PIC12(L)F1612/16(L)F1613

8/14-Pin, 8-Bit Flash Microcontroller

Description

PIC12(L)F1612/16(L)F1613 microcontrollers deliver on-chip features that are unique to the design for embedded control of small motors and general purpose applications in 8/14-pin count packages. Features like 10-bit A/D, CCP, 24-bit SMT and Zero-Cross Detection offer an excellent solution to the variety of applications. The product family also has a CRC+ memory scan and Windowed WDT to support safety-critical systems in home appliances, white goods and other end equipment.

Core Features

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

Memory

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

Operating Characteristics

- Operating Voltage Range:
    - 1.8V to 3.6V (PIC12LF1612/16F1613)
    - 2.3V to 5.5V (PIC12F1612/16F1613)
- Temperature Range:
    - Industrial: -40°C to 85°C
    - Extended: -40°C to 125°C

eXtreme Low-Power (XLP) Features

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

Digital Peripherals

- Complementary Waveform Generator (CWG):
    - Rising and falling edge dead-band control
    - Full-bridge, half-bridge, 1-channel drive
    - Multiple signal sources
- Two Capture/Compare/PWM (CCP) modules
- Two Signal Measurement Timers (SMT):
    - 24-bit timer/counter with prescaler
    - Multiple gate and clock inputs
- 8-Bit Timers (TMR2+HLT/4/6):
    - Up to 3 Timer2/4/6 with Hardware Limit Timer (HLT)
    - Monitors Fault Conditions: Stall, Stop, etc.
    - Multiple modes
    - 8-bit timer/counter with prescaler
    - 8-bit period register and postscaler
    - Asynchronous H/W Reset sources
- Cyclic Redundancy Check with Memory Scan (CRC/SCAN):
    - Software configurable

- Up to 11 I/O Pins and One Input-only Pin:
  - Individually programmable pull-ups
  - Slew rate control
  - Interrupt-on-change with edge-select

Intelligent Analog Peripherals

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

Clocking Structure

- 16 MHz Internal Oscillator:
  - ±1% at calibration
  - Selectable frequency range from 32 MHz to 31 kHz
- 31 kHz Low-Power Internal Oscillator
- 4x Phase-Locked Loop (PLL):
  - For up to 32 MHz internal operation
- External Oscillator Block with:
  - Three external clock modes up to 32 MHz

TABLE 1: PIC12/16(L)F161X FAMILY TYPES

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

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

Data Sheet Index:

A.	DS40001737	PIC12(L)F1612/16(L)F1613 Data Sheet, 8/14-Pin, 8-bit Flash Microcontrollers
B.	DS40001769	PIC16(L)F1614/8 Data Sheet, 14/20-Pin, 8-bit Flash Microcontrollers
C.	DS40001770	PIC16(L)F1615/9 Data Sheet, 14/20-Pin, 8-bit Flash Microcontrollers

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

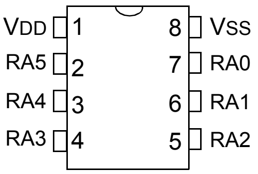
TABLE 2: PACKAGES

Packages	PDIP	SOIC	DFN	UDFN	TSSOP	QFN	UQFN	SSOP
PIC12(L)F1612	●	●	●	●				
PIC16(L)F1613	●	●			●	●	●	

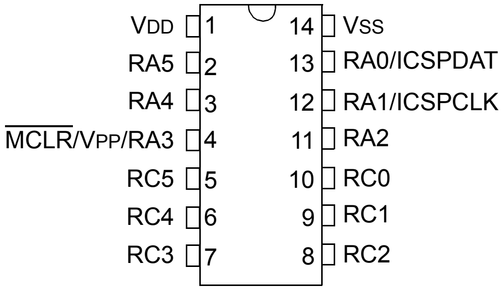
Note: Pin details are subject to change.

