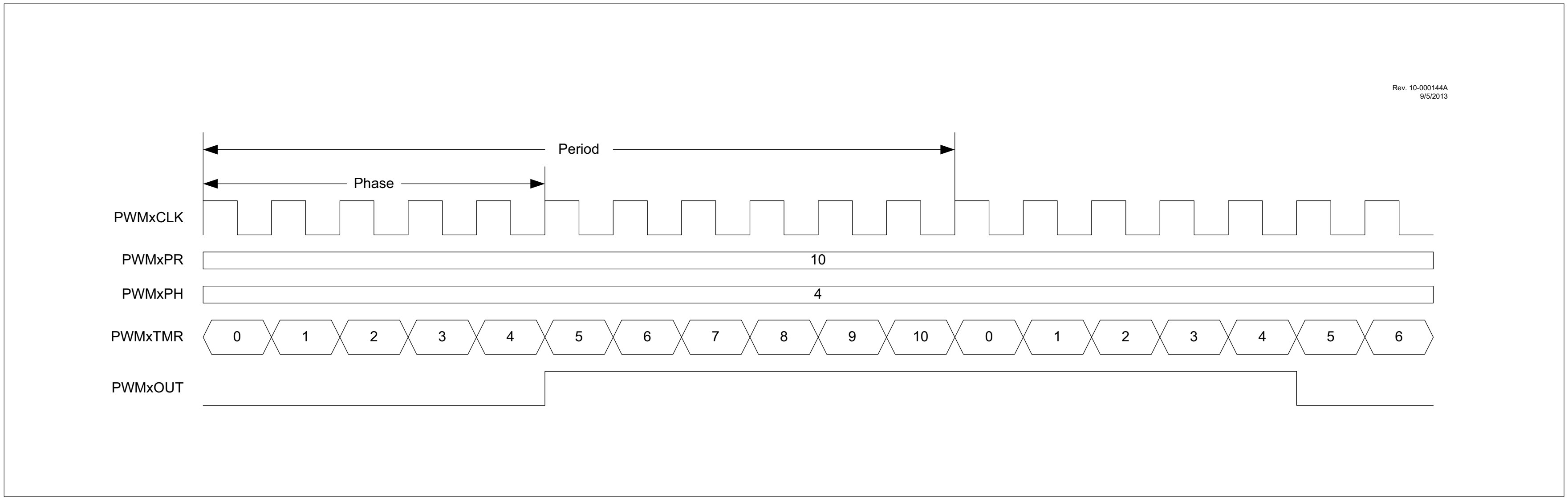
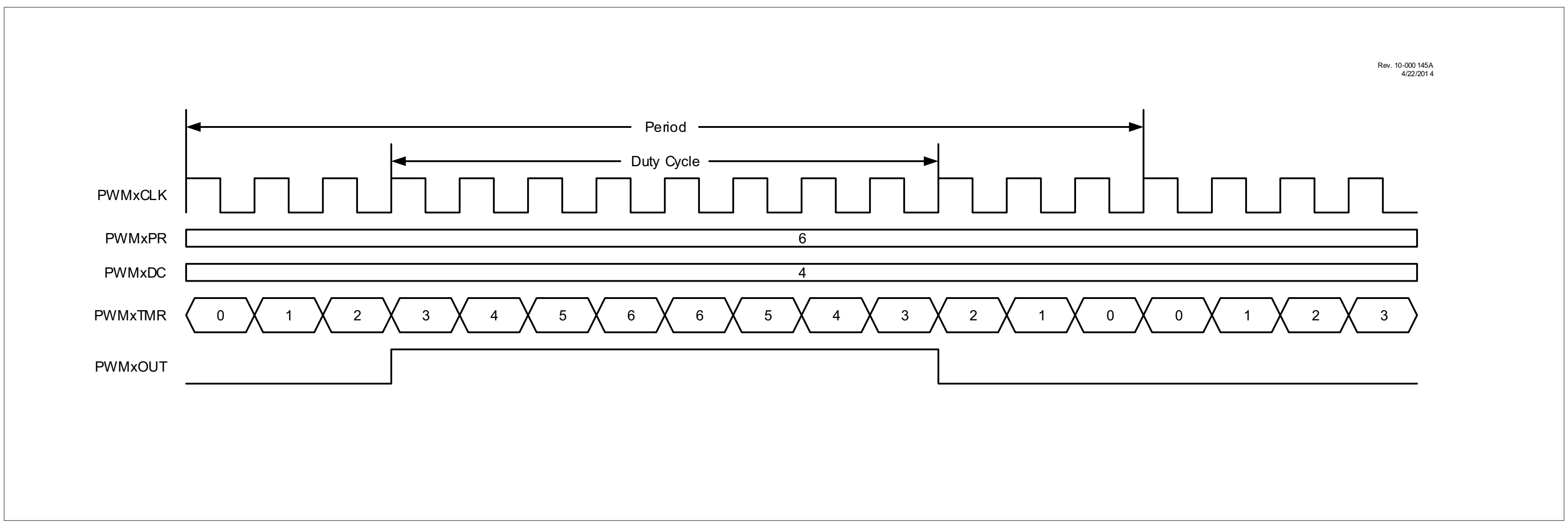


FIGURE 23-7: CENTER ALIGNED PWM MODE TIMING DIAGRAM



23.3 Offset Modes

The Offset modes provide the means to adjust the waveform of a slave PWM module relative to the waveform of a master PWM module in the same device.

23.3.1 INDEPENDENT RUN MODE

In Independent Run mode (OFM = `00`), the PWM module is unaffected by the other PWM modules in the device. The PWMxTMR associated with the PWM module in this mode starts counting as soon as the EN bit associated with this PWM module is set and continues counting until the EN bit is cleared. Period events reset the PWMxTMR to zero after which the timer continues to count.

A detailed timing diagram of this mode used with Standard PWM mode is shown in Figure 23-8.

23.3.2 SLAVE RUN MODE WITH SYNC START

In Slave Run mode with Sync Start (OFM = `01`), the slave PWMxTMR waits for the master's OF_match event. When this event occurs, if the EN bit is set, the PWMxTMR begins counting and continues to count until software clears the EN bit. Slave period events reset the PWMxTMR to zero after which the timer continues to count.

A detailed timing diagram of this mode used with Standard PWM mode is shown in Figure 23-9.

23.3.3 ONE-SHOT SLAVE MODE WITH SYNC START

In One-Shot Slave mode with Synchronous Start (OFM = `10`), the slave PWMxTMR waits until the master's OF_match event. The timer then begins counting, starting from the value that is already in the timer, and continues to count until the period match event. When the period event occurs the timer resets to zero and stops counting. The timer then waits until the next master OF_match event after which it begins counting again to repeat the cycle.

A detailed timing diagram of this mode used with Standard PWM mode is shown in Figure 23-10.

23.3.4 CONTINUOUS RUN SLAVE MODE WITH SYNC START AND TIMER RESET

In Continuous Run Slave mode with Synchronous Start and Timer Reset (OFM = `11`) the slave PWMxTMR is inhibited from counting after the slave PWM enable is set. The first master OF_match event starts the slave PWMxTMR. Subsequent master OF_match events reset the slave PWMxTMR timer value back to 1 after which the slave PWMxTMR continues to count. The next master OF_match event resets the slave PWMxTMR back to 1 to repeat the cycle. Slave period events that occur before the master's OF_match event will reset the slave PWMxTMR to zero after which the timer will continue to count. Slaves operating in this mode must have a PWMxPH register pair value equal to or greater than 1, otherwise, the phase match event will not occur precluding the start of the PWM output duty cycle.

The offset timing will persist If both the master and slave PWMxPR values are the same and the Slave Offset mode is changed to Independent Run mode while the PWM module is operating.

A detailed timing diagram of this mode used in Standard PWM mode is shown in Figure 23-11.

Note:	Unexpected results will occur if the slave PWM_clock is a higher frequency than the master PWM_clock.

23.3.5 OFFSET MATCH IN CENTER ALIGNED MODE

When a master is operating in Center Aligned mode the offset match event depends on which direction the PWMxTMR is counting. Clearing the OFO bit of the PWMxOFCON register will cause the OF_match event to occur when the timer is counting up. Setting the OFO bit of the PWMxOFCON register will cause the OF_match event to occur when the timer is counting down. The OFO bit is ignored in non-center aligned modes.

The OFO bit is double buffered and requires setting the LDA bit to take effect when the PWM module is operating.

Detailed timing diagrams of Center Aligned mode using offset match control in Independent Slave with Sync Start mode can be seen in Figure 23-12 and Figure 23-13.

FIGURE 23-8: INDEPENDENT RUN MODE TIMING DIAGRAM

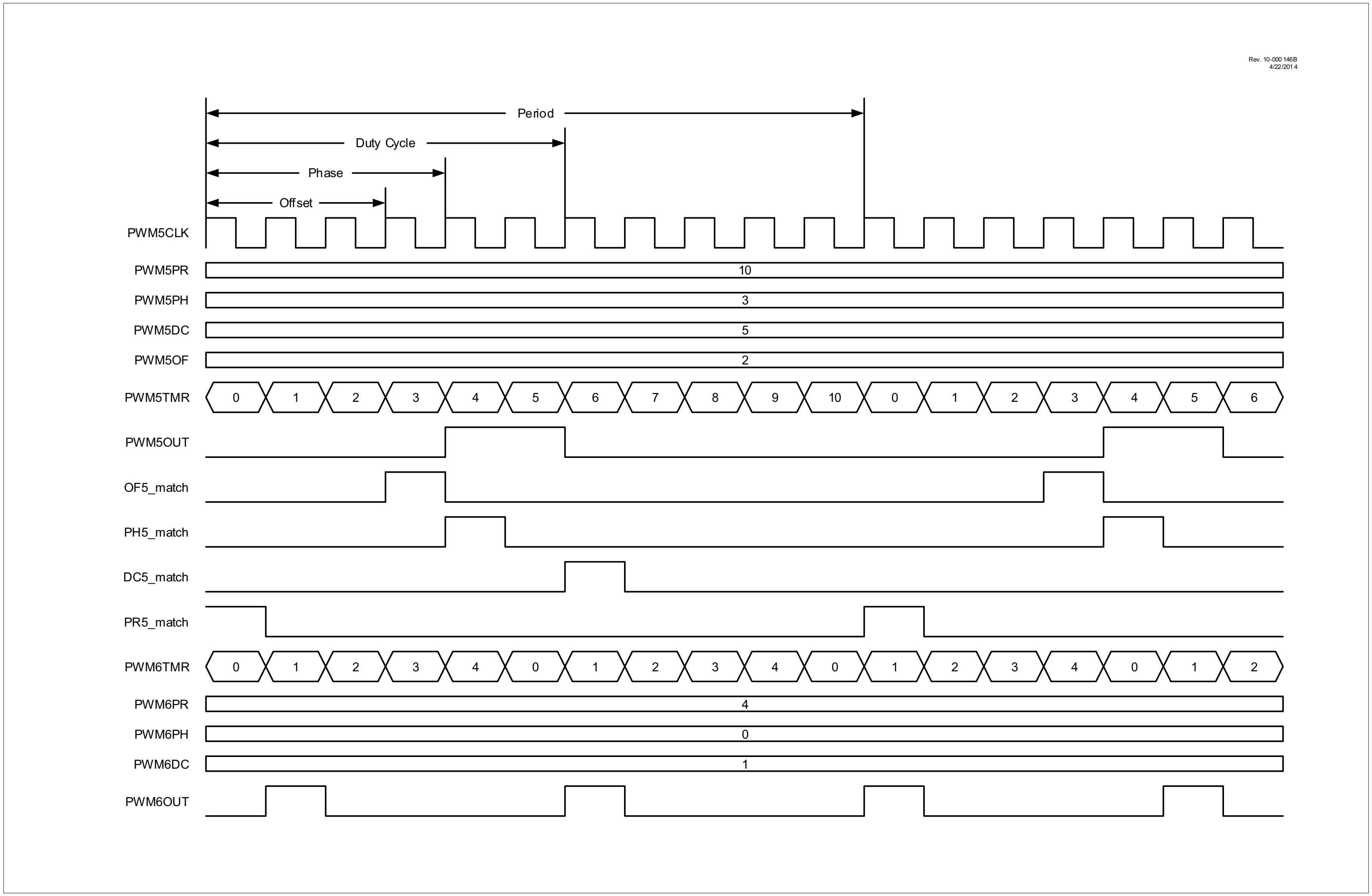


FIGURE 23-9: SLAVE RUN MODE WITH SYNC START TIMING DIAGRAM

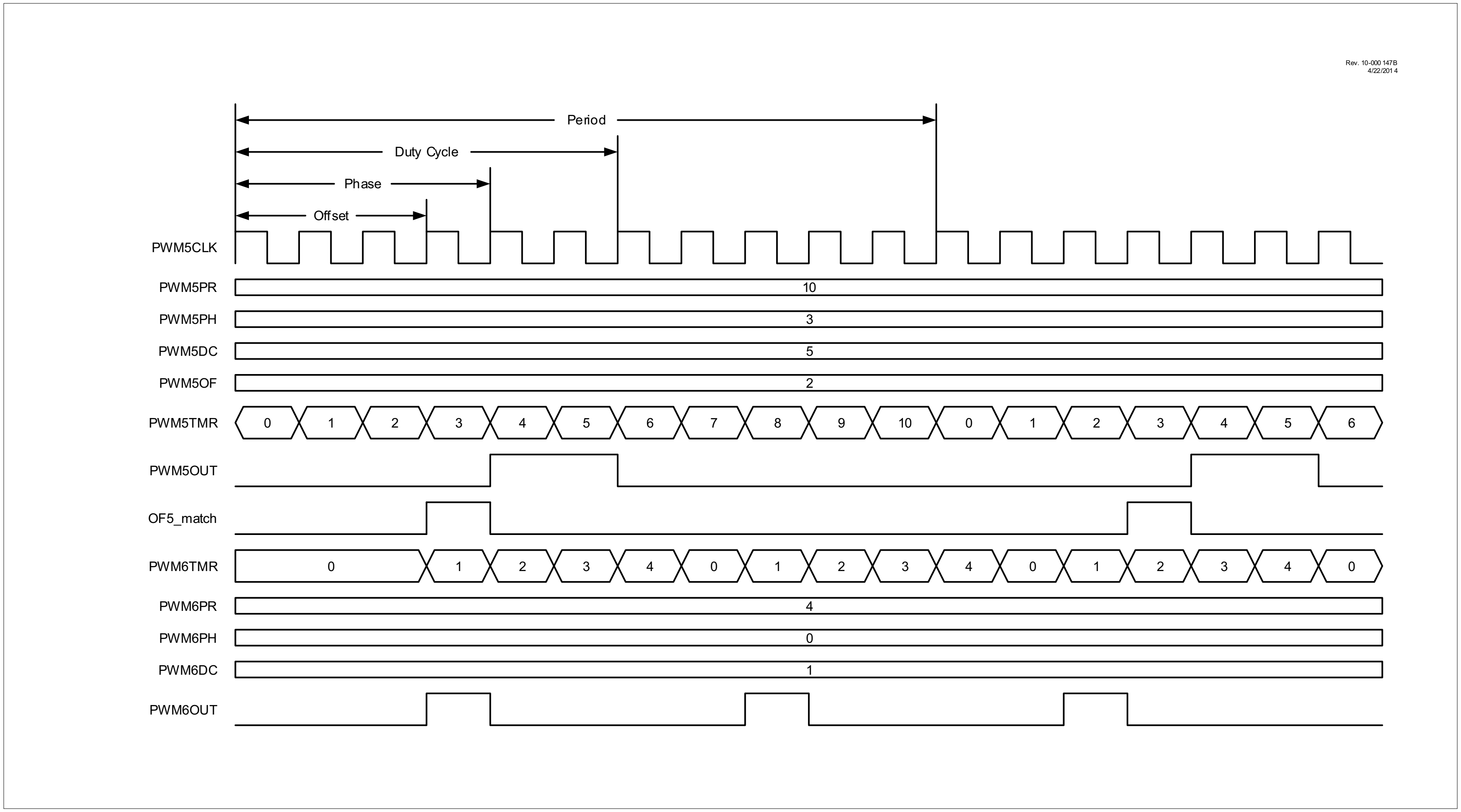


FIGURE 23-10: ONE-SHOT SLAVE RUN MODE WITH SYNC START TIMING DIAGRAM

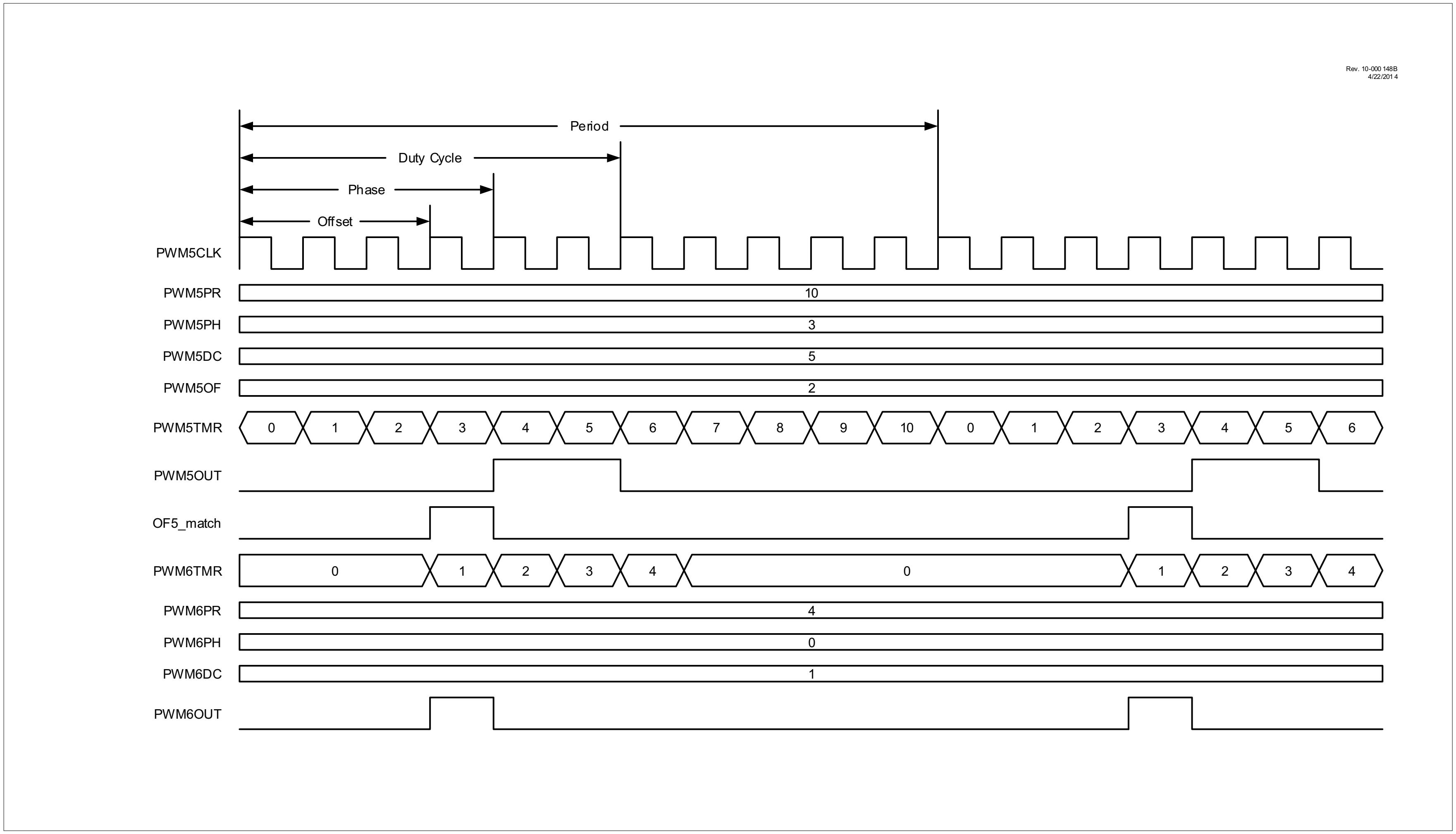


FIGURE 23-11: CONTINUOUS SLAVE RUN MODE WITH IMMEDIATE RESET AND SYNC START TIMING DIAGRAM

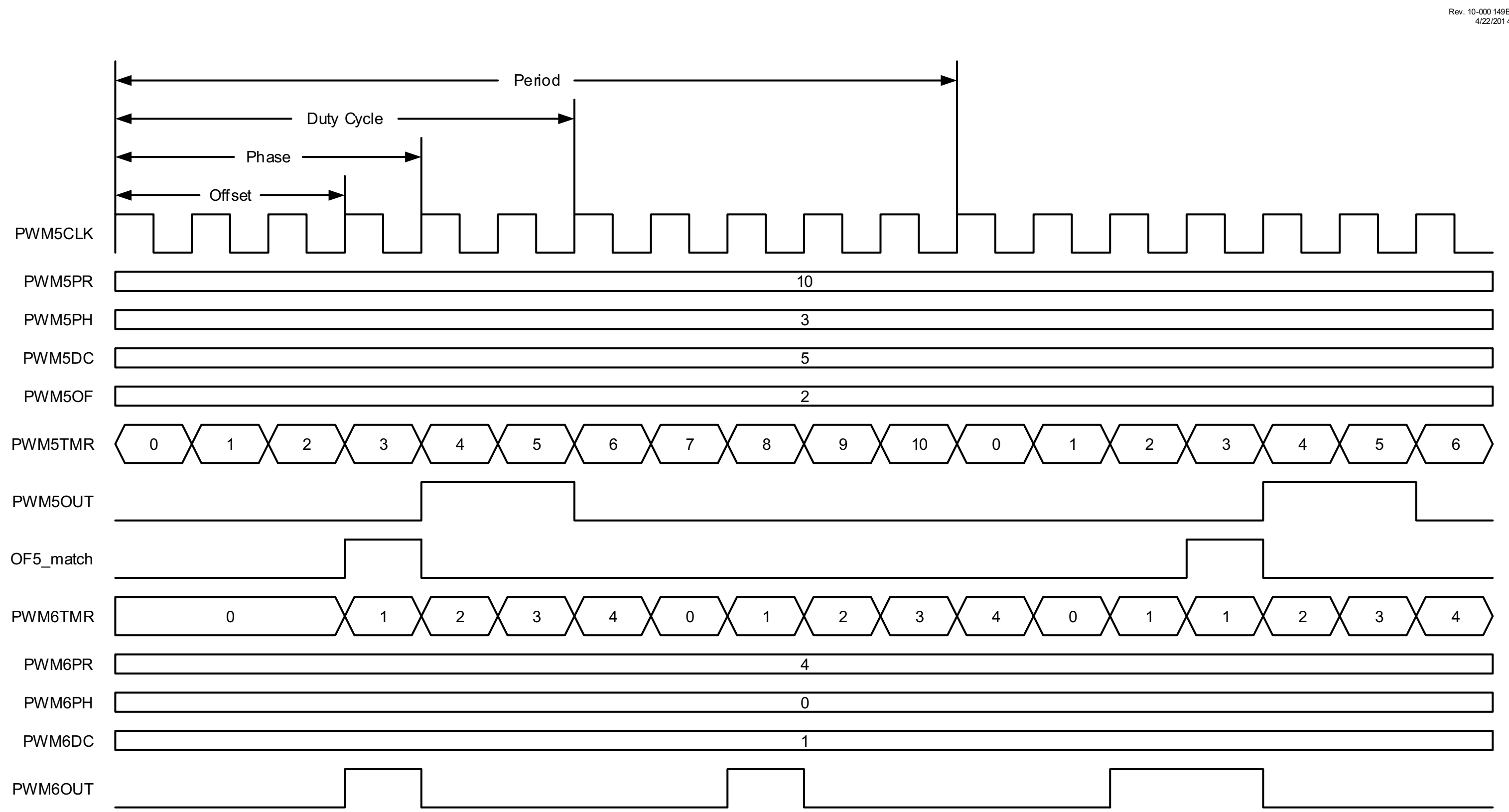


FIGURE 23-12: OFFSET MATCH ON INCREMENTING TIMER TIMING DIAGRAM

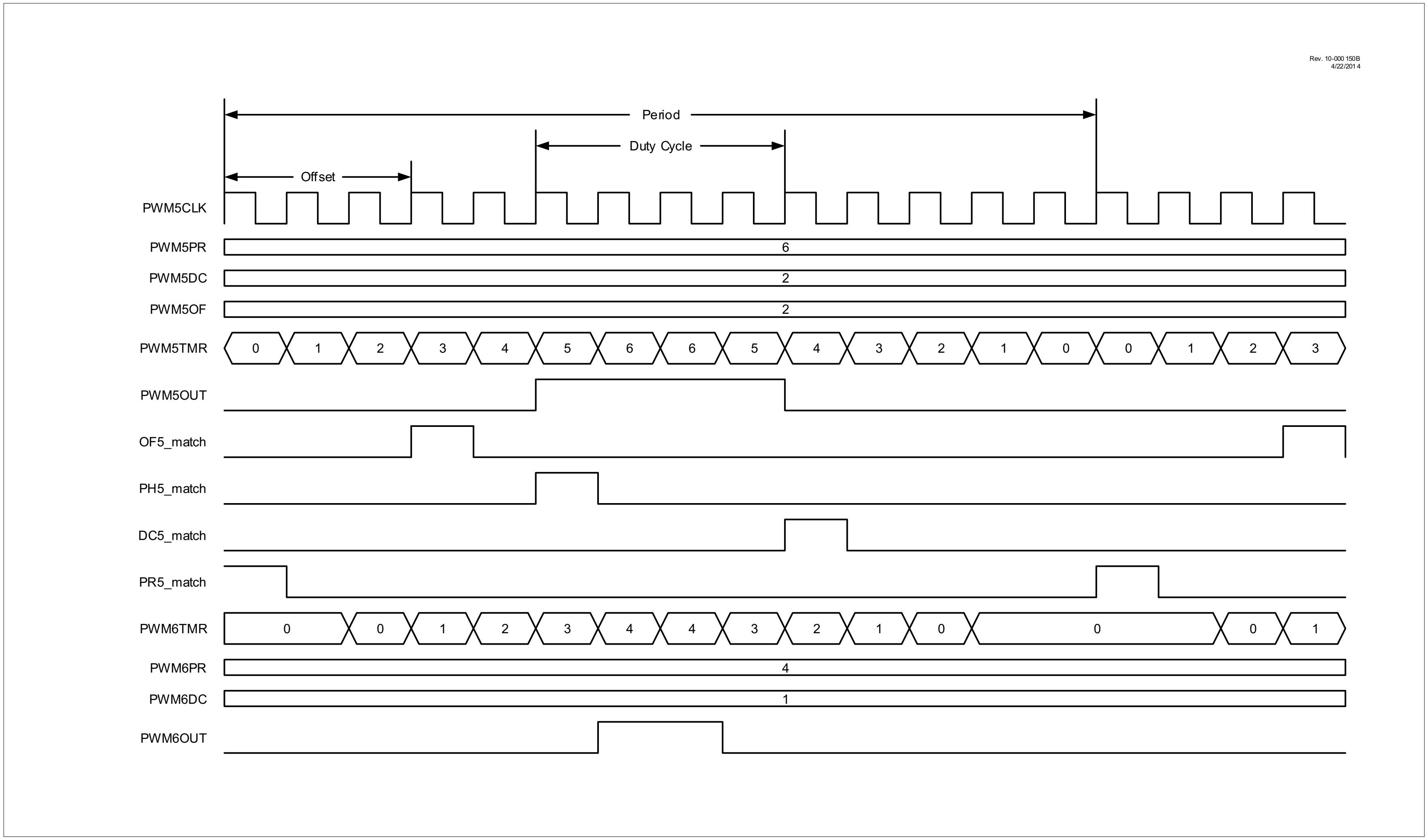
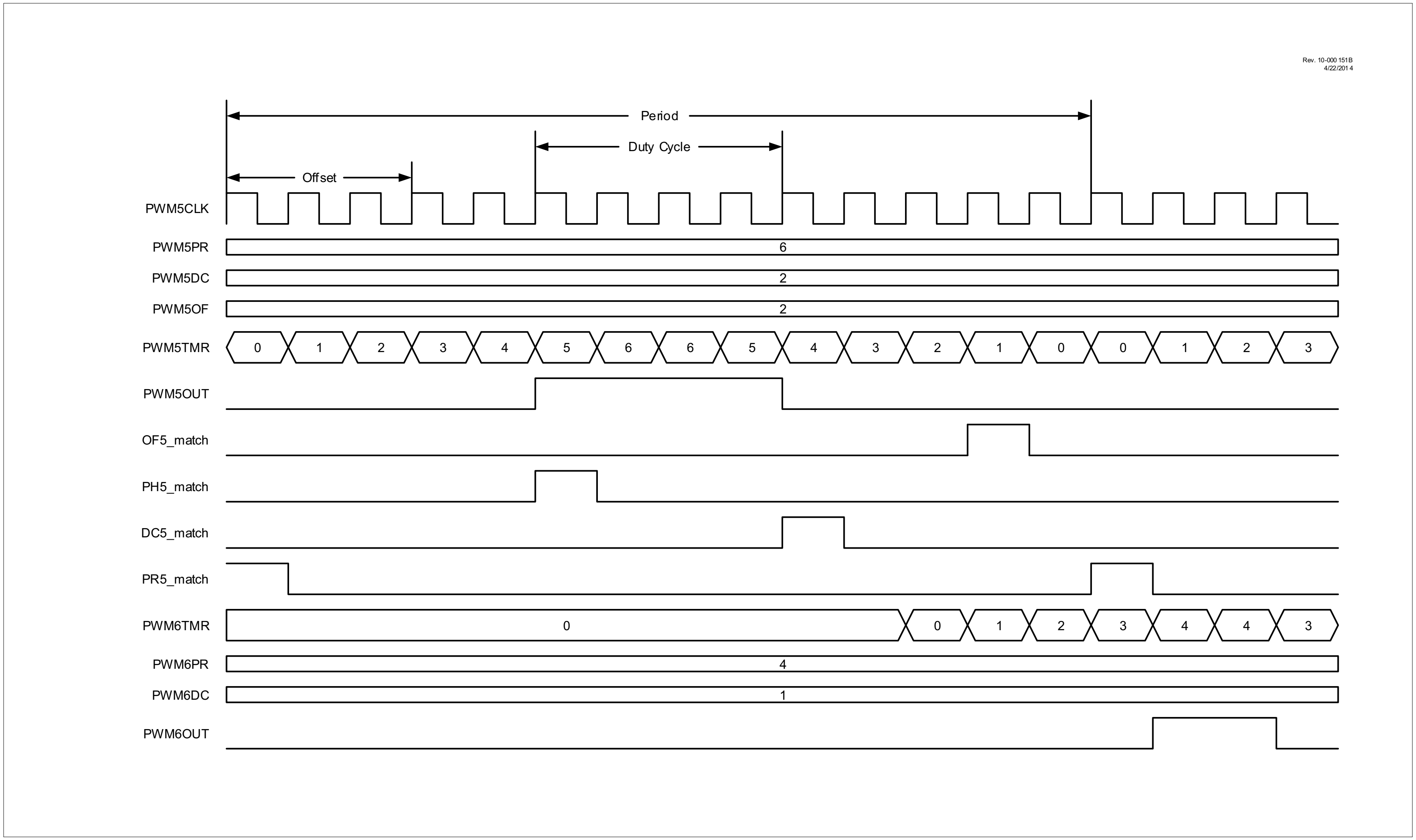


FIGURE 23-13: OFFSET MATCH ON DECREMENTING TIMER TIMING DIAGRAM



23.4 Reload Operation

Four of the PWM module control register pairs and one control bit are double buffered so that all can be updated simultaneously. These include:

- PWMxPHH:PWMxPHL register pair
- PWMxDCH:PWMxDCL register pair
- PWMxPRH:PWMxPRL register pair
- PWMxOFH:PWMxOFL register pair
- ODO control bit

When written to, these registers do not immediately affect the operation of the PWM. By default, writes to these registers will not be loaded into the PWM operating buffer registers until after the arming conditions are met. The arming control has two methods of operation:

- Immediate
- Triggered

The LDT bit of the PWMxLDCON register controls the arming method. Both methods require the LDA bit to be set. All four buffer pairs will load simultaneously at the loading event.

23.4.1 IMMEDIATE RELOAD

When the LDT bit is clear then the immediate mode is selected and the buffers will be loaded at the first period event after the LDA bit is set. Immediate reloading is used when a PWM module is operating stand-alone or when the PWM module is operating as a master to other slave PWM modules.

23.4.2 TRIGGERED RELOAD

When the LDT bit is set then the Triggered mode is selected and a trigger event is required for the LDA bit to take effect. The trigger source is the buffer load event of one of the other PWM modules in the device. The triggering source is selected by the LDS<1:0> bits of the PWMxLDCON register. The buffers will be loaded at the first period event following the trigger event. Triggered reloading is used when a PWM module is operating as a slave to another PWM and it is necessary to synchronize the buffer reloads in both modules.

Note 1:	The buffer load operation clears the LDA bit.
2:	If the LDA bit is set at the same time as PWMxTMR = PWMxPR, the LDA bit is ignored until the next period event. Such is the case when triggered reload is selected and the triggering event occurs simultaneously with the target's period event

23.5 Operation in Sleep Mode

Each PWM module will continue to operate in Sleep mode when either the HFINTOSC or LFINTOSC is selected as the clock source by PWMxCLKCON<1:0>.

23.6 Interrupts

Each PWM module has four independent interrupts based on the phase, duty cycle, period, and offset match events. The interrupt flag is set on the rising edge of each of these signals. Refer to Figures 23-8 and 23-12 for detailed timing diagrams of the match signals.

23.7 Register Definitions: PWM Control

Long bit name prefixes for the 16-bit PWM peripherals are shown in Table 23-1. Refer to **Section 1.1 "Register and Bit Naming Conventions"** for more information

TABLE 23-1:

Peripheral	Bit Name Prefix
PWM1	PWM1
PWM2	PWM2
PWM3	PWM3
PWM4	PWM4

REGISTER 23-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	U-0	R/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EN	—	OUT	POL	MODE<1:0>		—	—
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is 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 **EN:** PWM Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **OUT:** Output State of the PWM module

bit 4 **POL:** PWM Output Polarity Control bit
1 = PWM output active state is low
0 = PWM output active state is high

bit 3-2 **MODE<1:0>:** PWM Mode Control bits
11 = Center Aligned mode
10 = Toggle On Match mode
01 = Set On Match mode
00 = Standard PWM mode

bit 1-0 **Unimplemented:** Read as '0'

REGISTER 23-2: PWMxINTE: PWM INTERRUPT ENABLE 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
—	—	—	—	OFIE	PHIE	DCIE	PRIE
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 **OFIE**: Offset Interrupt Enable bit
1 = Interrupt CPU on Offset Match
0 = Do not interrupt CPU on Offset Match

bit 2 **PHIE:** Phase Interrupt Enable bit
1 = Interrupt CPU on Phase Match
0 = Do not Interrupt CPU on Phase Match

bit 1 **DCIE:** Duty Cycle Interrupt Enable bit
1 = Interrupt CPU on Duty Cycle Match
0 = Do not interrupt CPU on Duty Cycle Match

bit 0 **PRIE:** Period Interrupt Enable bit
1 = Interrupt CPU on Period Match
0 = Do not interrupt CPU on Period Match

REGISTER 23-3: PWMxINTF: PWM INTERRUPT REQUEST REGISTER

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	OFIF	PHIF	DCIF	PRIF
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is 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 **OFIF**: Offset Interrupt Flag bit(1)
1 = Offset Match Event occurred
0 = Offset Match Event did not occur

bit 2 **PHIF:** Phase Interrupt Flag bit(1)
1 = Phase Match Event occurred
0 = Phase Match Event did not occur

bit 1 **DCIF:** Duty Cycle Interrupt Flag bit(1)
1 = Duty Cycle Match Event occurred
0 = Duty Cycle Match Event did not occur

bit 0 **PRIF:** Period Interrupt Flag bit(1)
1 = Period Match Event occurred
0 = Period Match Event did not occur

Note 1: Bit is forced clear by hardware while module is disabled (EN = 0).

REGISTER 23-4: PWMxCLKCON: PWM CLOCK CONTROL REGISTER

U-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
—	PS<2:0>			—	—	CS<1:0>	
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **PS<2:0>:** Clock Source Prescaler Select bits
111 = Divide clock source by 128
110 = Divide clock source by 64
101 = Divide clock source by 32
100 = Divide clock source by 16
011 = Divide clock source by 8
010 = Divide clock source by 4
001 = Divide clock source by 2
000 = No Prescaler

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **CS<1:0>:** Clock Source Select bits
11 = Reserved
10 = LFINTOSC (continues to operate during Sleep)
01 = HFINTOSC (continues to operate during Sleep)
00 = FOSC

REGISTER 23-5: PWMxLDCON: PWM RELOAD TRIGGER SOURCE SELECT REGISTER

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
LDA(1)	LDT	—	—	—	—	LDS<1:0>	
bit 7							bit 0

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

bit 7 **LDA:** Load Buffer Armed bit(1)
If LDT = 1:
1 = Load the OFx, PHx, DCx and PRx buffers at the end of the period when the selected trigger occurs.
0 = Do not load buffers/load has completed
If LDT = 0:
1 = Load OF, PH, DC and PR buffers at the end of the current period
0 = Do not load buffers or load has completed

bit 6 **LDT:** Load Buffer on Trigger bit
1 = Load buffers on trigger enabled
0 = Load on trigger disabled
Load the OFx, PHx, DCx and PRx buffers at the end of **every** period after the selected trigger occurs.
Reload internal double buffers at the end of current period. PWMxLDS bits are ignored.

bit 5-2 **Unimplemented:** Read as ‘0’

bit 1-0 **LDS<1:0>:** Load Trigger Source Select bits
11 = LD4_trigger(2)
10 = LD3_trigger(2)
01 = LD2_trigger(2)
00 = LD1_trigger(2)

Note 1: This bit is cleared by the module after a reload operation. It can be cleared in software to clear an existing arming event.
2: The LD_trigger corresponding to the PWM used becomes reserved.

REGISTER 23-6: PWMxOFCON: PWM OFFSET TRIGGER SOURCE SELECT REGISTER

U-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
—	OFM<1:0>		OFO(2)	—	—	OFS<1:0>	
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-5 **OFM<1:0>:** Offset Mode Select bits
- 11 = Continuous Slave Run mode with Immediate Reset and synchronized start, when the selected Offset Trigger occurs.
- 10 = One-shot Slave Run mode with synchronized start, when the selected Offset Trigger occurs
- 01 = Independent Slave Run mode with synchronized start, when the selected Offset Trigger occurs
- 00 = Independent Run mode

bit 4 **OFO:** Offset Match Output Control bit

If MODE<1:0> = 11 (PWM center aligned mode):
- 1 = OFx_match occurs on counter match when counter decrementing, (second match)
- 0 = OFx_match occurs on counter match when counter incrementing, (first match)

If MODE<1:0> = 11 (all other modes):
bit is ignored

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1-0 **OFS<1:0>:** Offset Trigger Source Select bits
- 11 = OF4_match(1)
- 10 = OF3_match(1)
- 01 = OF2_match(1)
- 00 = OF1_match(1)

Note 1: The OF_match corresponding to the PWM used becomes reserved.

REGISTER 23-7: PWMxPHH: PWMx PHASE COUNT 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
PH<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 **PH<15:8>**: PWM Phase High bits
Upper eight bits of PWM phase count

REGISTER 23-8: PWMxPHL: PWMx PHASE COUNT 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
PH<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 **PH<7:0>**: PWM Phase Low bits
Lower eight bits of PWM phase count

REGISTER 23-9: PWMxDCH: PWMx DUTY CYCLE COUNT 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
DC<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	**DC<15:8>**: PWM Duty Cycle High bits Upper eight bits of PWM duty cycle count

REGISTER 23-10: PWMxDCL: PWMx DUTY CYCLE COUNT 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
DC<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	**DC<7:0>**: PWM Duty Cycle Low bits Lower eight bits of PWM duty cycle count

REGISTER 23-11: PWMxPRH: PWMx PERIOD COUNT 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
PR<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 **PR<15:8>**: PWM Period High bits
Upper eight bits of PWM period count

REGISTER 23-12: PWMxPRL: PWMx PERIOD COUNT 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
PR<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 **PR<7:0>**: PWM Period Low bits
Lower eight bits of PWM period count

REGISTER 23-13: PWMxOFH: PWMx OFFSET COUNT 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
OF<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 **OF<15:8>**: PWM Offset High bits
Upper eight bits of PWM offset count

REGISTER 23-14: PWMxOFL: PWMx OFFSET COUNT 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
OF<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 **OF<7:0>**: PWM Offset Low bits
Lower eight bits of PWM offset count

REGISTER 23-15: PWMxTMRH: PWMx TIMER 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
TMR<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 **TMR<15:8>**: PWM Timer High bits
Upper eight bits of PWM timer counter

REGISTER 23-16: PWMxTMRL: PWMx TIMER 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
TMR<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 **TMR<7:0>**: PWM Timer Low bits
Lower eight bits of PWM timer counter

Note: There are no long and short bit name variants for the following three mirror registers

REGISTER 23-17: PWMEN: PWMEN BIT ACCESS 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
—	—	—	—	PWM4EN_A	PWM3EN_A	PWM2EN_A	PWM1EN_A
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 **PWMxEN:** PWM4/PWM3/PWM2/PWM1 Enable bits
Mirror copy of EN bits in PWMxCON<7>

REGISTER 23-18: PWMLD: LD BIT ACCESS 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
—	—	—	—	PWM4LDA_A	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A
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 **PWMxLDA:** PWM4/PWM3/PWM2/PWM1 LD bits
Mirror copy of LD bits in PWMxLDCON<7>

REGISTER 23-19: PWMOUT: PWMOUT BIT ACCESS 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
—	—	—	—	PWM4OUT_A	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A
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 **PWMxOUT:** PWM4/PWM3/PWM2/PWM1 Output bits
Mirror copy of OUT bits in PWMxCON<5>

TABLE 23-2: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

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>		70
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	90
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	93
PWMEN	—	—	—	—	PWM4EN_A	PWM3EN_A	PWM2EN_A	PWM1EN_A	245
PWMLD	—	—	—	—	PWM4LDA_A	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A	245
PWMOUT	—	—	—	—	PWM4OUT_A	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A	245
PWM1PHL	PH<7:0>								240
PWM1PHH	PH<15:8>								240
PWM1DCL	DC<7:0>								241
PWM1DCH	DC<15:8>								241
PWM1PRL	PR<7:0>								242
PWM1PRH	PR<15:8>								242
PWM1OFL	OF<7:0>								243
PWM1OFH	OF<15:8>								243
PWM1TMRL	TMR<7:0>								244
PWM1TMRH	TMR<15:8>								244
PWM1CON	EN	—	OUT	POL	MODE<1:0>		—	—	235
PWM1INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	236
PWM1INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	236
PWM1CLKCON	—	PS<2:0>			—	—	CS<1:0>		237
PWM1LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		238
PWM1OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		239
PWM2PHL	PH<7:0>								240
PWM2PHH	PH<15:8>								240
PWM2DCL	DC<7:0>								241
PWM2DCH	DC<15:8>								241
PWM2PRL	PR<7:0>								242
PWM2PRH	PR<15:8>								242
PWM2OFL	OF<7:0>								243
PWM2OFH	OF<15:8>								243
PWM2TMRL	TMR<7:0>								244
PWM2TMRH	TMR<15:8>								244
PWM2CON	EN	—	OUT	POL	MODE<1:0>		—	—	235
PWM2INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	236
PWM2INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	236
PWM2CLKCON	—	PS<2:0>			—	—	CS<1:0>		237
PWM2LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		238
PWM2OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		239
PWM3PHL	PH<7:0>								240
PWM3PHH	PH<15:8>								240
PWM3DCL	DC<7:0>								241
PWM3DCH	DC<15:8>								241
PWM3PRL	PR<7:0>								242
PWM3PRH	PR<15:8>								242
PWM3OFL	OF<7:0>								243
PWM3OFH	OF<15:8>								243
PWM3TMRL	TMR<7:0>								244
PWM3TMRH	TMR<15:8>								244
PWM3CON	EN	—	OUT	POL	MODE<1:0>		—	—	235
PWM3INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	236
PWM3INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	236

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

TABLE 23-2: SUMMARY OF REGISTERS ASSOCIATED WITH PWM (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PWM3CLKCON	—	PS<2:0>			—	—	CS<1:0>		237
PWM3LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		238
PWM3OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		239
PWM4PHL	PH<7:0>								240
PWM4PHH	PH<15:8>								240
PWM4DCL	DC<7:0>								241
PWM4DCH	DC<15:8>								241
PWM4PRL	PR<7:0>								242
PWM4PRH	PR<15:8>								242
PWM4OFL	OF<7:0>								243
PWM4OFH	OF<15:8>								243
PWM4TMRL	TMR<7:0>								244
PWM4TMRH	TMR<15:8>								244
PWM4CON	EN	—	OUT	POL	MODE<1:0>		—	—	235
PWM4INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	236
PWM4INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	236
PWM4CLKCON	—	PS<2:0>			—	—	CS<1:0>		237
PWM4LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		238
PWM4OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		239

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

TABLE 23-3: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	56
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		

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

24.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces a complementary waveform with dead-band delay from a selection of input sources.