PIN DIAGRAMS

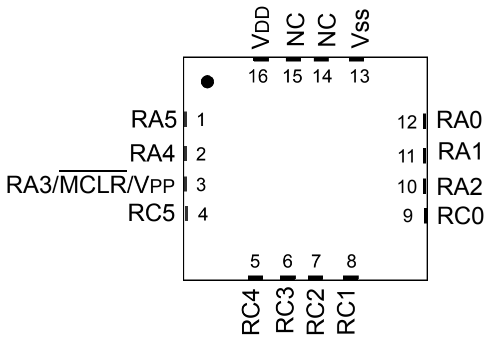
8-pin PDIP, SOIC, DFN, UDFN



14-pin PDIP, SOIC, TSSOP



16-pin QFN, UQFN



PIN ALLOCATION TABLES

TABLE 3: 8-PIN ALLOCATION TABLE (PIC12(L)F1612)

I/O	8-Pin PDIP, SOIC, DFN, UDFN	A/D	Reference	Comparator	Timers	CCP	CWG	ZCD	Interrupt	SMT	Pull-up	Basic
RA0	7	AN0	DAC1OUT1	C1IN+	—	CCP2	CWG1B	—	IOC	—	Y	ICSPDAT
RA1	6	AN1	VREF+	C1IN0-	—	—	—	ZCD1OUT	IOC	—	Y	ICSPCLK
RA2	5	AN2	—	C1OUT	T0CKI	CCP1	CWG1A CWG1IN	ZCD1IN	INT IOC	SMTSIG2	Y	—
RA3	4	—	—	—	T1G(1) T6IN	—	—	—	IOC	SMTWIN2	Y	$\overline{\text{MCLR}}$/VPP
RA4	3	AN3	—	C1IN1-	T1G	—	CWG1B(1)	—	IOC	SMTSIG1	Y	CLKOUT
RA5	2	—	—	—	T1CKI T2IN	CCP1(1)	CWG1A(1)	—	IOC	SMTWIN1	Y	CLKIN
VDD	1	—	—	—	—	—	—	—	—	—	—	VDD
VSS	8	—	—	—	—	—	—	—	—	—	—	VSS

Note 1: Alternate pin function selected with the APFCON register.

TABLE 4: 14/16-PIN ALLOCATION TABLE (PIC16(L)F1613)

I/O	14-Pin PDIP, SOIC, TSSOP	16-Pin QFN, UQFN	A/D	Reference	Comparator	Timers	CCP	CWG	ZCD	Interrupt	SMT	Pull-up	Basic
RA0	13	12	AN0	DAC1OUT1	C1IN+	—	—	—	—	IOC	—	Y	ICSPDAT
RA1	12	11	AN1	VREF+	C1IN0- C2IN0-	—	—	—	ZCD1OUT	IOC	—	Y	ICSPCLK
RA2	11	10	AN2	—	C1OUT	T0CKI T4IN	—	CWG1IN	ZCD1IN	INT IOC	—	Y	—
RA3	4	3	—	—	—	T1G(1) T6IN	—	—	—	IOC	SMTWIN2	Y	$\overline{\text{MCLR}}$/VPP
RA4	3	2	AN3	—	—	T1G	—	—	—	IOC	SMTSIG1	Y	CLKOUT
RA5	2	1	—	—	—	T1CKI T2IN	CCP2(1)	—	—	IOC	SMTWIN1	Y	CLKIN
RC0	10	9	AN4	—	C2IN+	—	—	—	—	IOC	—	Y	—
RC1	9	8	AN5	—	C1IN1- C2IN1-	T4IN	—	—	—	IOC	SMTSIG2	Y	—
RC2	8	7	AN6	—	C1IN2- C2IN2-	—	—	CWG1D	—	IOC	—	Y	—
RC3	7	6	AN7	—	C1IN3- C2IN3-	—	CCP2	CWG1C	—	IOC	—	Y	—
RC4	6	5	—	—	C2OUT	—	—	CWG1B	—	IOC	—	Y	—
RC5	5	4	—	—	—	—	CCP1	CWG1A	—	IOC	—	Y	—
VDD	1	16	—	—	—	—	—	—	—	—	—	—	VDD
VSS	14	13	—	—	—	—	—	—	—	—	—	—	VSS

Note 1: Alternate pin function selected with the APFCON register.