The CWG module has the following features:

- Selectable dead-band clock source control
- Selectable input sources
- Output enable control
- Output polarity control
- Dead-band control with independent 6-bit rising and falling edge dead-band counters
- Auto-shutdown control with:
  - Selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control

24.1 Fundamental Operation

The CWG generates two output waveforms from the selected input source.

The off-to-on transition of each output can be delayed from the on-to-off transition of the other output, thereby, creating a time delay immediately where neither output is driven. This is referred to as dead time and is covered in **Section 24.5 "Dead-Band Control"**. A typical operating waveform, with dead band, generated from a single input signal is shown in Figure 24-2.

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. This is referred to as auto-shutdown and is covered in **Section 24.9 "Auto-Shutdown Control"**.

24.2 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 G1CS0 bit of the CWGxCON0 register (Register 24-1).

24.3 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 24-1.

TABLE 24-1: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name
CWG input pin	CWGxIN pin
Comparator C1	C1OUT_sync
Comparator C2	C2OUT_sync
PWM1	PWM1_output
PWM2	PWM2_output
PWM3	PWM3_output
PWM4	PWM4_output

The input sources are selected using the GxIS<2:0> bits in the CWGxCON1 register (Register 24-2).

24.4 Output Control

Immediately after the CWG module is enabled, the complementary drive is configured with both CWGxA and CWGxB drives cleared.

24.4.1 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 GxPOLA and GxPOLB bits of the CWGxCON0 register.

FIGURE 24-1: SIMPLIFIED CWG BLOCK DIAGRAM

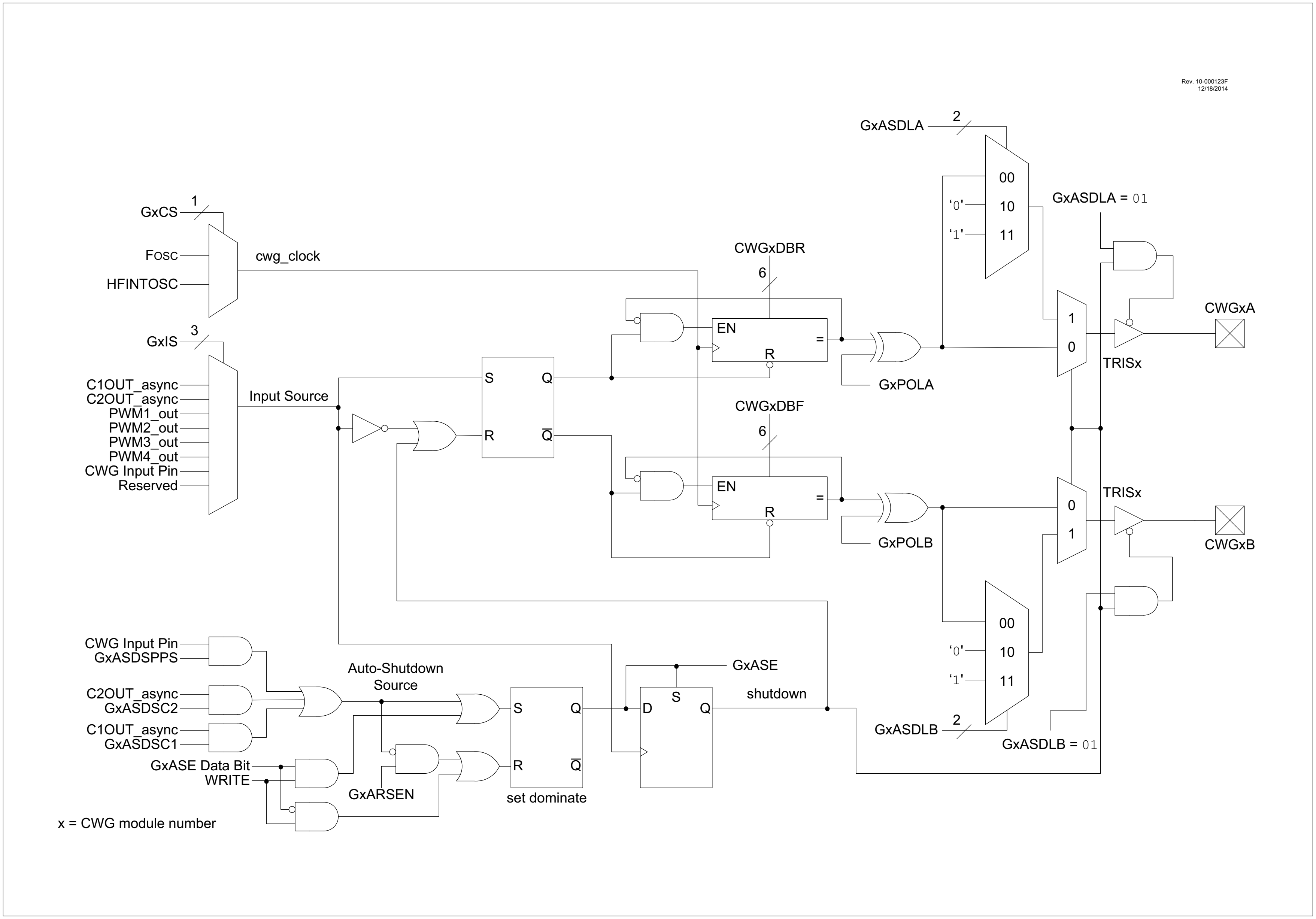
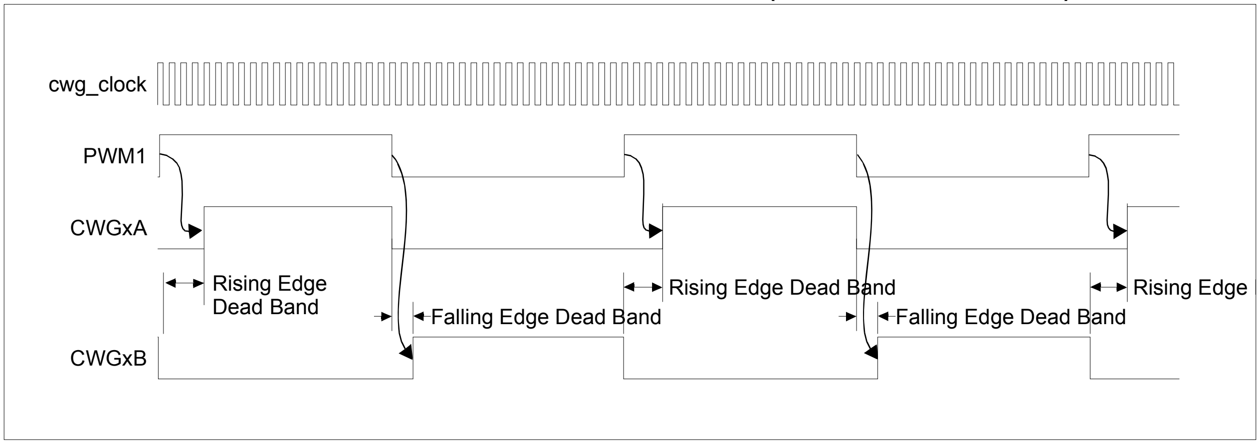


FIGURE 24-2: TYPICAL CWG OPERATION WITH PWM1 (NO AUTO-SHUTDOWN)



24.5 Dead-Band Control

Dead-band control provides for non-overlapping output signals to prevent shoot-through current in power switches. The CWG contains two 6-bit dead-band counters. One dead-band counter is used for the rising edge of the input source control. The other is used for the falling edge of the input source control.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling dead-band counter registers. See CWGxDBR and CWGxDBF registers (Register 24-4 and Register 24-5, respectively).

24.6 Rising Edge Dead Band

The rising edge dead-band delays the turn-on of the CWGxA output from when the CWGxB output is turned off. The rising edge dead-band time starts when the rising edge of the input source signal goes true. When this happens, the CWGxB output is immediately turned off and the rising edge dead-band delay time starts. When the rising edge dead-band delay time is reached, the CWGxA output is turned on.

The CWGxDBR register sets the duration of the dead-band interval on the rising edge of the input source signal. This duration is from 0 to 64 counts of dead band.

Dead band is always counted off the edge on the input source signal. A count of 0 (zero), indicates that no dead band is present.

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.

24.7 Falling Edge Dead Band

The falling edge dead band delays the turn-on of the CWGxB output from when the CWGxA output is turned off. The falling edge dead-band time starts when the falling edge of the input source goes true. When this happens, the CWGxA output is immediately turned off and the falling edge dead-band delay time starts. When the falling edge dead-band delay time is reached, the CWGxB output is turned on.

The CWGxDBF register sets the duration of the dead-band interval on the falling edge of the input source signal. This duration is from 0 to 64 counts of dead band.

Dead band is always counted off the edge on the input source signal. A count of 0 (zero), indicates that no dead band is present.

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 24-3 and Figure 24-4 for examples.

FIGURE 24-3: DEAD-BAND OPERATION, CWGxDBR = 01H, CWGxDBF = 02H

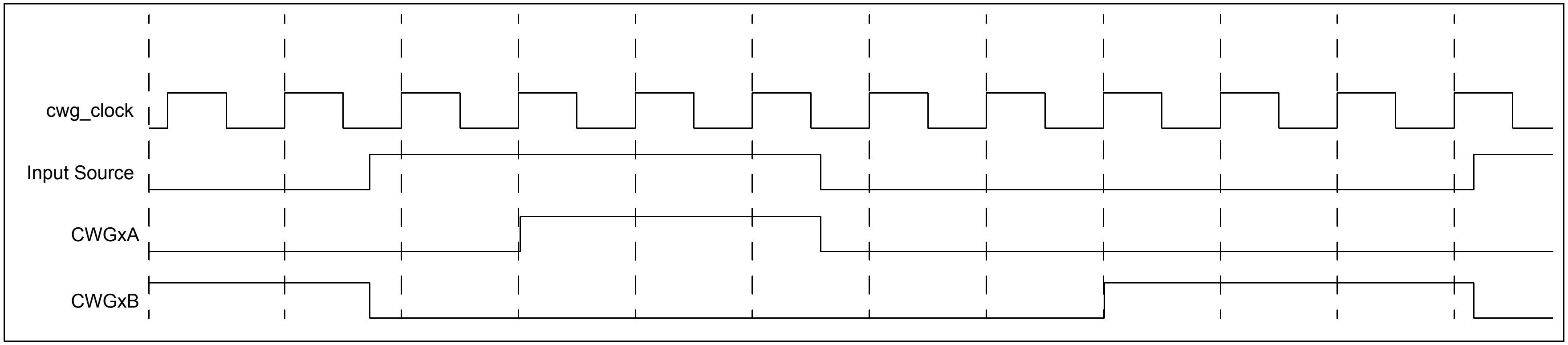
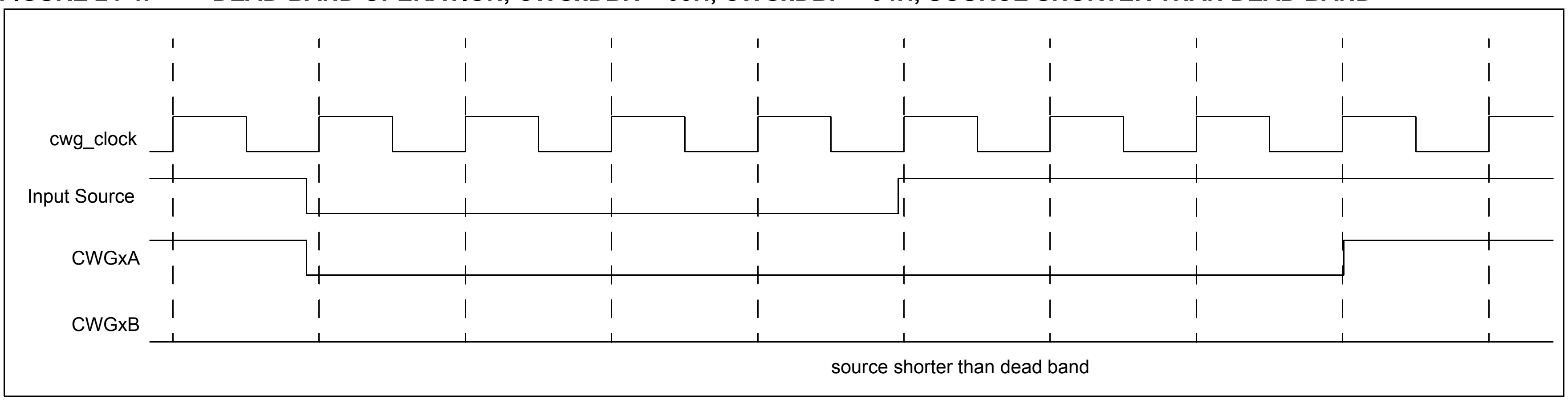


FIGURE 24-4: DEAD-BAND OPERATION, CWGxDBR = 03H, CWGxDBF = 04H, SOURCE SHORTER THAN DEAD BAND



24.8 Dead-Band Uncertainty

When the rising and falling edges of the input source triggers the dead-band counters, the input may be asynchronous. This will create some uncertainty in the dead-band time delay. The maximum uncertainty is equal to one CWG clock period. Refer to Equation 24-1 for more detail.

EQUATION 24-1: DEAD-BAND UNCERTAINTY

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

Example:

$$Fcwg\_clock = 16\ MHz$$

Therefore:

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

$$= \frac{1}{16\ MHz}$$

$$= 62.5ns$$

24.9 Auto-Shutdown Control

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.

24.9.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

24.9.1.1 Software Generated Shutdown

Setting the GxASE bit of the CWGxCON2 register will force the CWG into the shutdown state.

When auto-restart is disabled, the shutdown state will persist as long as the GxASE bit is set.

When auto-restart is enabled, the GxASE bit will clear automatically and resume operation on the next rising edge event. See Figure 24-6.

24.9.1.2 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. Any combination of two input sources can be selected to cause a shutdown condition. The sources are:

- Comparator C1 – C1OUT_async
- Comparator C2 – C2OUT_async
- $\overline{\text{CWG1FLT}}$

Shutdown inputs are selected using the GxASDS0 and GxASDS1 bits of the CWGxCON2 register. (Register 24-3).

Note:	Shutdown inputs are level sensitive, not edge sensitive. The shutdown state cannot be cleared, except by disabling auto-shutdown, as long as the shutdown input level persists.

24.10 Operation During Sleep

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

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

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

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

24.11 Configuring the CWG

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

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

24.11.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the GxASDLA and GxASDLB bits of the CWGxCON2 register (Register 24-3). GxASDLA controls the CWG1A override level and GxASDLB controls the CWG1B override level. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not apply to the override level.

24.11.2 AUTO-SHUTDOWN RESTART

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

- Software controlled
- Auto-restart

The restart method is selected with the GxARSEN bit of the CWGxCON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 24-5 and Figure 24-6.

24.11.2.1 Software Controlled Restart

When the GxARSEN bit of the CWGxCON2 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 GxASE bit will remain set. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

24.11.2.2 Auto-Restart

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

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

FIGURE 24-5: SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (GxARSEN = 0, GxASDLA = 01, GxASDLB = 01)

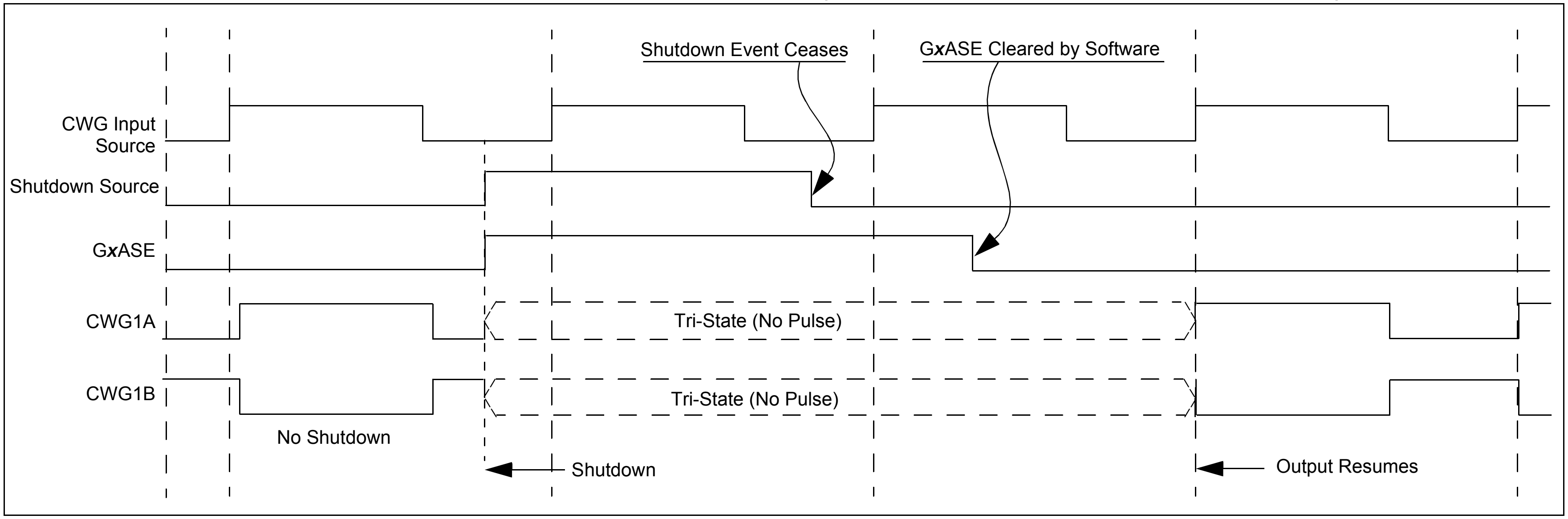
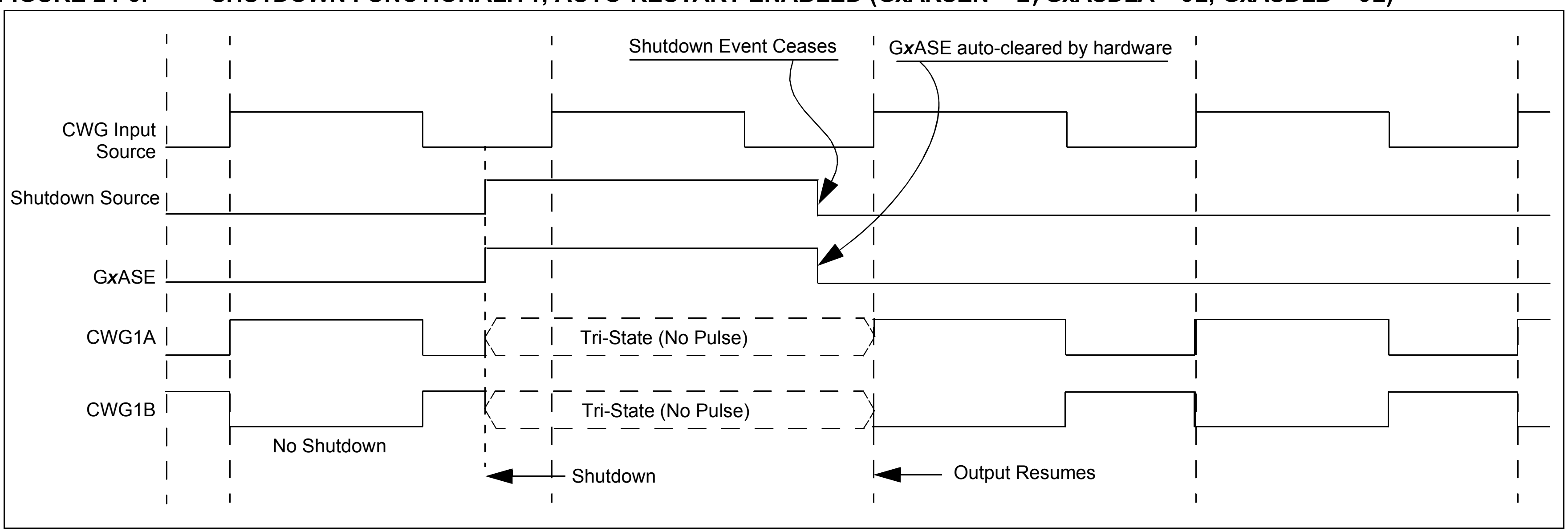


FIGURE 24-6: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (GxARSEN = 1, GxASDLA = 01, GxASDLB = 01)



24.12 Register Definitions: CWG Control

REGISTER 24-1: CWGxCON0: CWG CONTROL REGISTER 0

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
GxEN	—	—	GxPOLB	GxPOLA	—	—	GxCS0
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 **GxEN:** CWGx Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6-5 **Unimplemented**: Read as ‘0’

bit 4 **GxPOLB:** CWGxB Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 3 **GxPOLA:** CWGxA Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 2-1 **Unimplemented**: Read as ‘0’

bit 0 **GxCS0:** CWGx Clock Source Select bit
1 = HFINTOSC
0 = FOSC

REGISTER 24-2: CWGxCON1: CWG CONTROL REGISTER 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-0/0	R/W-0/0	R/W-0/0
GxASDLB<1:0>		GxASDLA<1:0>		—	GxIS<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 7-6 **GxASDLB<1:0>:** CWGx Shutdown State for CWGxB
When an auto shutdown event is present (GxASE = 1):
11 = CWGxB pin is driven to ‘1’, regardless of the setting of the GxPOLB bit.
10 = CWGxB pin is driven to ‘0’, regardless of the setting of the GxPOLB bit.
01 = CWGxB pin is tri-stated
00 = CWGxB pin is driven to its inactive state after the selected dead-band interval. GxPOLB still will control the polarity of the output.

bit 5-4 **GxASDLA<1:0>:** CWGx Shutdown State for CWGxA
When an auto shutdown event is present (GxASE = 1):
11 = CWGxA pin is driven to ‘1’, regardless of the setting of the GxPOLA bit.
10 = CWGxA pin is driven to ‘0’, regardless of the setting of the GxPOLA bit.
01 = CWGxA pin is tri-stated
00 = CWGxA pin is driven to its inactive state after the selected dead-band interval. GxPOLA still will control the polarity of the output.

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **GxIS<2:0>:** CWGx Input Source Select bits
111 = Reserved
110 = CWG input pin
101 = PWM4 – PWM4_out
100 = PWM3 – PWM3_out
011 = PWM2 – PWM2_out
010 = PWM1 – PWM1_out
001 = Comparator C2 – C2OUT_async
000 = Comparator C1 – C1OUT_async

REGISTER 24-3: CWGxCON2: CWG CONTROL REGISTER 2

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
GxASE	GxARSEN	—	—	GxASDSC2	GxASDSC1	GxASDSPPS	—
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 **GxASE:** Auto-Shutdown Event Status bit
1 = An auto-shutdown event has occurred
0 = No auto-shutdown event has occurred

bit 6 **GxARSEN:** Auto-Restart Enable bit
1 = Auto-restart is enabled
0 = Auto-restart is disabled

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3 **GxASDSC2:** CWG Auto-shutdown on Comparator C2 Enable bit
1 = Shutdown when Comparator C2 output (C2OUT_async) is high
0 = Comparator C2 output has no effect on shutdown

bit 2 **GxASDSC1:** CWG Auto-shutdown on Comparator C1 Enable bit
1 = Shutdown when Comparator C1 output (C1OUT_async) is high
0 = Comparator C1 output has no effect on shutdown

bit 1 **GxASDSPPS:** CWG Input Pin Enable bit
1 = Shutdown when CWG input pin (CWGxIN) is high
0 = CWG input pin (CWGxIN) signal has no effect on shutdown

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 24-4: CWGxDBR: COMPLEMENTARY WAVEFORM GENERATOR (CWGx) RISING DEAD-BAND COUNT 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
—	—	CWGxDBR<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 **CWGxDBR<5:0>:** Complementary Waveform Generator (CWGx) Rising Counts
`11 1111` = 63-64 counts of dead band
`11 1110` = 62-63 counts of dead band
•
•
•
`00 0010` = 2-3 counts of dead band
`00 0001` = 1-2 counts of dead band
`00 0000` = 0 counts of dead band

REGISTER 24-5: CWGxDBF: COMPLEMENTARY WAVEFORM GENERATOR (CWGx) FALLING DEAD-BAND COUNT 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
—	—	CWGxDBF<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 **CWGxDBF<5:0>:** Complementary Waveform Generator (CWGx) Falling Counts
`11 1111` = 63-64 counts of dead band
`11 1110` = 62-63 counts of dead band
•
•
•
`00 0010` = 2-3 counts of dead band
`00 0001` = 1-2 counts of dead band
`00 0000` = 0 counts of dead band. Dead-band generation is bypassed.

TABLE 24-2: 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
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	122
CWG1CON0	G1EN	—	—	G1POLB	G1POLA	—	—	G1CS0	255
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	G1IS<2:0>			256
CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC2	G1ASDSC1	G1ASDSPPS	—	257
CWG1DBF	—	—	CWG1DBF<5:0>						258
CWG1DBR	—	—	CWG1DBR<5:0>						258
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	121

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

25.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

In Program/Verify mode the program memory, user IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the "*PIC16(L)F157x Memory Programming Specification*" (DS40001766).

25.1 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.

25.2 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, MCLR must be held at VIL for as long as Program/Verify mode is to be maintained.

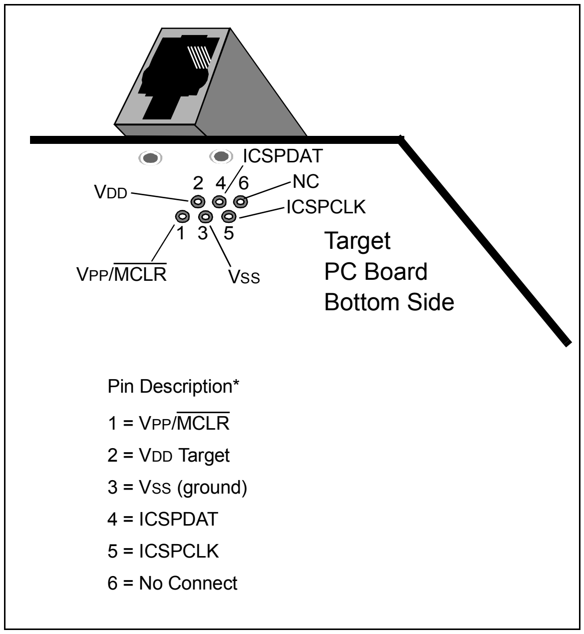
If low-voltage programming is enabled (LVP = 1), the MCLR Reset function is automatically enabled and cannot be disabled. See **Section 6.5 "MCLR"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

25.3 Common Programming Interfaces

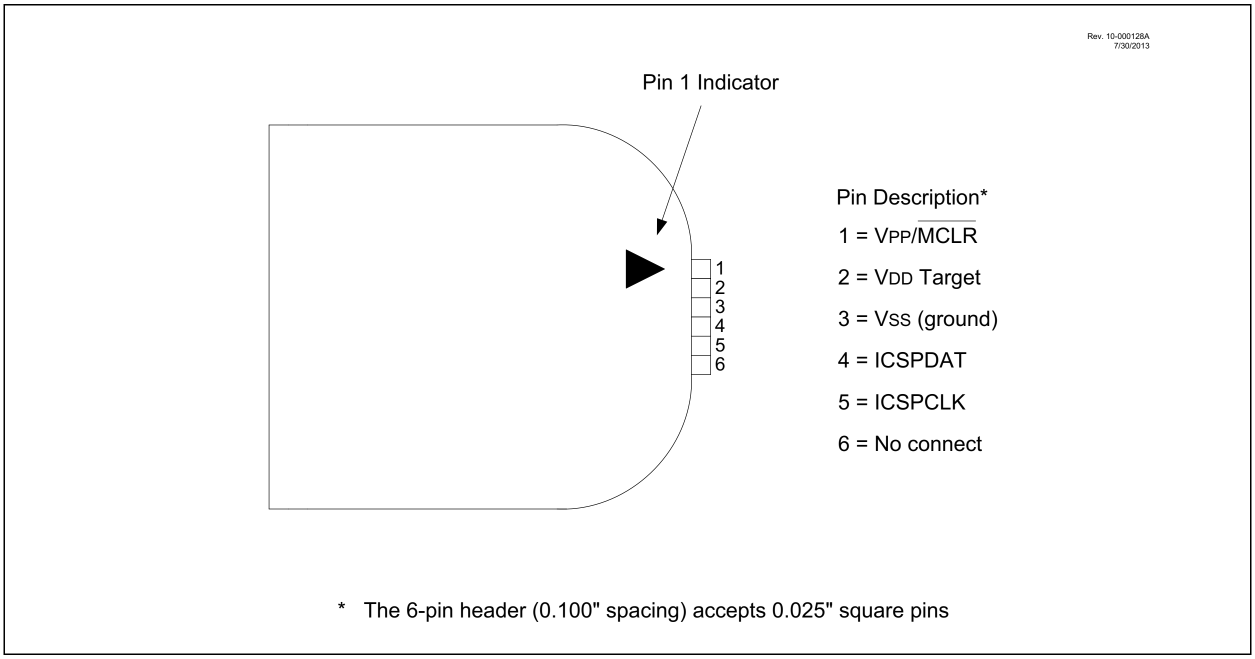
Connection to a target device is typically done through an ICSP™ header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6-connector) configuration. See Figure 25-1.

FIGURE 25-1: 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 25-2.

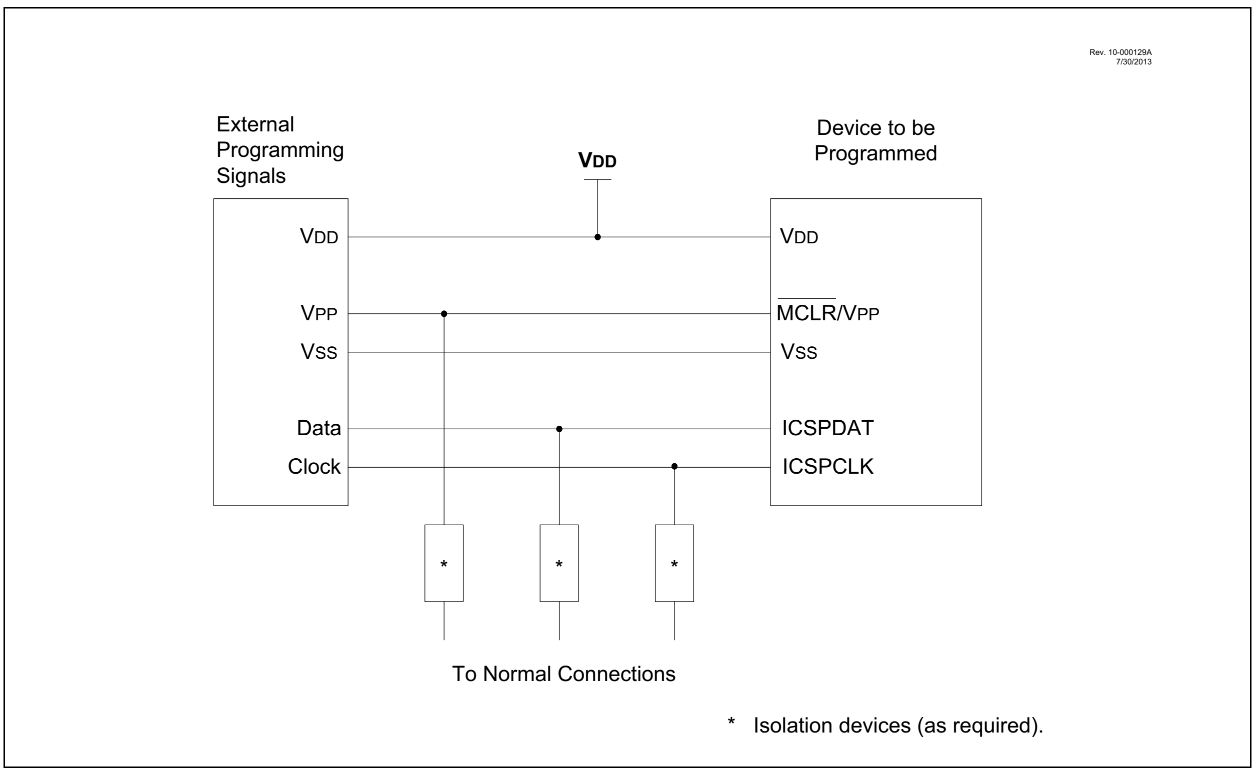
FIGURE 25-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE



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 25-3 for more information.

FIGURE 25-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



26.0 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 26-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (`CALL`, `CALLW`)
- Returns from interrupts or subroutines take two cycles (`RETURN`, `RETLW`, `RETFIE`)
- Program branching takes two cycles (`GOTO`, `BRA`, `BRW`, `BTFSS`, `BTFSC`, `DECFSZ`, `INCSFZ`)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '`0xhh`' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

26.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 26-1: 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 26-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{TO}$	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
$\overline{PD}$	Power-Down bit

FIGURE 26-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13 ... 8	7	6 ... 0
OPCODE	d	f (FILE #)

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 ... 10	9 ... 7	6 ... 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 ... 8	7 ... 0
OPCODE	k (literal)

k = 8-bit immediate value

CALL and GOTO instructions only

13 ... 11	10 ... 0
OPCODE	k (literal)

k = 11-bit immediate value

MOVLP instruction only

13 ... 7	6 ... 0
OPCODE	k (literal)

k = 7-bit immediate value

MOVLB instruction only

13 ... 5	4 ... 0
OPCODE	k (literal)

k = 5-bit immediate value

BRA instruction only

13 ... 9	8 ... 0
OPCODE	k (literal)

k = 9-bit immediate value

FSR Offset instructions

13 ... 7	6	5 ... 0
OPCODE	n	k (literal)

n = appropriate FSR
k = 6-bit immediate value

FSR Increment instructions

13 ... 3	2	1 ... 0
OPCODE	n	m (mode)

n = appropriate FSR
m = 2-bit mode value

OPCODE only

13 ... 0
OPCODE

TABLE 26-3: ENHANCED MID-RANGE INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	–	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
BYTE ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
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 SKIP OPERATIONS									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

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.

TABLE 26-3: ENHANCED MID-RANGE INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
CONTROL OPERATIONS									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	–	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	–	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
INHERENT OPERATIONS									
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$, $\overline{PD}$	
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	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register with W	1	00	0000	0110	0fff		
C-COMPILER OPTIMIZED									
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00	0000	0001	0nmm kkkk	Z	**2, 3**
	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 modifier, mm	1	00	0000	0001	kkkk		**2, 3**
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk			**2**

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.

26.2 Instruction Descriptions

ADDFSR — Add Literal to FSRn

Syntax:	[*label*] ADDFSR FSRn, k
Operands:	$-32 \le k \le 31$ $n \in [0, 1]$
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.

ADDLW — Add literal and W

Syntax:	[*label*] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	$(W) + k \rightarrow (W)$
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.

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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'.

ADDWFC — ADD W and CARRY bit to f

Syntax:	[*label*] ADDWFC f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(W) + (f) + (C) \rightarrow$ dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory 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'.

ANDLW — AND literal with W

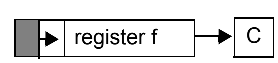
Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. (k) $\rightarrow$ (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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'.

ASRF — Arithmetic Right Shift

Syntax:	[*label*] ASRF f {,d}
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<7>) $\rightarrow$ dest<7> (f<7:1>) $\rightarrow$ dest<6:0>, (f<0>) $\rightarrow$ 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'.



BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit ‘b’ in register ‘f’ is cleared.

BRA	Relative Branch
Syntax:	[*label*] BRA label [*label*] BRA $+k
Operands:	$-256 \le$ label - PC + 1 ≤ 255 $-256 \le k \le 255$
Operation:	(PC) + 1 + k → 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) → PC
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit ‘b’ in register ‘f’ is set.

BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	skip if (f<b>) = 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.

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<b>) = 1
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘0’, the next instruction is executed. If bit ‘b’ is ‘1’, then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

CALL — Call Subroutine

Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 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 instruction.

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.

CLRF — Clear f

Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register ‘f’ are cleared and the Z bit is set.

CLRW — Clear W

Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT — Clear Watchdog Timer

Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF — Complement f

Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\overline{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register ‘f’ are complemented. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

DECF — Decrement f

Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

DECFSZ — **Decrement f, Skip if 0**

Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [0,1]
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are decremented. 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:	[*label*] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	k → PC<10:0> PCLATH<6:3> → PC<14:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

INCF — **Increment f**

Syntax:	[*label*] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	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 is placed back in register ‘f’.

INCFSZ — **Increment f, Skip if 0**

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [0,1]
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	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 is placed back in register ‘f’. If the result is ‘1’, the next instruction is executed. If the result is ‘0’, a NOP is executed instead, making it a 2-cycle instruction.

IORLW — **Inclusive OR literal with W**

Syntax:	[*label*] IORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the 8-bit literal ‘k’. The result is placed in the W register.

IORWF — **Inclusive OR W with f**

Syntax:	[*label*] IORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [0,1]
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register ‘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:	[*label*] LSLF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>) → C (f<6:0>) → dest<7:1> 0 → 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'.

LSRF — Logical Right Shift

Syntax:	[*label*] LSRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	0 → 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. 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'.

MOVF — Move f

Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register f is moved to a destination dependent upon the 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.
Words:	1
Cycles:	1
<u>Example:</u>	`MOVF FSR, 0`
	After Instruction W = value in FSR register Z = 1

MOVIW — Move INDFn to W

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
$-32 \leq k \leq 31$

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

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

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

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

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

MOVLB — Move literal to BSR

Syntax: [*label*] MOVLB k

Operands: $0 \leq k \leq 31$

Operation: k → BSR

Status Affected: None

Description: The 5-bit literal ‘k’ is loaded into the Bank Select Register (BSR).

MOVLP — Move literal to PCLATH

Syntax: [*label*] MOVLP k

Operands: $0 \leq k \leq 127$

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal ‘k’ is loaded into the PCLATH register.

MOVLW — Move literal to W

Syntax: [*label*] MOVLW k

Operands: $0 \leq k \leq 255$

Operation: k → (W)

Status Affected: None

Description: The 8-bit literal ‘k’ is loaded into W register. The “don’t cares” will assemble as ‘0’s.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

MOVWF — Move W to f

Syntax: [*label*] MOVWF f

Operands: $0 \leq f \leq 127$

Operation: (W) → (f)

Status Affected: None

Description: Move data from W register to register ‘f’.

Words: 1

Cycles: 1

Example:
```
MOVWF   OPTION_REG
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

MOVWI — Move W to INDFn

Syntax: [*label*] MOVWI ++FSRn
[*label*] MOVWI --FSRn
[*label*] MOVWI FSRn++
[*label*] MOVWI FSRn--
[*label*] MOVWI k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
$-32 \leq k \leq 31$

Operation: W → INDFn
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected: None

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

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

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

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

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

NOP — No Operation

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation.

Words: 1

Cycles: 1

Example: `NOP`

OPTION — Load OPTION_REG Register with W

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION_REG

Status Affected: None

Description: Move data from W register to OPTION_REG register.

RESET — Software Reset

Syntax: [*label*] RESET

Operands: None

Operation: Execute a device Reset. Resets the nRI flag of the PCON register.

Status Affected: None

Description: This instruction provides a way to execute a hardware Reset by software.

RETFIE — Return from Interrupt

Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS $\rightarrow$ PC, 1 $\rightarrow$ GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words:	1
Cycles:	2
<u>Example:</u>	`RETFIE` After Interrupt PC = TOS GIE = `1`

RETLW — Return with literal in W

Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k $\rightarrow$ (W); TOS $\rightarrow$ PC
Status Affected:	None
Description:	The W register is loaded with the 8-bit 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

<u>Example:</u>