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** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

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

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

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

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

- Microchip’s Worldwide Website; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

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

Customer Notification System

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

1.0 DEVICE OVERVIEW

The PIC12(L)F1612/16(L)F1613 are described within this data sheet. The block diagram of these devices are shown in Figure 1-1, the available peripherals are shown in Table 1-1, and the pin out descriptions are shown in Tables 1-2 and 1-3.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC12(L)F1612	PIC16(L)F1613
Analog-to-Digital Converter (ADC)		●	●
Complementary Wave Generator (CWG)		●	●
Cyclic Redundancy Check (CRC)		●	●
Digital-to-Analog Converter (DAC)		●	●
Fixed Voltage Reference (FVR)		●	●
Temperature Indicator		●	●
Windowed Watchdog Timer (WDT)		●	●
Zero Cross Detection (ZCD)		●	●
Capture/Compare/PWM (CCP) Modules			
	CCP1	●	●
	CCP2	●	●
Comparators			
	C1	●	●
	C2		●
Signal Measurement Timer (SMT)			
	SMT1	●	●
	SMT2	●	●
Timers			
	Timer0	●	●
	Timer1	●	●
	Timer2	●	●
	Timer4	●	●
	Timer6	●	●

1.1 Register and Bit Naming Conventions

1.1.1 REGISTER NAMES

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

1.1.2 BIT NAMES

There are two variants for bit names:

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

1.1.2.1 Short Bit Names

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

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

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

1.1.2.2 Long Bit Names

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

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

1.1.2.3 Bit Fields

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

```
COG1CON0bits.MD = 0x5;
```

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

Example 1:

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

Example 2:

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

1.1.3 REGISTER AND BIT NAMING EXCEPTIONS

1.1.3.1 Status, Interrupt, and Mirror Bits

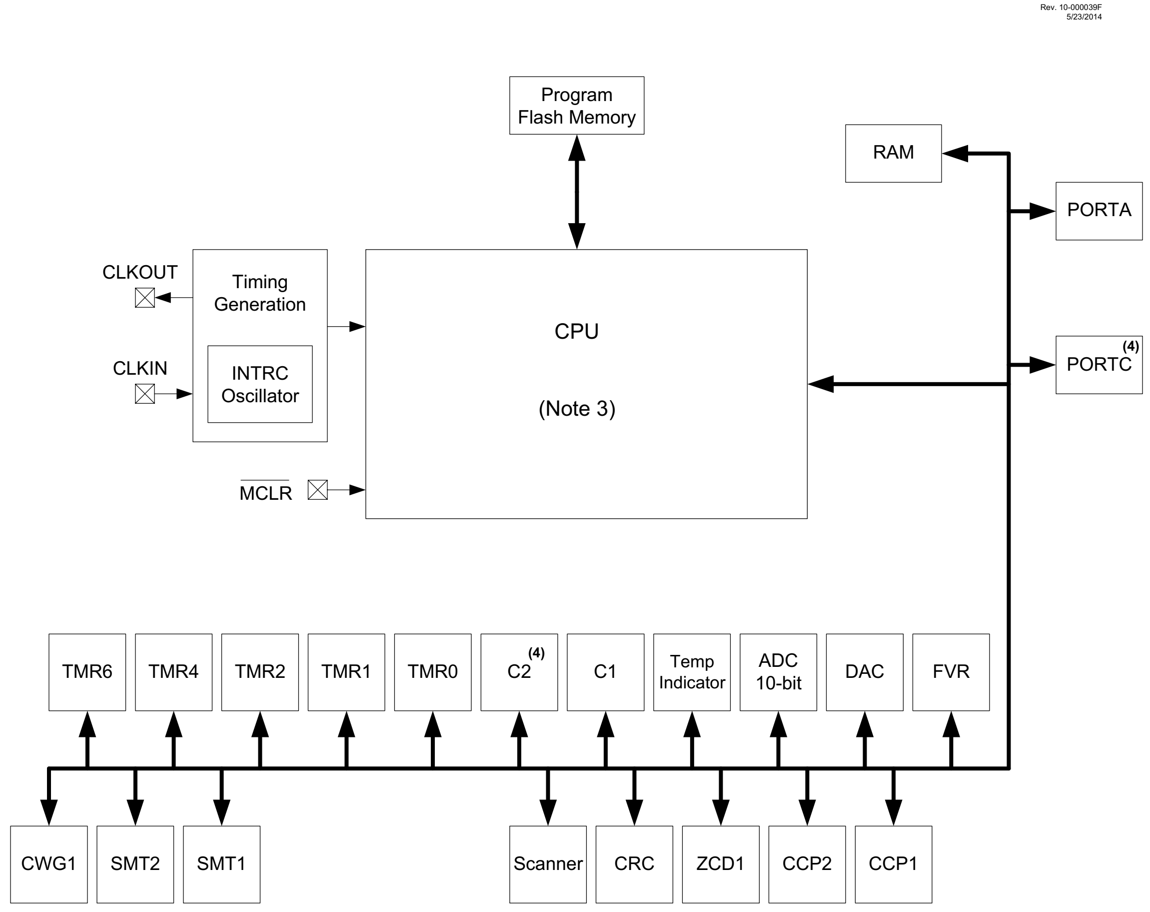
Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