```
      CALL TABLE;W contains table
          ;offset value
      •   ;W now has table value
TABLE •
      •
      ADDWF PC  ;W = offset
      RETLW k1  ;Begin table
      RETLW k2  ;
      •
      •
      •
      RETLW kn  ; End of table
```

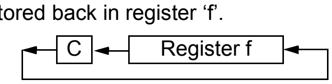
Before Instruction
W = 0x07
After Instruction
W = value of k8

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.

RLF — Rotate Left f through Carry

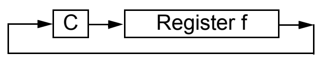
Syntax:	[*label*] RLF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '`0`', the result is placed in the W register. If 'd' is '`1`', the result is stored back in register 'f'.



Words:	1
Cycles:	1
<u>Example:</u>	`RLF REG1,0` Before Instruction `REG1 = 1110 0110` `C = 0` After Instruction `REG1 = 1110 0110` `W = 1100 1100` `C = 1`

RRF — Rotate Right f through Carry

Syntax:	[*label*] RRF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
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’.



SLEEP — Enter Sleep mode

Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped.

SUBLW — Subtract W from literal

Syntax:	[*label*] SUBLW k
Operands:	0 ≤ k ≤ 255
Operation:	k - (W) → (W)
Status Affected:	C, DC, Z
Description:	The W register is subtracted (2’s complement method) from the 8-bit literal ‘k’. The result is placed in the W register.

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF — Subtract W from f

Syntax:	[*label*] SUBWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - (W) → (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 ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SUBWFB — Subtract W from f with Borrow

Syntax:	SUBWFB f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) – (W) – ($\overline{B}$) → dest
Status Affected:	C, DC, Z
Description:	Subtract W and the BORROW flag (CARRY) from register ‘f’ (2’s complement 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 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register ‘f’ are exchanged. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed in register ‘f’.

TRIS	Load TRIS Register with W
Syntax:	[*label*] TRIS f
Operands:	5 ≤ f ≤ 7
Operation:	(W) → 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 ≤ k ≤ 255
Operation:	(W) .XOR. k → (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:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .XOR. (f) → (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’.

27.0 ELECTRICAL SPECIFICATIONS

27.1 Absolute Maximum Ratings(†)

Ambient temperature under bias .... -40°C to +125°C
Storage temperature .... -65°C to +150°C
Voltage on pins with respect to V_{SS}
on V_{DD} pin
PIC16F1574/5/8/9 .... -0.3V to +6.5V
PIC16LF1574/5/8/9 .... -0.3V to +4.0V
on $\overline{MCLR}$ pin .... -0.3V to +9.0V
on all other pins .... -0.3V to (V_{DD} + 0.3V)
Maximum current
on V_{SS} pin(1)
$-40°C \leq T_A \leq +85°C$ .... 170 mA
$-40°C \leq T_A \leq +125°C$ .... 70 mA
on V_{DD} pin(1)
$-40°C \leq T_A \leq +85°C$ .... 170 mA
$-40°C \leq T_A \leq +125°C$ .... 70 mA
on any I/O pin .... ±25 mA
Clamp current, I_K (V_{PIN} < 0 or V_{PIN} > V_{DD}) .... ±20 mA

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 27-6: “Thermal Characteristics” to calculate device specifications.

† 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.

27.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \le V_{DD} \le V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \le T_A \le T_{A\_MAX}$

VDD — Operating Supply Voltage(1)

PIC16LF1574/5/8/9

- V_{DDMIN} (Fosc ≤ 16 MHz) .... +1.8V
- V_{DDMIN} (Fosc ≤ 32 MHz) .... +2.5V
- V_{DDMAX} .... +3.6V

PIC16F1574/5/8/9

- V_{DDMIN} (Fosc ≤ 16 MHz) .... +2.3V
- V_{DDMIN} (Fosc ≤ 32 MHz) .... +2.5V
- V_{DDMAX} .... +5.5V

TA — Operating Ambient Temperature Range

Industrial Temperature

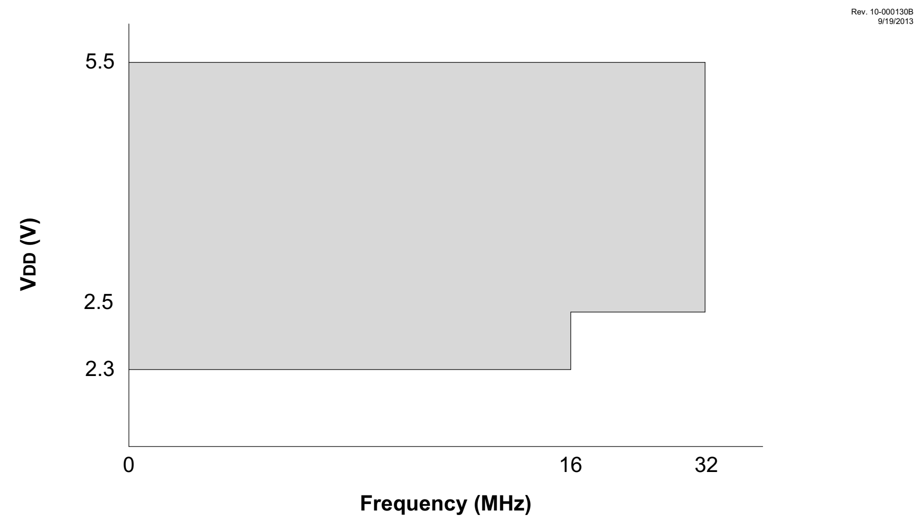
- T_{A_MIN} .... -40°C
- T_{A_MAX} .... +85°C

Extended Temperature

- T_{A_MIN} .... -40°C
- T_{A_MAX} .... +125°C

Note 1: See Parameter D001, DS Characteristics: Supply Voltage.

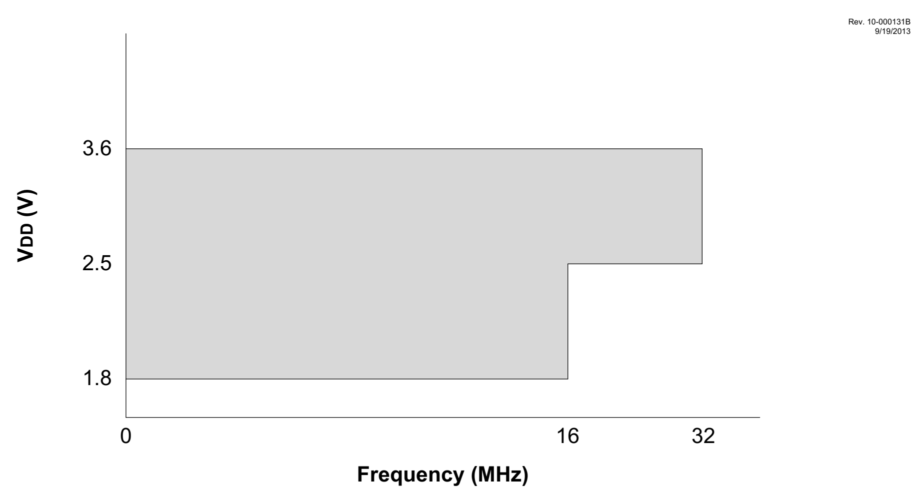
FIGURE 27-1: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16F1574/5/8/9 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: Refer to Table 27-7 for each Oscillator mode's supported frequencies.

FIGURE 27-2: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16LF1574/5/8/9 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: Refer to Table 27-7 for each Oscillator mode's supported frequencies.

27.3 DC Characteristics

TABLE 27-1: SUPPLY VOLTAGE

PIC16LF1574/5/8/9			Standard Operating Conditions (unless otherwise stated)				
PIC16F1574/5/8/9							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
D001	VDD	**Supply Voltage**					
			VDDMIN 1.8 2.5	 — —	VDDMAX 3.6 3.6	 V V	 FOSC ≤ 16 MHz FOSC ≤ 32 MHz **(Note 3)**
D001			2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz **(Note 3)**
D002*	VDR	**RAM Data Retention Voltage(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°C ≤ TA ≤ +85°C
D003A	VADFVR	**FVR Gain Voltage Accuracy for ADC**	-4	—	+4	%	1x VFVR, ADFVR = 01, VDD ≥ 2.5V 2x VFVR, ADFVR = 10, VDD ≥ 2.5V 4x VFVR, ADFVR = 11, VDD ≥ 4.75V
D003B	VCDAFVR	**FVR Gain Voltage Accuracy for Comparator**	-4	—	+4	%	1x VFVR, CDAFVR = 01, VDD ≥ 2.5V 2x VFVR, CDAFVR = 10, VDD ≥ 2.5V 4x VFVR, CDAFVR = 11, VDD ≥ 4.75V
D004*	SVDD	**VDD Rise Rate(2)**	0.05	—	—	V/ms	Ensures that the Power-on Reset signal is released properly.

* These parameters are characterized but not tested.

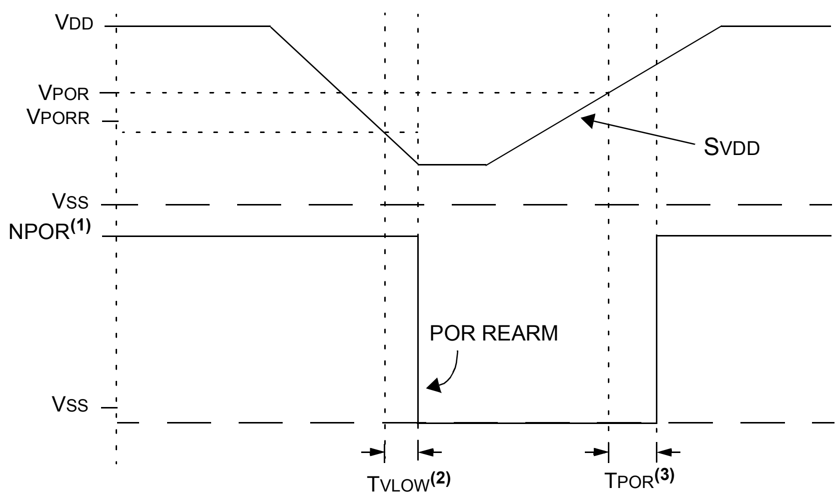
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: See Figure 27-3, POR and POR REARM with Slow Rising VDD.

3: PLL required for 32 MHz operation.

FIGURE 27-3: POR AND POR REARM WITH SLOW RISING VDD



Note 1: When NPOR is low, the device is held in Reset.
2: TPOR 1 μs typical.
3: TVLOW 2.7 μs typical.

TABLE 27-2: SUPPLY CURRENT (IDD)[1,2]

PIC16LF1574/5/8/9		Standard Operating Conditions (unless otherwise stated)					
PIC16F1574/5/8/9							
Param. No.	**Device Characteristics**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**	
						VDD	**Note**
D013		—	41	51	μA	1.8	FOSC = 1 MHz, External Clock (ECM), Medium-Power mode
		—	69	80	μA	3.0	
D013		—	79	107	μA	2.3	FOSC = 1 MHz, External Clock (ECM), Medium-Power mode
		—	105	138	μA	3.0	
		—	151	184	μA	5.0	
D014		—	134	152	μA	1.8	FOSC = 4 MHz, External Clock (ECM), Medium-Power mode
		—	234	268	μA	3.0	
D014		—	201	255	μA	2.3	FOSC = 4 MHz, External Clock (ECM), Medium-Power mode
		—	270	329	μA	3.0	
		—	344	431	μA	5.0	
D015		—	7	13	μA	1.8	FOSC = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	9	14	μA	3.0	
D015		—	15	25	μA	2.3	FOSC = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	18	28	μA	3.0	
		—	20	29	μA	5.0	
D016		—	128	174	μA	1.8	FOSC = 500 kHz, MFINTOSC
		—	153	203	μA	3.0	
D016		—	166	241	μA	2.3	FOSC = 500 kHz, MFINTOSC
		—	187	273	μA	3.0	
		—	249	332	μA	5.0	
D017*		—	0.6	0.7	mA	1.8	FOSC = 8 MHz, HFINTOSC
		—	0.9	1.1	mA	3.0	
D017*		—	0.7	1.0	mA	2.3	FOSC = 8 MHz, HFINTOSC
		—	1.0	1.1	mA	3.0	
		—	1.1	1.2	mA	5.0	
D018		—	0.9	1.0	mA	1.8	FOSC = 16 MHz, HFINTOSC
		—	1.3	1.4	mA	3.0	
D018		—	1.1	1.3	mA	2.3	FOSC = 16 MHz, HFINTOSC
		—	1.3	1.5	mA	3.0	
		—	1.5	1.8	mA	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.

Note 1: The test conditions for all IDD measurements in active operation mode are: CLKIN = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VSS; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

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

3: PLL required for 32 MHz operation.

TABLE 27-2: SUPPLY CURRENT (IDD)(1,2) (CONTINUED)

PIC16LF1574/5/8/9		Standard Operating Conditions (unless otherwise stated)					
PIC16F1574/5/8/9							
Param. No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
D018A*		—	2.3	2.8	mA	3.0	FOSC = 32 MHz, HFINTOSC **(Note 3)**
D018A*		—	2.5	2.9	mA	3.0	FOSC = 32 MHz, HFINTOSC **(Note 3)**
		—	2.6	3.0	mA	5.0	
D019A		—	2.0	2.2	mA	3.0	FOSC = 32 MHz, External Clock (ECH), High-Power mode **(Note 3)**
D019A		—	2.1	2.3	mA	3.0	FOSC = 32 MHz, External Clock (ECH), High-Power mode **(Note 3)**
		—	2.2	2.7	mA	5.0	