1.1.3.2 Legacy Peripherals

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

- EUSART
- MSSP

FIGURE 1-1: PIC12(L)F1612/16(L)F1613 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)F1613 only.

TABLE 1-2: PIC12(L)F1612 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT1/ CCP2/CWG1B(1)/ ICSPDAT	RA0	TTL/ST	CMOS/OD	General purpose I/O.
	AN0	AN	—	ADC Channel input.
	C1IN+	AN	—	Comparator positive input.
	DAC1OUT1	—	AN	Digital-to-Analog Converter output.
	CCP2	TTL/ST	CMOS/OD	Capture/Compare/PWM2.
	CWG1B	—	CMOS/OD	CWG complementary output B.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/AN1/VREF+/C1IN0-/ ZCD1OUT/ICSPCLK	RA1	TTL/ST	CMOS/OD	General purpose I/O.
	AN1	AN	—	ADC Channel input.
	VREF+	AN	—	Voltage Reference input.
	C1IN0-	AN	—	Comparator negative input.
	ZCD1OUT	—	CMOS	Zero-Cross Detect output.
	ICSPCLK	ST	—	ICSP Programming Clock.
RA2/AN2/C1OUT/T0CKI/T4IN/ CCP1(1)/CWG1A(1)/ CWG1IN/ZCD1IN/INT/SMTSIG2	RA2	TTL/ST	CMOS/OD	General purpose I/O.
	AN2	AN	—	ADC Channel input.
	C1OUT	—	CMOS/OD	Comparator output.
	T0CKI	TTL/ST	—	Timer0 clock input.
	T4IN	TTL/ST	—	Timer4 input.
	CCP1	TTL/ST	CMOS/OD	Capture/Compare/PWM1.
	CWG1A	—	CMOS/OD	CWG complementary output A.
	CWG1IN	TTL/ST	—	CWG complementary input.
	ZCD1IN	AN	—	Zero-Cross Detect input.
	INT	TTL/ST	—	External interrupt.
	SMTSIG2	TTL/ST	—	SMT2 signal input.
RA3/VPP/T1G(1)/T6IN/ SMTWIN2/$\overline{\text{MCLR}}$	RA3	TTL/ST	—	General purpose input with IOC and WPU.
	VPP	HV	—	Programming voltage.
	T1G	TTL/ST	—	Timer1 Gate input.
	T6IN	TTL/ST	—	Timer6 input.
	SMTWIN2	TTL/ST	—	SMT2 window input.
	$\overline{\text{MCLR}}$	TTL/ST	—	Master Clear with internal pull-up.
RA4/AN3/C1IN1-/T1G(1)/ CWG1B(1)/SMTSIG1/ CLKOUT	RA4	TTL/ST	CMOS/OD	General purpose I/O.
	AN3	AN	—	ADC Channel input.
	C1IN1-	AN	—	Comparator negative input.
	T1G	TTL/ST	—	Timer1 Gate input.
	CWG1B	—	CMOS/OD	CWG complementary output B.
	SMTSIG1	TTL/ST	—	SMT1 signal input.
	CLKOUT	—	CMOS	FOSC/4 output.

Legend:
AN = Analog input or output
TTL = TTL compatible input
HV = High Voltage
CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
XTAL = Crystal
OD = Open-Drain
I^2C = Schmitt Trigger input with I^2C levels

Note 1: Alternate pin function selected with the APFCON register (Register 12-1).