D019B		—	2.6	6.8	μA	1.8	FOSC = 32 kHz, External Clock (ECL), Low-Power mode
		—	5.0	9.6	μA	3.0	
D019B		—	14	23	μA	2.3	FOSC = 32 kHz, External Clock (ECL), Low-Power mode
		—	18	29	μA	3.0	
		—	20	30	μA	5.0	
D019C		—	21	29	μA	1.8	FOSC = 500 kHz, External Clock (ECL), Low-Power mode
		—	35	44	μA	3.0	
D019C		—	34	46	μA	2.3	FOSC = 500 kHz, External Clock (ECL), Low-Power mode
		—	43	59	μA	3.0	
		—	49	61	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.

Note 1: The test conditions for all IDD measurements in active operation mode are: CLKIN = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VSS; $\overline{MCLR}$ = VDD; WDT disabled.

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

3: PLL required for 32 MHz operation.

TABLE 27-3: POWER-DOWN CURRENTS (IPD)(1,2)

PIC16LF1574/5/8/9		Operating Conditions: (unless otherwise stated) Low-Power Sleep Mode						
PIC16F1574/5/8/9		**Low-Power Sleep Mode, VREGPM = 1**						
Param. No.	**Device Characteristics**	**Min.**	**Typ†**	**Max. +85°C**	**Max. +125°C**	**Units**	**Conditions**	
							V_{DD}	**Note**
D022	Base IPD	—	0.10	0.7	3.0	μA	1.8	WDT, BOR, and FVR disabled, all Peripherals inactive
		—	0.10	1.0	3.4	μA	3.0	
D022	Base IPD	—	0.3	1.1	3.3	μA	2.3	WDT, BOR, and FVR disabled, all Peripherals inactive, Low-Power Sleep mode, VREGPM = 1
		—	0.4	3.5	4.4	μA	3.0	
		—	0.5	4.2	5.2	μA	5.0	
D022A	Base IPD	—	10.4	16.1	17.3	μA	2.3	WDT, BOR, and FVR disabled, all Peripherals inactive, Normal-Power Sleep mode, VREGPM = 0
		—	12.7	21.9	24.2	μA	3.0	
		—	13.8	24.2	25.3	μA	5.0	
D023		—	0.4	1.0	3.4	μA	1.8	WDT Current
		—	0.6	1.3	4.1	μA	3.0	
D023		—	0.6	2.0	4.8	μA	2.3	WDT Current
		—	0.7	2.2	5.1	μA	3.0	
		—	0.9	2.5	5.5	μA	5.0	
D023A		—	15	21	23	μA	1.8	FVR Current
		—	26	33	34	μA	3.0	
D023A		—	19	28	29	μA	2.3	FVR Current
		—	22	35	36	μA	3.0	
		—	23	38	41	μA	5.0	
D024		—	7.5	10.4	12.7	μA	3.0	BOR Current
D024		—	8.1	11.5	12.7	μA	3.0	BOR Current
		—	9.2	13.8	15	μA	5.0	
D24A		—	0.3	2.3	4.6	μA	3.0	LPBOR Current
D24A		—	0.5	2.3	4.6	μA	3.0	LPBOR Current
		—	0.6	3.5	5.8	μA	5.0	
D026		—	0.1	0.9	3.2	μA	1.8	ADC Current **(Note 3)**, No conversion in progress
		—	0.1	1.0	3.5	μA	3.0	
D026		—	0.3	1.5	4.4	μA	2.3	ADC Current **(Note 3)**, No conversion in progress
		—	0.4	1.7	4.5	μA	3.0	
		—	0.5	1.8	4.6	μA	5.0	
D026A*		—	288	—	—	μA	1.8	ADC Current **(Note 3)**, Conversion in progress
		—	288	—	—	μA	3.0	
D026A*		—	322	—	—	μA	2.3	ADC Current **(Note 3)**, Conversion in progress
		—	322	—	—	μA	3.0	
		—	322	—	—	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.

Note 1: The peripheral Δ 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 V_{SS}.

3: ADC clock source is FRC.

TABLE 27-3: POWER-DOWN CURRENTS (I_{PD})(1,2) (CONTINUED)

PIC16LF1574/5/8/9		Operating Conditions: (unless otherwise stated) Low-Power Sleep Mode						
PIC16F1574/5/8/9		**Low-Power Sleep Mode, VREGPM = 1**						
Param. No.	**Device Characteristics**	**Min.**	**Typ†**	**Max. +85°C**	**Max. +125°C**	**Units**	**Conditions**	
							VDD	**Note**
D027		—	5	9	11	μA	1.8	Comparator, CxSP = 0
		—	5	10	12	μA	3.0	
D027		—	15	23	25	μA	2.3	Comparator, CxSP = 0
		—	17	27	29	μA	3.0	
		—	19	28	30	μA	5.0	
D028A		—	23	41	42	μA	1.8	Comparator, Normal Power, CxSP = 1 **(Note 1)**
		—	25	42	44	μA	3.0	
D028A		—	33	55	56	μA	2.3	Comparator, Normal Power, CxSP = 1 VREGPM = 1 **(Note 1)**
		—	34	59	60	μA	3.0	
		—	36	60	61	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.

Note 1: The peripheral Δ 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 27-4: I/O PORTS

Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
Standard Operating Conditions (unless otherwise stated)							
	VIL	**Input Low Voltage**					
		I/O PORT:					
D030		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D030A			—	—	0.15 VDD	V	1.8V ≤ VDD ≤ 4.5V
D031		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
		with I²C™ levels	—	—	0.3 VDD	V	
		with SMbus levels	—	—	0.8	V	2.7V ≤ VDD ≤ 5.5V
D032		$\overline{MCLR}$	—	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O PORT:					
D040		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD ≤ 4.5V
D041		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
		with I²C™ levels	0.7 VDD	—	—	V	
		with SMbus levels	2.1	—	—	V	2.7V ≤ VDD ≤ 5.5V
D042		$\overline{MCLR}$	0.8 VDD	—	—	V	
	IIL	**Input Leakage Current(1)**					
D060		I/O Ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
			—	± 5	± 1000	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 125°C
D061		$\overline{MCLR}$(2)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
	IPUR	**Weak Pull-up Current**					
D070*			25	100	200	μA	VDD = 3.3V, VPIN = VSS
			25	140	300	μA	VDD = 5.0V, VPIN = VSS
	VOL	**Output Low Voltage**					
D080		I/O Ports	—	—	0.6	V	IOL = 8 mA, VDD = 5V IOL = 6 mA, VDD = 3.3V IOL = 1.8 mA, VDD = 1.8V
	VOH	**Output High Voltage**					
D090		I/O Ports	VDD - 0.7	—	—	V	IOH = 3.5 mA, VDD = 5V IOH = 3 mA, VDD = 3.3V IOH = 1 mA, VDD = 1.8V
		Capacitive Loading Specifications on Output Pins					
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 $\overline{MCLR}$ 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.

TABLE 27-5: MEMORY PROGRAMMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
		Program Memory Programming Specifications					
D110	V_{IHH}	Voltage on $\overline{MCLR}$/V_{PP} pin	8.0	—	9.0	V	**(Note 2)**
D111	I_{DDP}	Supply Current during Programming	—	—	10	mA	
D112	V_{BE}	V_{DD} for Bulk Erase	2.7	—	V_{DDMAX}	V	
D113	V_{PEW}	V_{DD} for Write or Row Erase	V_{DDMIN}	—	V_{DDMAX}	V	
D114	I_{PPPGM}	Current on $\overline{MCLR}$/V_{PP} during Erase/Write	—	1.0	—	mA	
D115	I_{DDPGM}	Current on V_{DD} during Erase/Write	—	5.0	—	mA	
		Program Flash Memory					
D121	E_P	Cell Endurance	10K	—	—	E/W	-40°C ≤ T_A ≤ +85°C **(Note 1)**
D122	V_{PRW}	V_{DD} for Read/Write	V_{DDMIN}	—	V_{DDMAX}	V	
D123	T_{IW}	Self-timed Write Cycle Time	—	2	2.5	ms	
D124	T_{RETD}	Characteristic Retention	—	40	—	Year	Provided no other specifications are violated
D125	E_{HEFC}	High-Endurance Flash Cell	100K	—	—	E/W	0°C ≤ T_A ≤ +60°C, lower byte last 128 addresses

† 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 27-6: THERMAL CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)					
Param. No.	**Sym.**	**Characteristic**	**Typ.**	**Units**	**Conditions**
TH01	θ_{JA}	Thermal Resistance Junction to Ambient	70	°C/W	14-pin PDIP package
			95.3	°C/W	14-pin SOIC package
			100	°C/W	14-pin TSSOP package
			31.8	°C/W	16-pin UQFN 4x4mm package
			62.2	°C/W	20-pin PDIP package
			77.7	°C/W	20-pin SOIC package
			87.3	°C/W	20-pin SSOP package
			32.8	°C/W	20-pin UQFN 4x4mm package
TH02	θ_{JC}	Thermal Resistance Junction to Case	32.75	°C/W	14-pin PDIP package
			31	°C/W	14-pin SOIC package
			24.4	°C/W	14-pin TSSOP package
			24.4	°C/W	16-pin UQFN 4x4mm package
			27.5	°C/W	20-pin PDIP package
			23.1	°C/W	20-pin SOIC package
			31.1	°C/W	20-pin SSOP package
			27.4	°C/W	20-pin UQFN 4x4mm package
TH03	T_{JMAX}	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	Internal Power Dissipation	—	W	PINTERNAL = IDD x VDD(1)
TH06	PI/O	I/O Power Dissipation	—	W	$P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	PDER	Derated Power	—	W	$P_{DER} = PD_{MAX} (T_J - T_A)/\theta_{JA}$(2)

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

27.4 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	CLKIN
ck	CLKOUT	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ or $\overline{\text{WR}}$
di	SDIx	sc	SCKx
do	SDO	ss	$\overline{\text{SS}}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 27-4: LOAD CONDITIONS

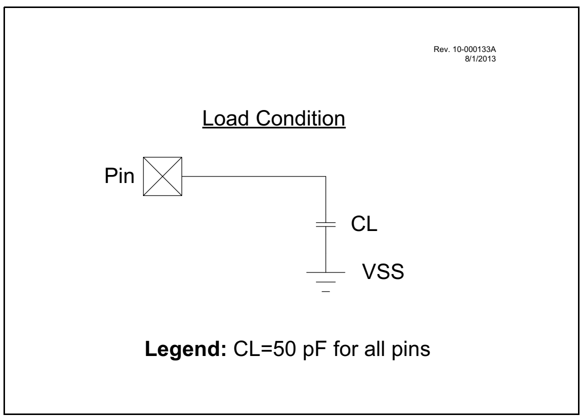
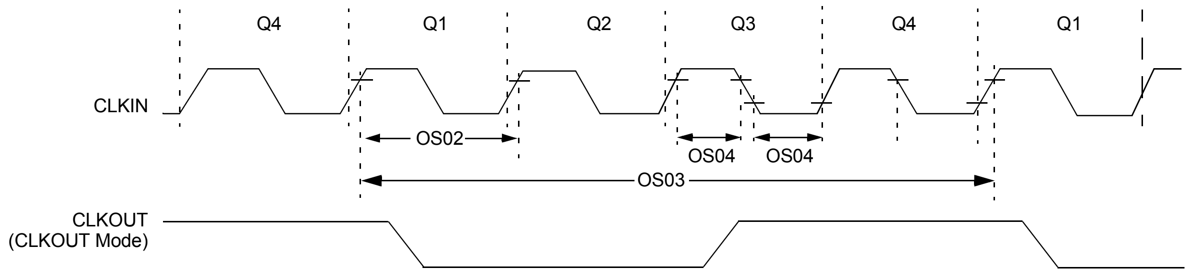


FIGURE 27-5: CLOCK TIMING



Note 1: See Table 27-10.

TABLE 27-7: CLOCK OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)

Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS01	FOSC	External CLKIN Frequency(1)	DC	—	0.5	MHz	External Clock (ECL)
			DC	—	4	MHz	External Clock (ECM)
			DC	—	20	MHz	External Clock (ECH)
OS02	TOSC	External CLKIN Period(1)	50	—	∞	ns	External Clock (EC)
OS03	TCY	Instruction Cycle Time(1)	200	TCY	DC	ns	TCY = 4/FOSC

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min” values with an external clock applied to CLKIN pin. When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

TABLE 27-8: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	**Sym.**	**Characteristic**	**Freq. Tolerance**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency[1]	±2%	—	16.0	—	MHz	VDD = 3.0V, TA = 25°C, **(Note 2)**
OS09	LFOSC	Internal LFINTOSC Frequency	—	—	31	—	kHz	
OS10*	TIOSC ST	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	15	μs	
OS10A*	TLFOSC ST	LFINTOSC Wake-up from Sleep Start-up Time	—	—	0.5	—	ms	-40°C ≤ TA ≤ +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: 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 27-6: "HFINTOSC Frequency Accuracy over Device VDD and Temperature.

FIGURE 27-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE

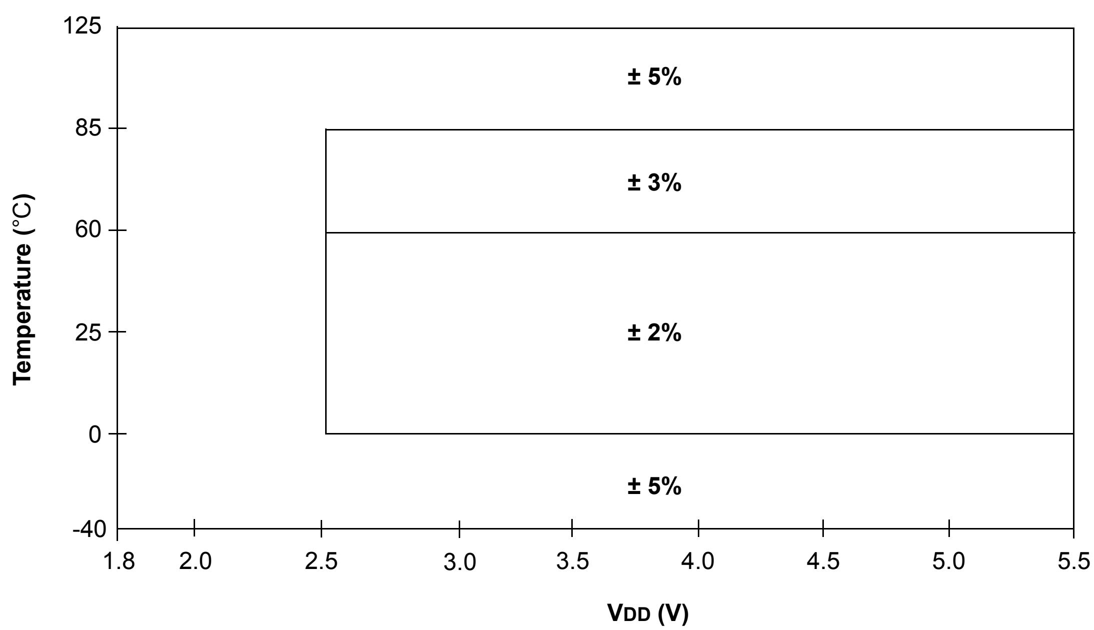


TABLE 27-9: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.7V TO 5.5V)

Param No.	Sym.	Characteristic	Min.	Typ†	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 3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 27-7: CLKOUT AND I/O TIMING

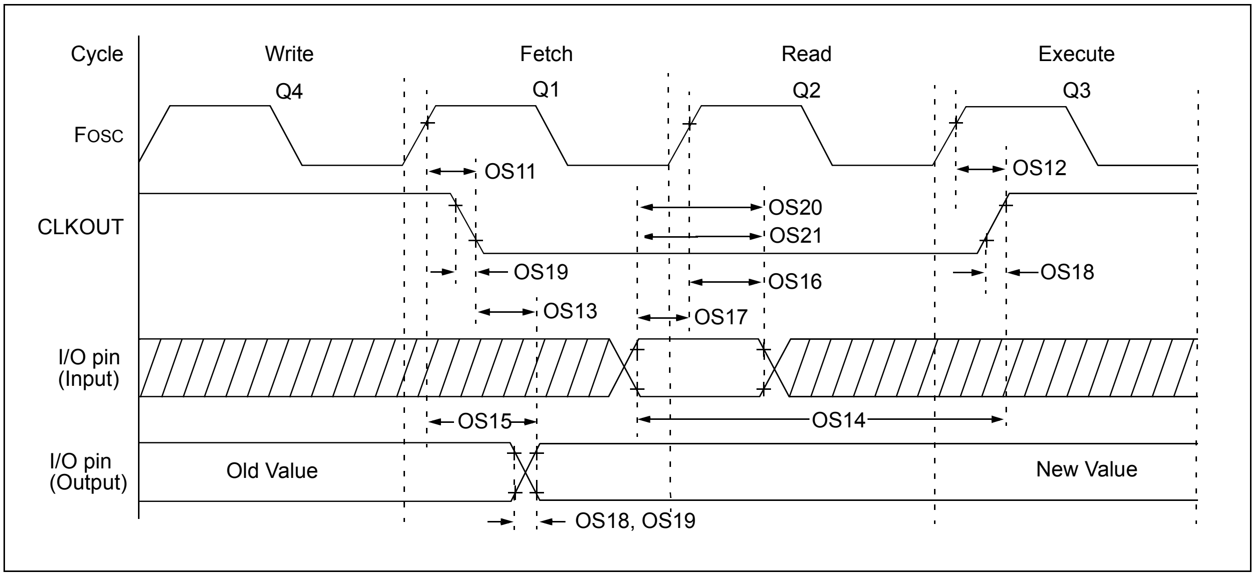
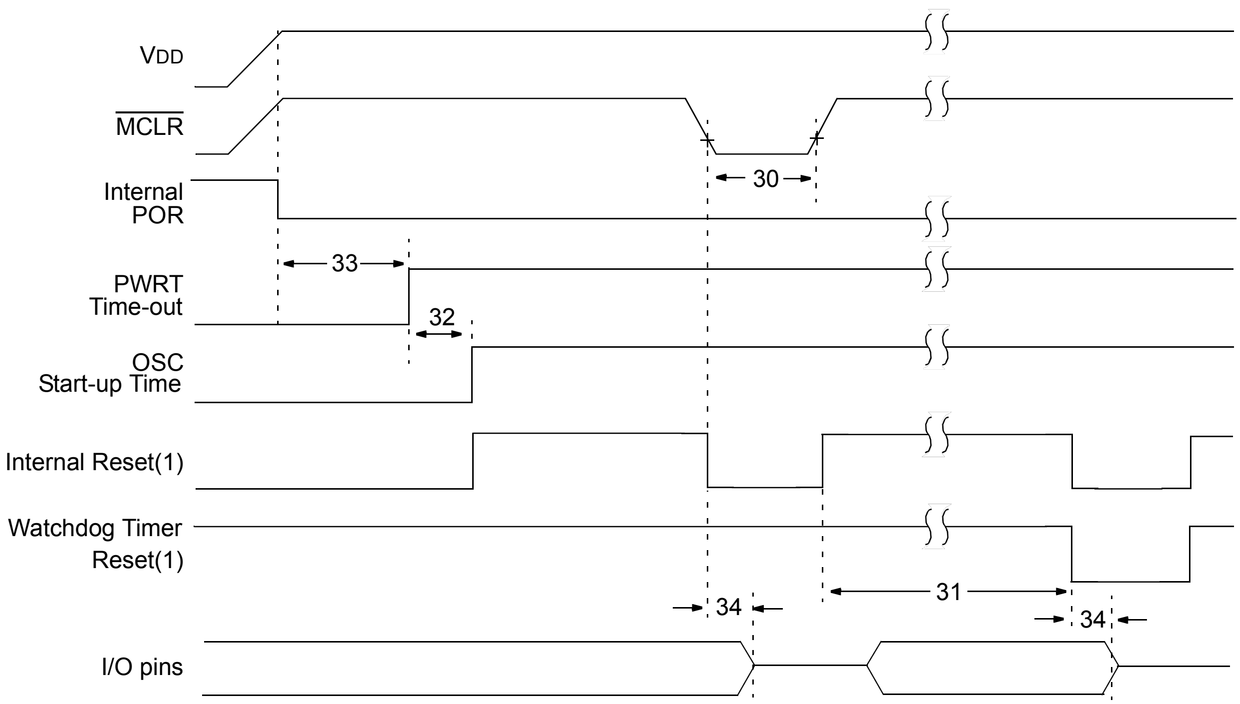


TABLE 27-10: CLKOUT AND I/O TIMING PARAMETERS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS11	TosH2ckL	FOSC↑ to CLKOUT↓(1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	FOSC↑ to CLKOUT↑(1)	—	—	72	ns	3.3V ≤ VDD ≤ 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 ≤ VDD ≤ 5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in setup time)	50	—	—	ns	3.3V ≤ VDD ≤ 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	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS19*	TioF	Port output fall time	— —	28 15	55 30	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	—	—	ns	

* These parameters are characterized but not tested.
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.
Note 1: Measurements are taken in EXTRC mode where CLKOUT output is 4 x TOSC.

FIGURE 27-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

TABLE 27-11: 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.**	**Typ†**	**Max.**	**Units**	**Conditions**
30	TMCL	$\overline{MCLR}$ Pulse Width (low)	2	—	—	μs	
31	TWDTLP	Low-Power Watchdog Timer Time-out Period	10	16	27	ms	VDD = 3.3V-5V, 1:16 Prescaler used
32	TOST	Oscillator Start-up Timer Period(1)	—	1024	—	TOSC	
33*	TPWRT	Power-up Timer Period	40	65	140	ms	$\overline{PWRTE}$ = 0
34*	TIOZ	I/O high-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0	μs	
35	VBOR	Brown-out Reset Voltage(2)	2.55 2.35 1.80	2.70 2.45 1.90	2.85 2.58 2.05	V V V	BORV = 0 BORV = 1 (PIC16F1574/5/8/9) BORV = 1 (PIC16LF1574/5/8/9)
36*	VHYST	Brown-out Reset Hysteresis	0	25	60	mV	-40°C ≤ TA ≤ +85°C
37*	TBORDC	Brown-out Reset DC Response Time	1	16	35	μs	VDD ≤ VBOR
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.

FIGURE 27-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS

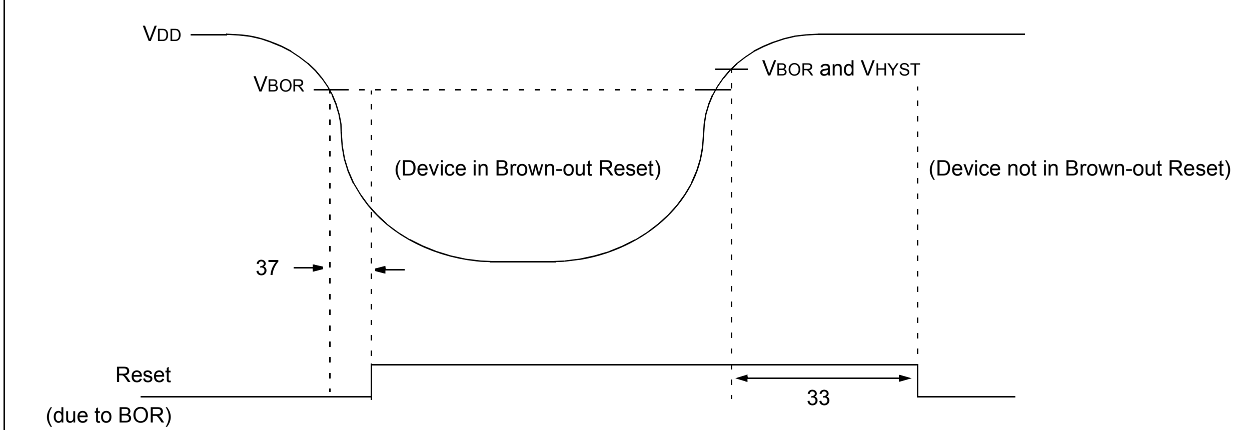


FIGURE 27-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

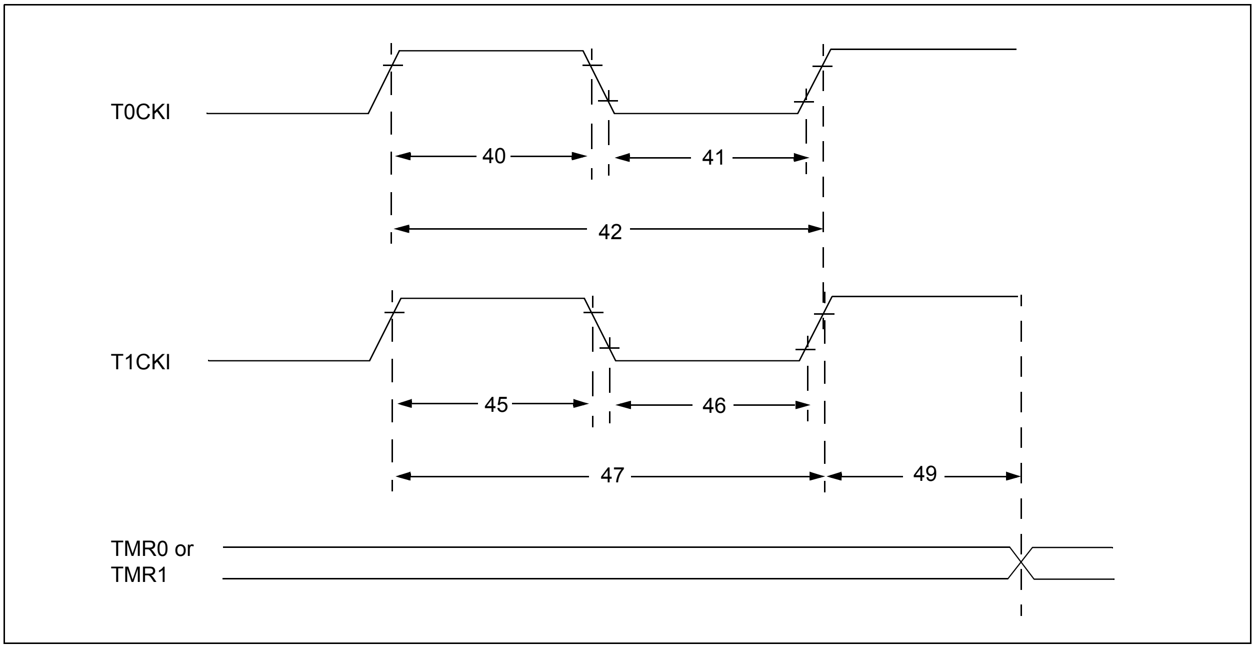


TABLE 27-12: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
40*	TT0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period		Greater of: 20 or (TCY + 40)/N	—	—	ns	N = prescale value
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	Greater of: 30 or (TCY + 40)/N	—	—	ns	N = prescale value
			Asynchronous	60	—	—	ns	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 27-13: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS(1,2,3)

Operating Conditions (unless otherwise stated)
V_{DD} = 3.0V, T_A = 25°C

Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD01	N_R	Resolution	—	—	10	bit	
AD02	E_{IL}	Integral Error	—	±1	±1.7	LSb	V_{REF} = 3.0V
AD03	E_{DL}	Differential Error	—	±1	±1	LSb	No missing codes V_{REF} = 3.0V
AD04	E_{OFF}	Offset Error	—	±1	±2.5	LSb	V_{REF} = 3.0V
AD05	E_{GN}	Gain Error	—	±1	±2.0	LSb	V_{REF} = 3.0V
AD06	V_{REF}	Reference Voltage	1.8	—	V_{DD}	V	V_{REF} = (V_{RPOS} - V_{RNEG}) (**Note 4**)
AD07	V_{AIN}	Full-Scale Range	V_{SS}	—	V_{REF}	V	
AD08	Z_{AIN}	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	Can go higher if external 0.01μF capacitor is present on input pin.

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Total Absolute Error includes integral, differential, offset and gain errors.

2: The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

3: See **Section 28.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

4: ADC V_{REF} is selected by ADPREF<0> bit.

FIGURE 27-11: ADC CONVERSION TIMING (ADC CLOCK FOSC-BASED)

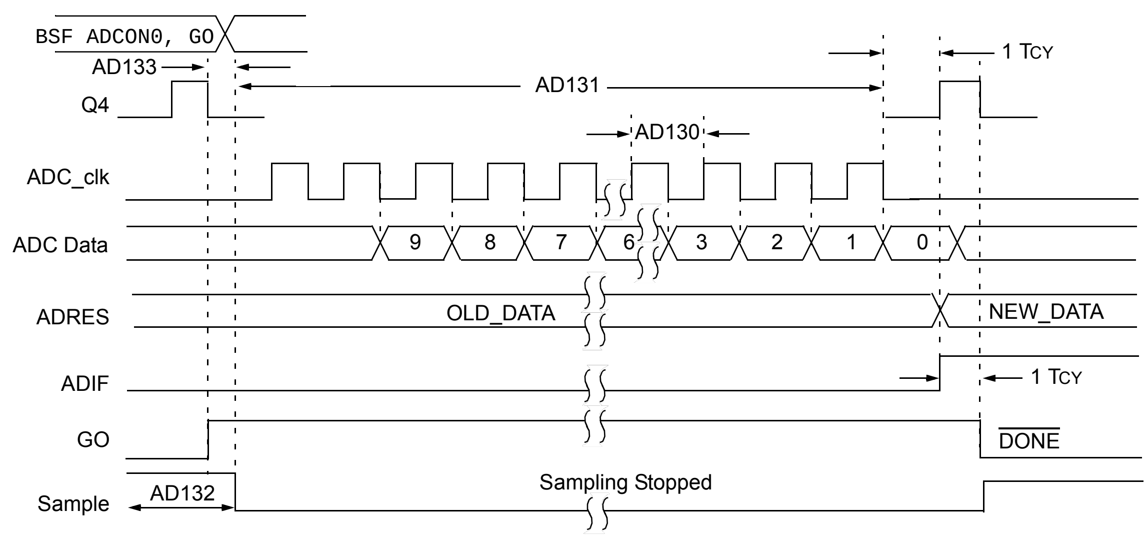
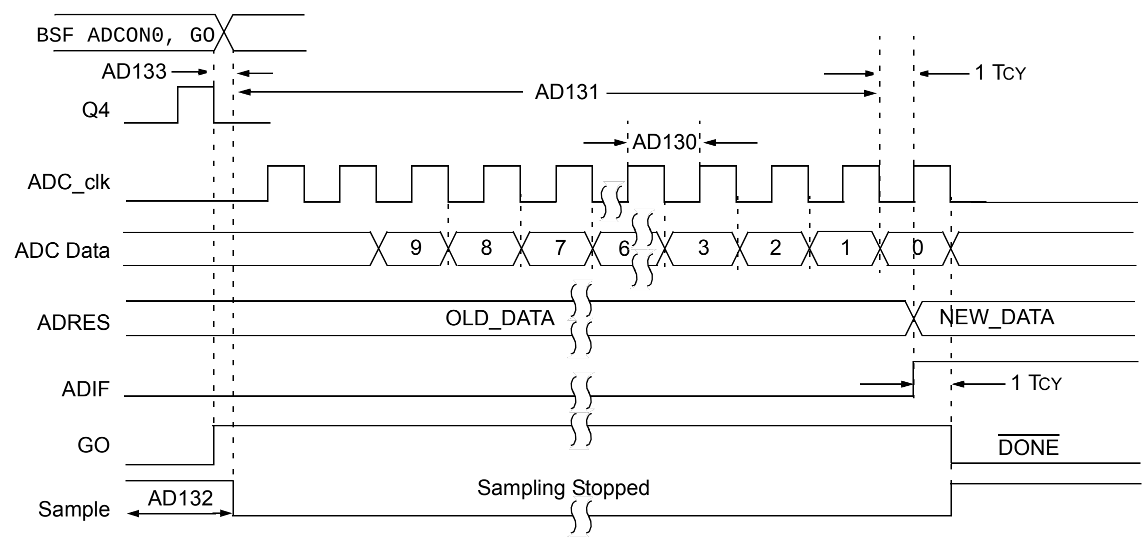


FIGURE 27-12: ADC CONVERSION TIMING (ADC CLOCK FROM FRC)



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

TABLE 27-14: ADC CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**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*	THCD	Holding Capacitor Disconnect Time	— —	1/2 TAD 1/2 TAD + 1TCY	— —		FOSC-based ADCS<2:0> = x11 (ADC FRC mode)

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The ADRES register may be read on the following TCY cycle.

TABLE 27-15: COMPARATOR SPECIFICATIONS(1)

Operating Conditions (unless otherwise stated)
VDD = 3.0V, TA = 25°C

Param. No.	Sym.	Characteristics	Min.	Typ.	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 Ration	—	50	—	dB	
CM04A	TRESP(2)	Response Time Rising Edge	—	400	800	ns	CxSP = 1
CM04B		Response Time Falling Edge	—	200	400	ns	CxSP = 1
CM04C		Response Time Rising Edge	—	1200	—	ns	CxSP = 0
CM04D		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 28.0 "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 27-16: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS(1)

Operating Conditions (unless otherwise stated)
VDD = 3.0V, TA = 25°C

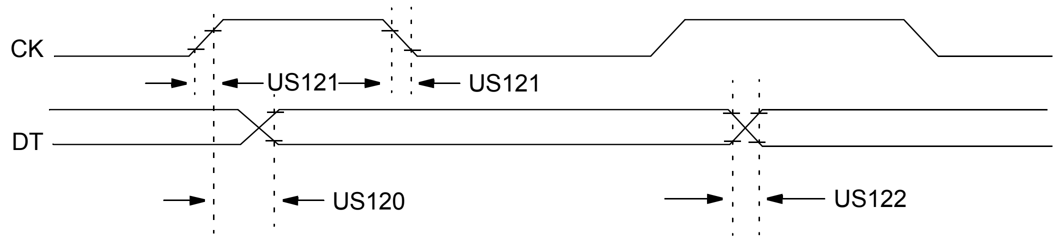
Param. No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
DAC01*	CLSB	Step Size	—	VDD/32	—	V	
DAC02*	CACC	Absolute Accuracy	—	—	± 1/2	LSb	
DAC03*	CR	Unit Resistor Value (R)	—	5K	—	Ω	
DAC04*	CST	Settling Time(2)	—	—	10	µs	

* These parameters are characterized but not tested.

Note 1: See **Section 28.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

2: Settling time measured while DACR<4:0> transitions from '0000' to '1111'.

FIGURE 27-13: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

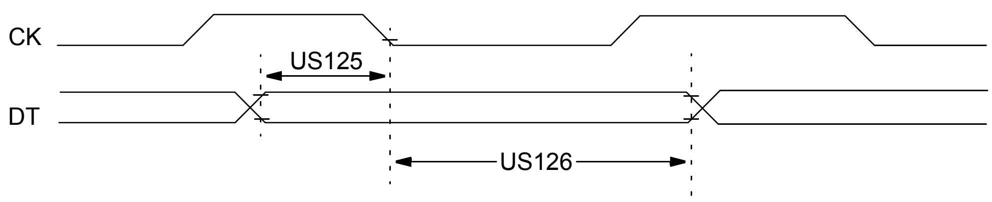


Note: Refer to Figure 27-4 for load conditions.

TABLE 27-17: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US120	TCKH2DTV	SYNC XMIT (Master and Slave) Clock high to data-out valid	—	80	ns	3.0V ≤ VDD ≤ 5.5V
			—	100	ns	1.8V ≤ VDD ≤ 5.5V
US121	TCKRF	Clock out rise time and fall time (Master mode)	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V
US122	TDTRF	Data-out rise time and fall time	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V

FIGURE 27-14: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 27-4 for load conditions.

TABLE 27-18: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US125	TDTV2CKL	SYNC RCV (Master and Slave) Data-hold before CK ↓ (DT hold time)	10	—	ns	
US126	TCKL2DTL	Data-hold after CK ↓ (DT hold time)	15	—	ns	

28.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

Graphs and charts are not available at this time.

29.0 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

29.1 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

29.2 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

29.3 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

29.4 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

29.5 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

29.6 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.

29.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

29.8 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 high-speed 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.

29.9 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 full-speed 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™).

29.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} 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.

29.11 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.

29.12 Third-Party Development Tools

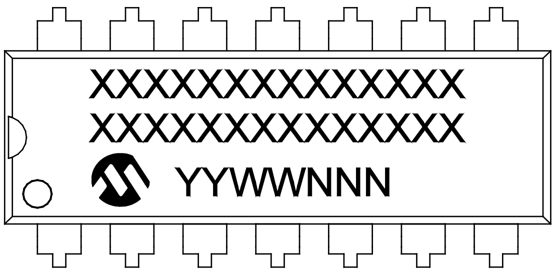
Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- 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®

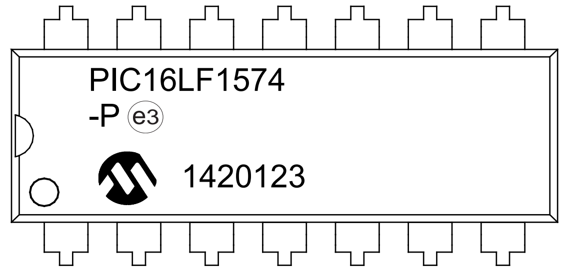
30.0 PACKAGING INFORMATION

30.1 Package Marking Information

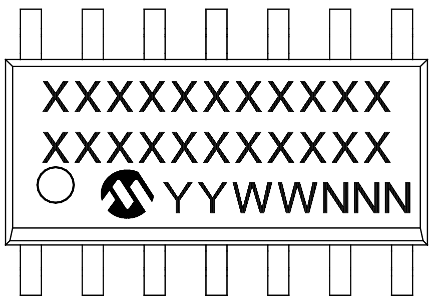
14-Lead PDIP (300 mil)



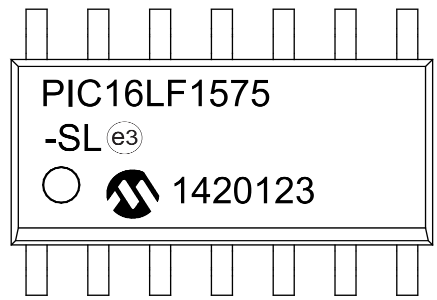
Example



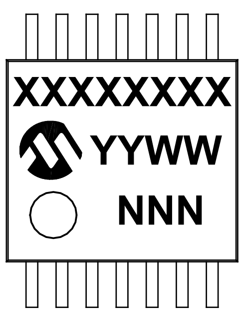
14-Lead SOIC (3.90 mm)



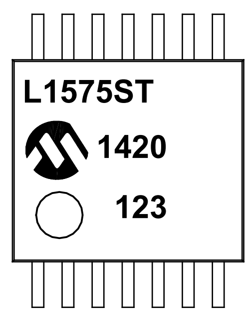
Example



14-Lead TSSOP (4.4 mm)



Example

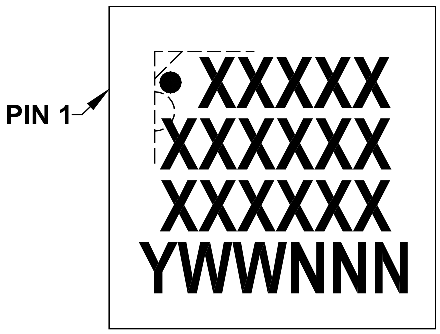


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

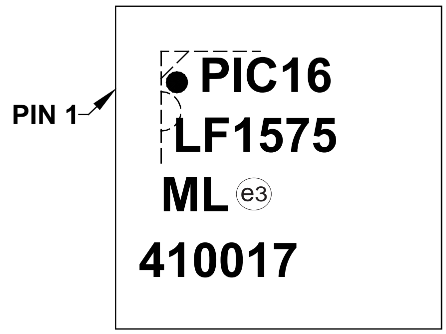
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Package Marking Information (Continued)

16-Lead UQFN (4x4x0.5mm)



Example

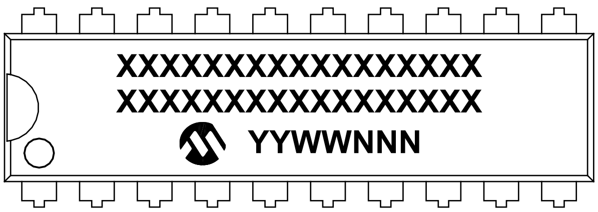


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

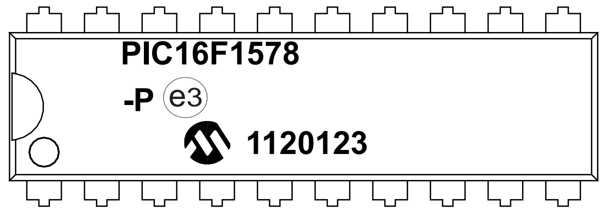
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Package Marking Information (Continued)

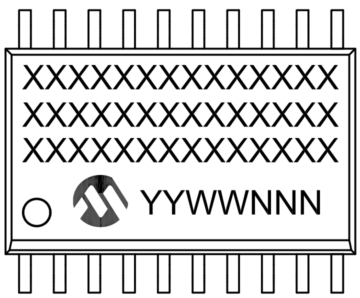
20-Lead PDIP (300 mil)



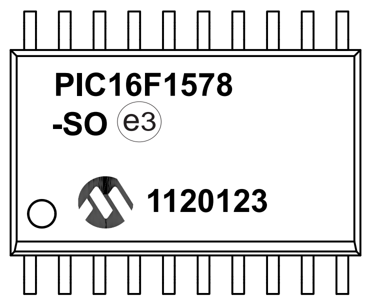
Example



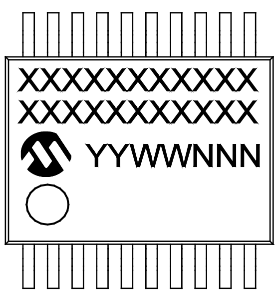
20-Lead SOIC (7.50 mm)



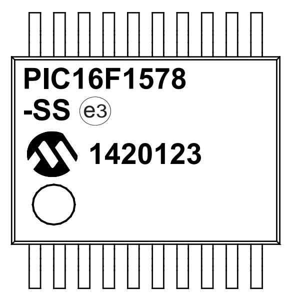
Example



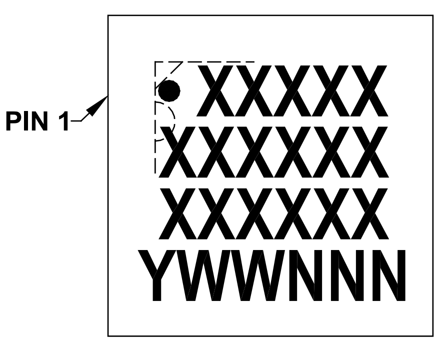
20-Lead SSOP (5.30 mm)



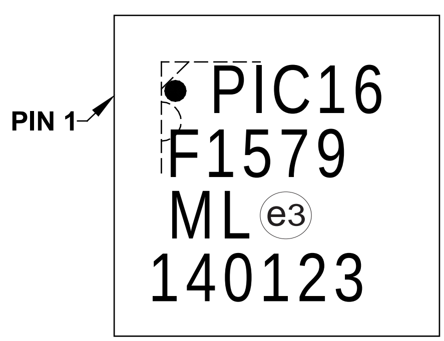
Example



20-Lead UQFN (4x4x0.5 mm)



Example

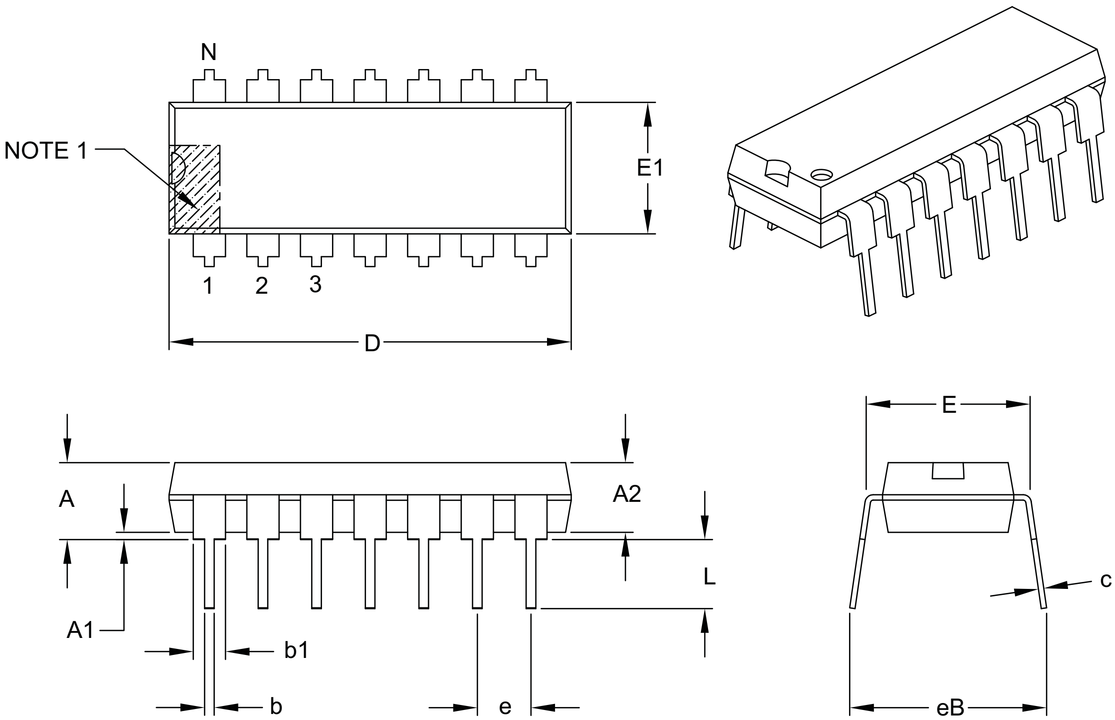


30.2 Package Details

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		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.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	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

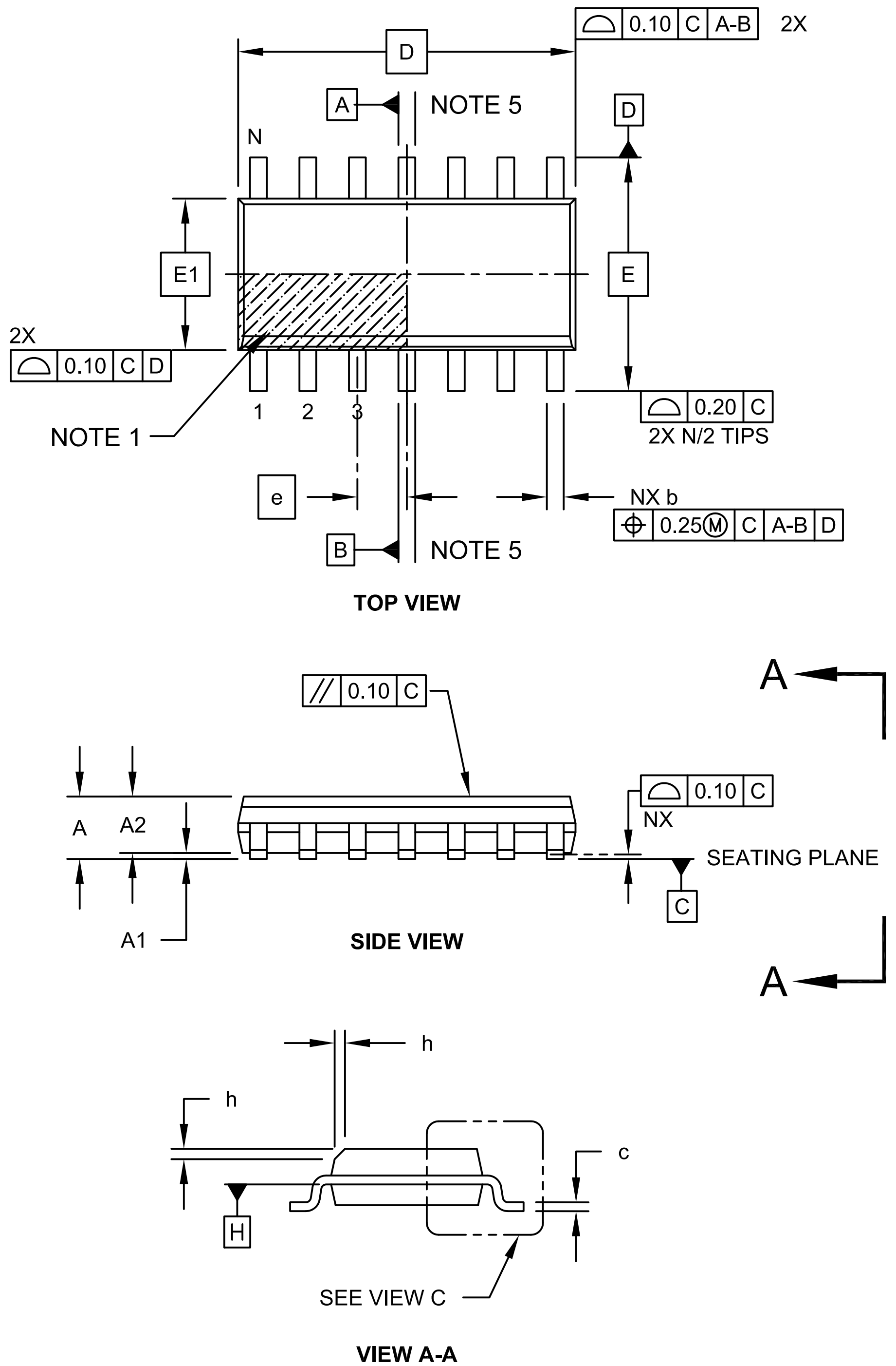
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   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]

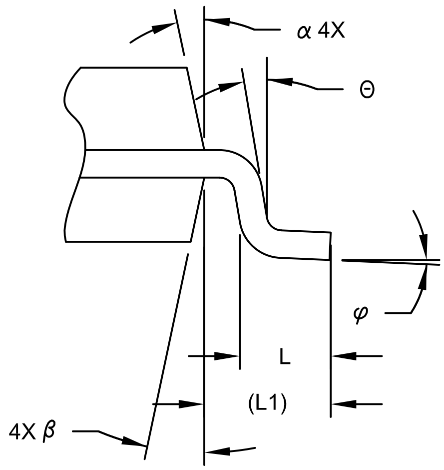
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



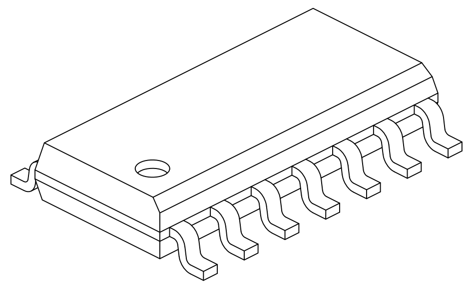
Microchip Technology Drawing No. C04-065C Sheet 1 of 2

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	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	c	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°

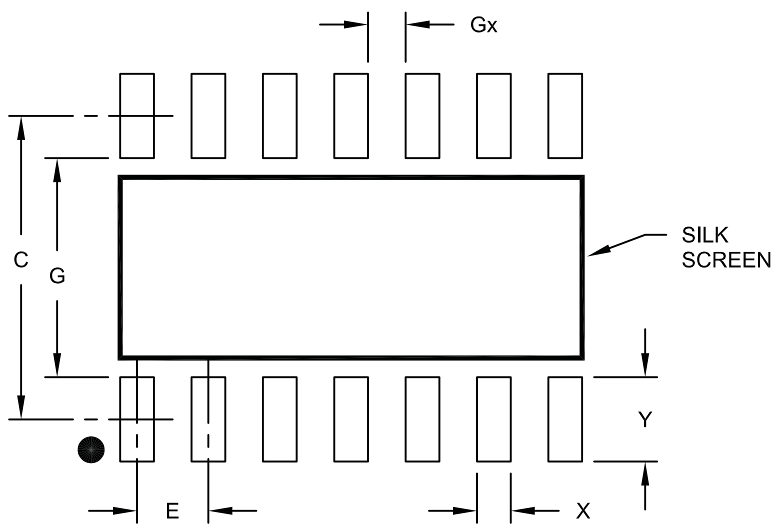
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
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.
4. 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 Limits	MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		5.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			1.50
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	3.90		

Notes:

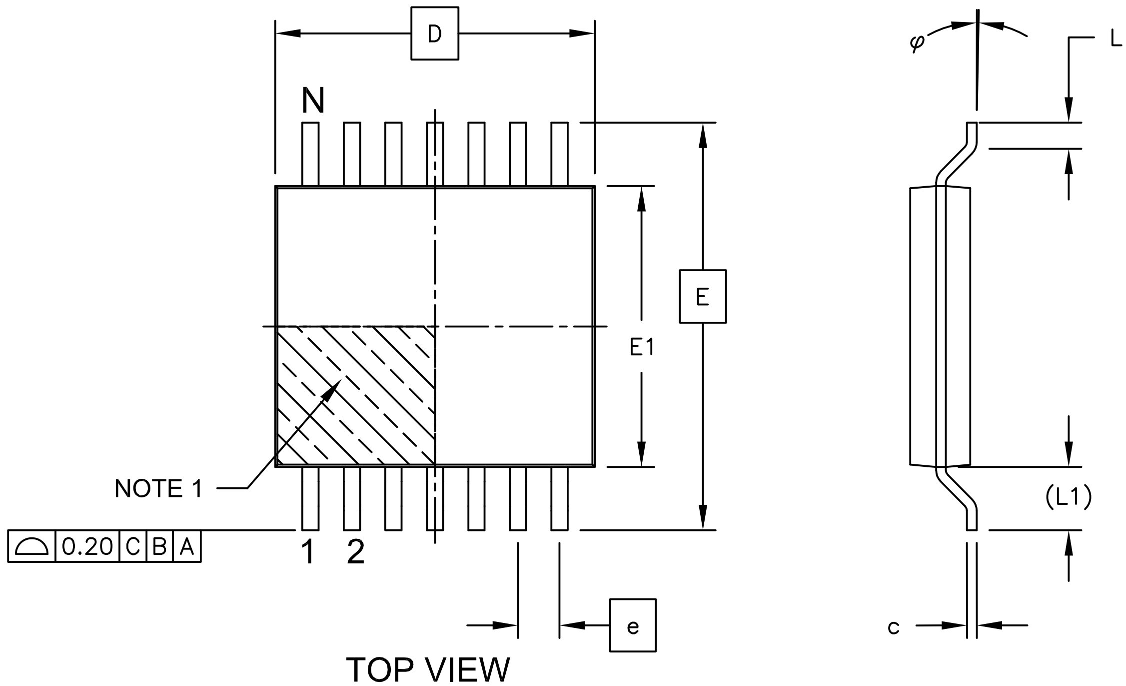
1. Dimensioning and tolerancing per ASME Y14.5M

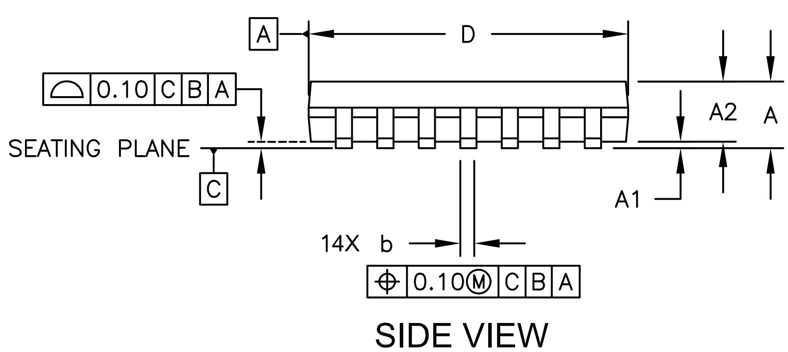
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

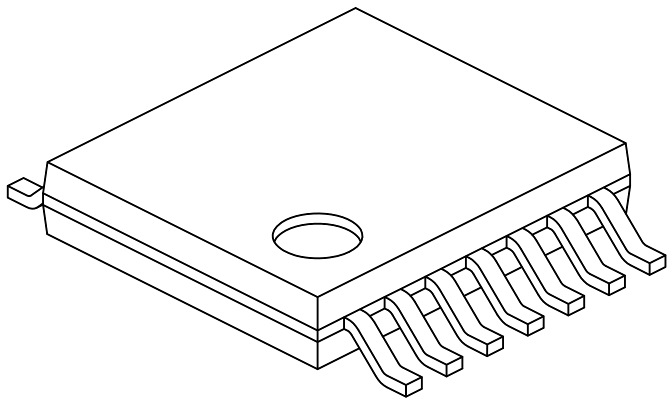




Microchip Technology Drawing C04-087C Sheet 1 of 2

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		0.65 BSC	
Overall Height	A	-	-	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	c	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.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

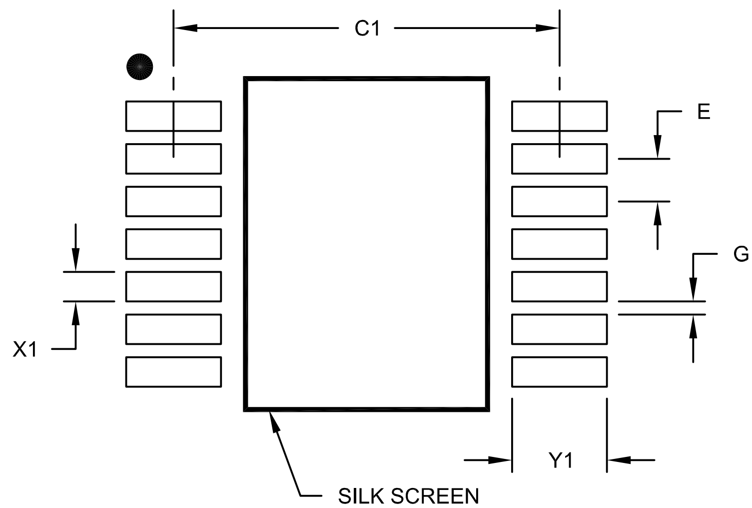
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 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



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:

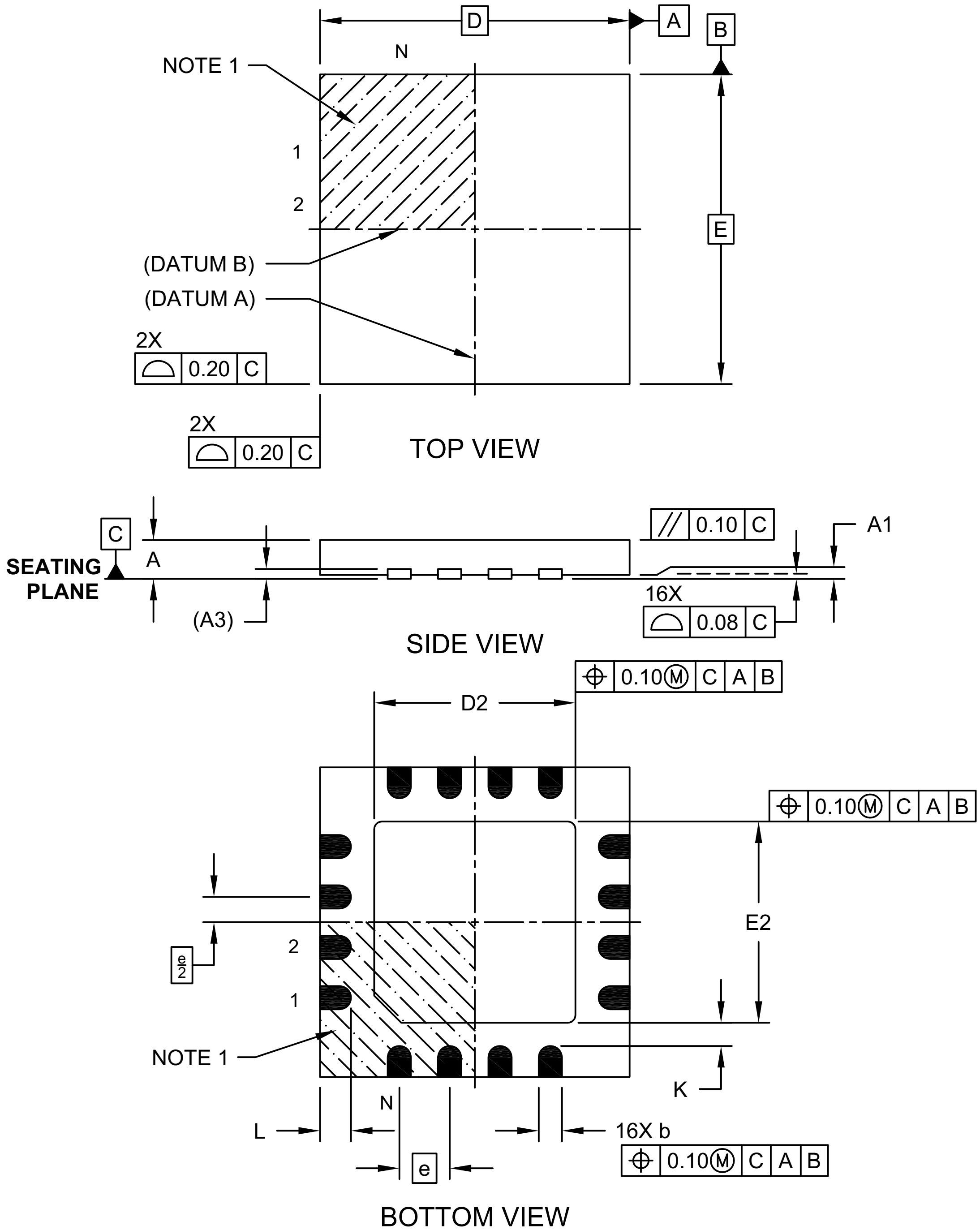
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

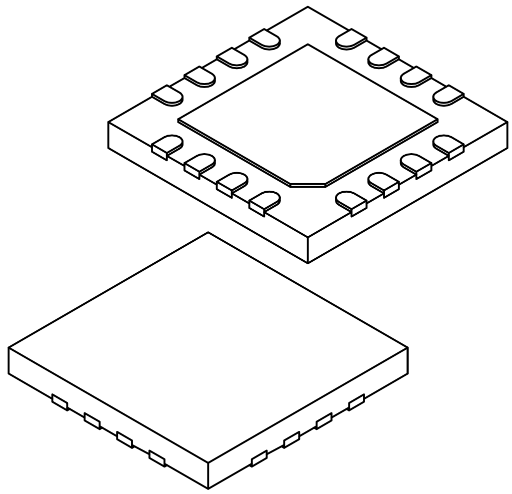
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-257A Sheet 1 of 2

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		16	
Pitch	e		0.65 BSC	
Overall Height	A	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	2.50	2.60	2.70
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.50	2.60	2.70
Terminal Width	b	0.25	0.30	0.35
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

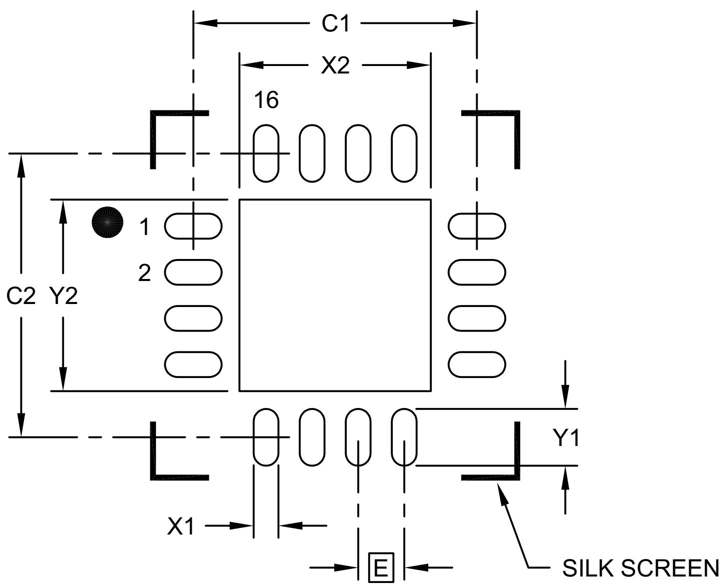
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-257A Sheet 2 of 2

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 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	E		0.65 BSC	
Optional Center Pad Width	X2			2.70
Optional Center Pad Length	Y2			2.70
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X16)	X1			0.35
Contact Pad Length (X16)	Y1			0.80

Notes:

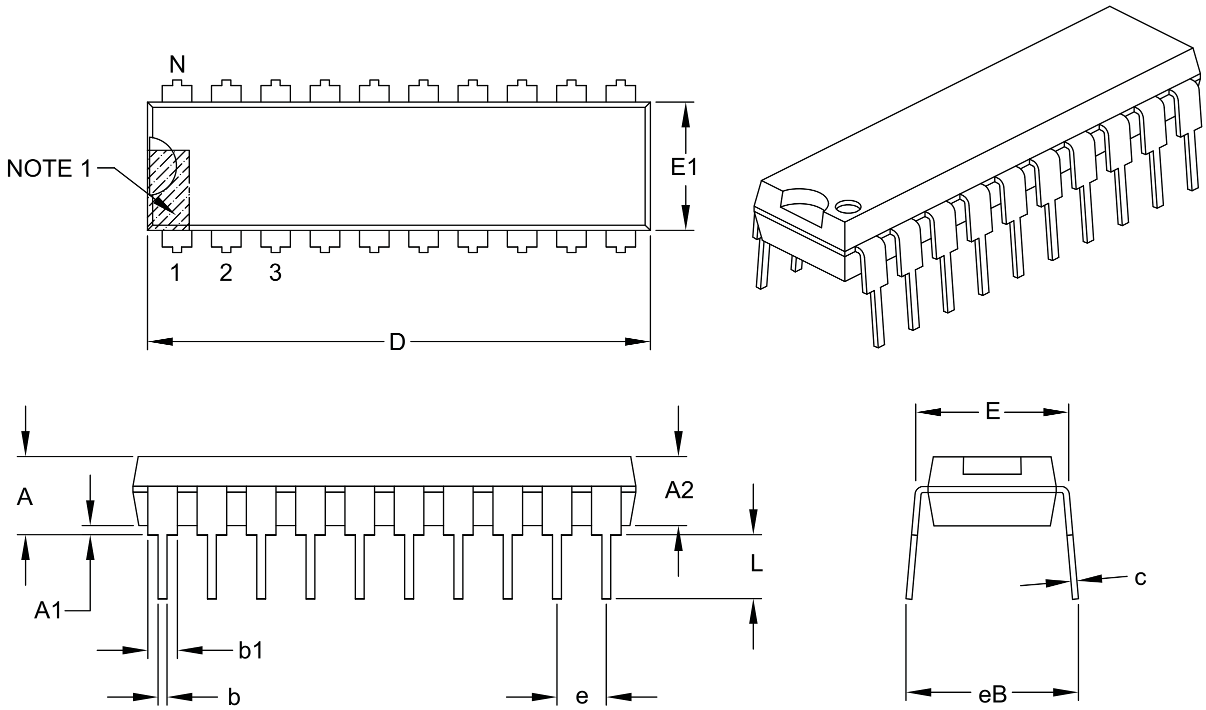
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2257A

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



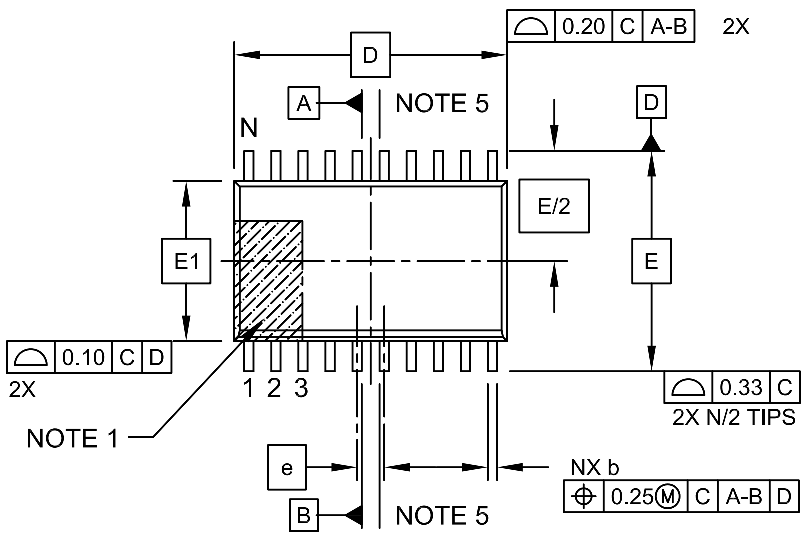
Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.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.

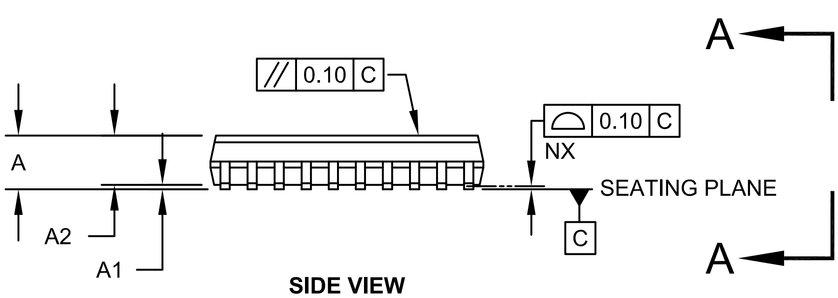
Microchip Technology Drawing C04-019B

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

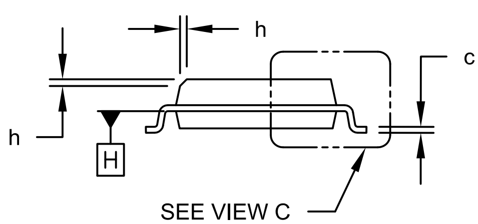
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

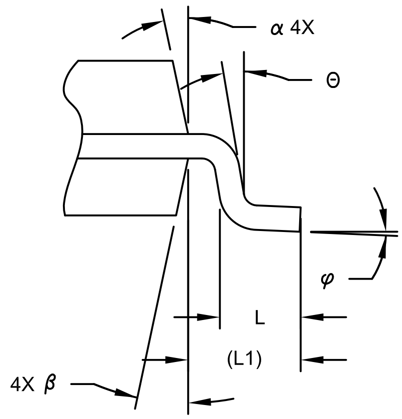


VIEW A-A

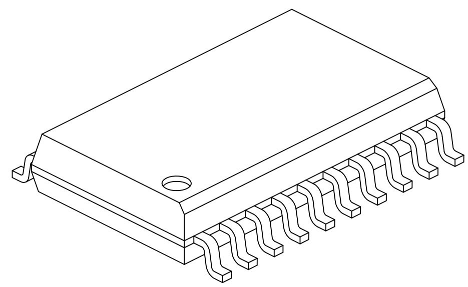
Microchip Technology Drawing C04-094C Sheet 1 of 2

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 Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	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	c	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°

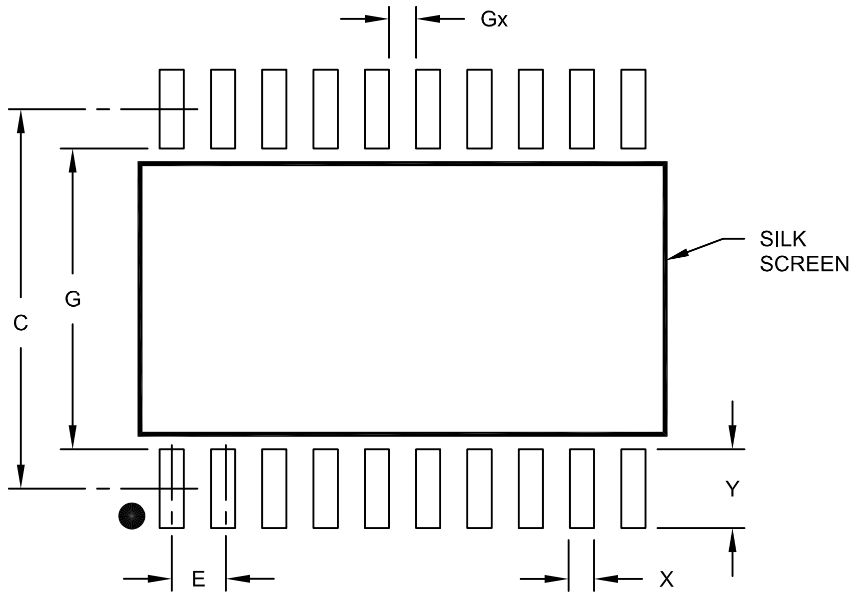
Notes:

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.
4. 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 Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		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:

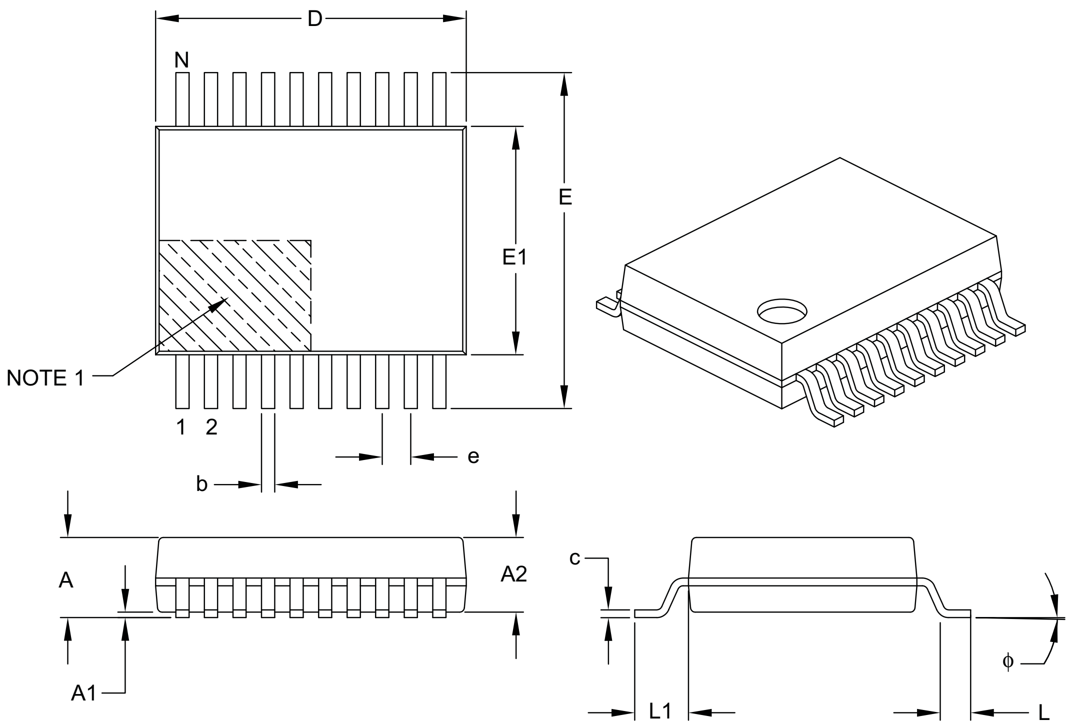
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

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
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	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.

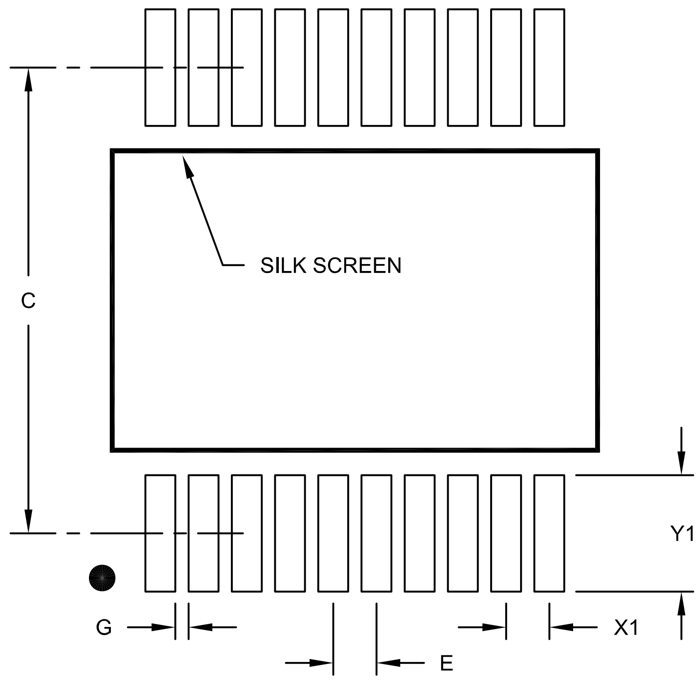
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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



RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

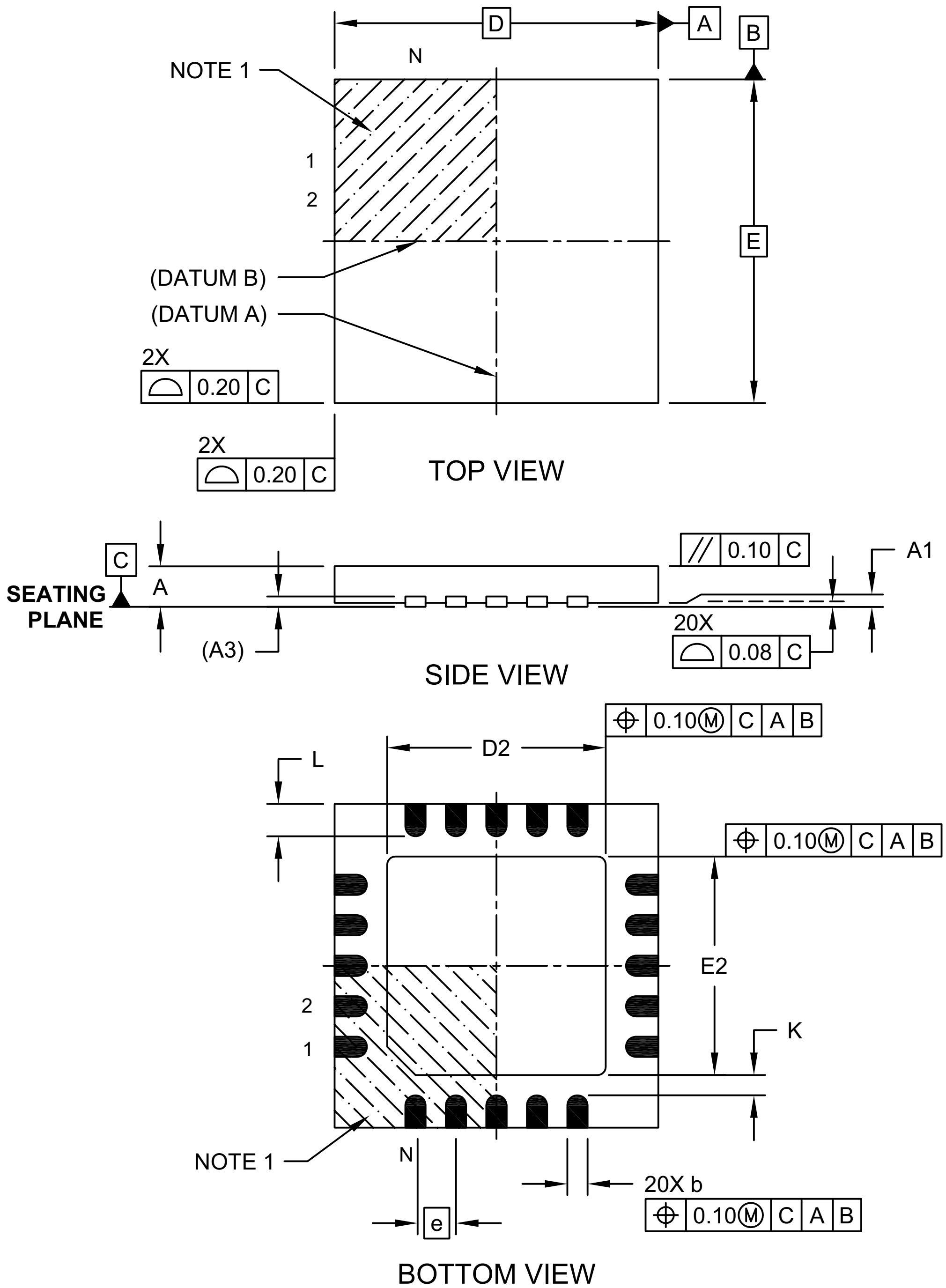
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

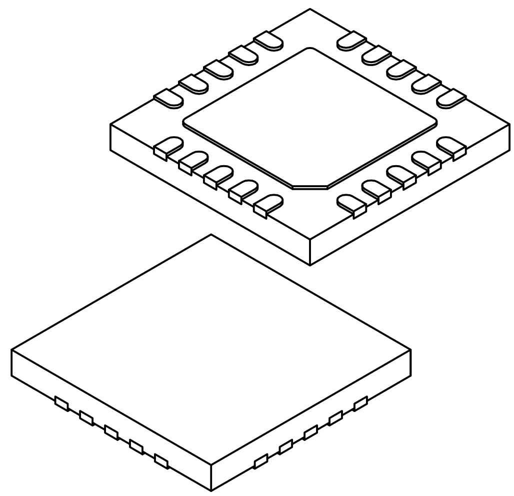
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



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



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Terminals	N		20	
Pitch	e		0.50 BSC	
Overall Height	A	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

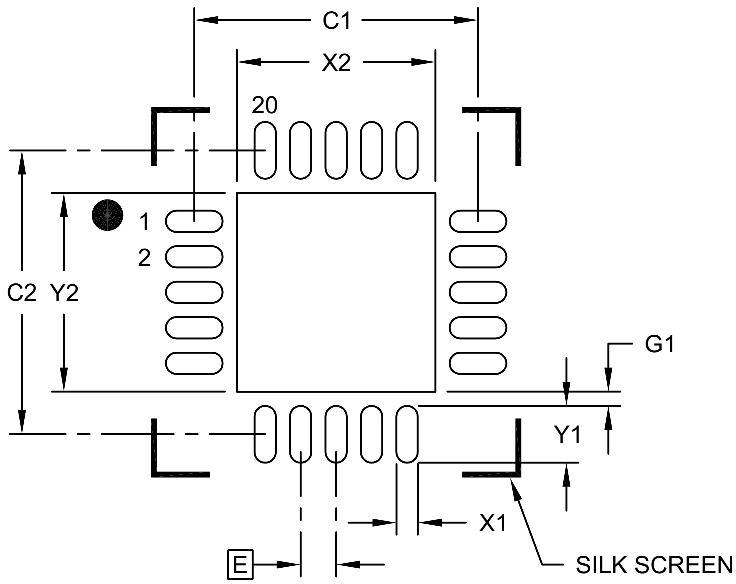
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   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	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 (2/2015)

Original release of this document.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site 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 web site 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)]	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC16LF1574, PIC16F1574, PIC16LF1575, PIC16F1575 PIC16LF1578, PIC16F1578, PIC16LF1579, PIC16F1579
Tape and Reel Option:	Blank = Standard packaging (tube or tray) T = Tape and Reel[(1)]
Temperature Range:	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)
Package:[(2)]	GZ = UQFN, 20-Lead (4x4x0.5mm) JQ = UQFN, 16-Lead (4x4x0.5mm) P = Plastic DIP SL = SOIC, 14-Lead SO = SOIC, 20-Lead SS = SSOP, 20-Lead ST = TSSOP, 14-Lead
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC16LF1578T - I/SO
Tape and Reel,
Industrial temperature,
SOIC package

b) PIC16F1575 - I/P
Industrial temperature
PDIP package

c) PIC16LF1574-E/JQ
Extended Temperature
UQFN Package

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.

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

NOTES:

ISBN: 978-1-63277-029-5

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

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, KEELOQ® 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.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983
Indianapolis

Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

01/27/15