TABLE 1-2: PIC12(L)F1612 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RA5/CLKIN/T1CKI/T2IN/CCP1[(1)]/CWG1A[(1)]/SMTWIN1	RA5	TTL/ST	CMOS/OD	General purpose I/O.
	CLKIN	CMOS	—	External clock input (EC mode).
	T1CKI	TTL/ST	—	Timer1 clock input.
	T2IN	TTL/ST	—	Timer2 input.
	CCP1	TTL/ST	CMOS/OD	Capture/Compare/PWM1.
	CWG1A	—	CMOS/OD	CWG complementary output A.
	SMTWIN1	TTL/ST	—	SMT1 window input.
VDD	VDD	Power	—	Positive supply.
VSS	VSS	Power	—	Ground reference.

Legend:

AN = Analog input or output
TTL = TTL compatible input
HV = High Voltage
CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
XTAL = Crystal
OD = Open-Drain
I^2C = Schmitt Trigger input with I^2C levels

Note 1: Alternate pin function selected with the APFCON register (Register 12-1).

TABLE 1-3: PIC16(L)F1613 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT1/ ICSPDAT	RA0	TTL/ST	CMOS/OD	General purpose I/O.
	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-/ ZCD1OUT/ICSPCLK	RA1	TTL/ST	CMOS/OD	General purpose I/O.
	AN1	AN	—	ADC Channel input.
	VREF+	AN	—	Voltage Reference input.
	C1IN0-	AN	—	Comparator negative input.
	C2IN0-	AN	—	Comparator negative input.
	ZCD1OUT	—	CMOS	Zero-Cross Detect output.
	ICSPCLK	ST	—	ICSP Programming Clock.
RA2/AN2/C1OUT/T0CKI/ CWG1IN/ZCD1IN/INT	RA2	TTL/ST	CMOS/OD	General purpose I/O.
	AN2	AN	—	ADC Channel input.
	C1OUT	—	CMOS/OD	Comparator output.
	T0CKI	TTL/ST	—	Timer0 clock input.
	CWG1IN	TTL/ST	—	CWG complementary input.
	ZCD1IN	AN	—	Zero-Cross Detect input.
	INT	TTL/ST	—	External interrupt.
RA3/VPP/T1G(1)/T6IN/ SMTWIN2/$\overline{\text{MCLR}}$	RA3	TTL/ST	—	General purpose input with IOC and WPU.
	VPP	HV	—	Programming voltage.
	T1G	TTL/ST	—	Timer1 Gate input.
	T6IN	TTL/ST	—	Timer6 input.
	SMTWIN2	TTL/ST	—	SMT2 window input.
	$\overline{\text{MCLR}}$	TTL/ST	—	Master Clear with internal pull-up.
RA4/AN3/T1G(1)/SMTSIG1/ CLKOUT	RA4	TTL/ST	CMOS/OD	General purpose I/O.
	AN3	AN	—	ADC Channel input.
	T1G	TTL/ST	—	Timer1 Gate input.
	SMTSIG1	TTL/ST	—	SMT1 signal input.
	CLKOUT	—	CMOS	FOSC/4 output.
RA5/CLKIN/T1CKI/T2IN/ CCP2(1)/SMTWIN1	RA5	TTL/ST	CMOS/OD	General purpose I/O.
	CLKIN	CMOS	—	External clock input (EC mode).
	T1CKI	TTL/ST	—	Timer1 clock input.
	T2IN	TTL/ST	—	Timer2 input.
	CCP2	TTL/ST	CMOS/OD	Capture/Compare/PWM2.
	SMTWIN1	TTL/ST	—	SMT1 window input.
RC0/AN4/C2IN+	RC0	TTL/ST	CMOS/OD	General purpose I/O.
	AN4	AN	—	ADC Channel input.
	C2IN+	AN	—	Comparator positive input.

Legend:
AN = Analog input or output
TTL = TTL compatible input
HV = High Voltage
CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
XTAL = Crystal
OD = Open-Drain
I^2C = Schmitt Trigger input with I^2C levels

Note 1: Alternate pin function selected with the APFCON register (Register 12-1).

TABLE 1-3: PIC16(L)F1613 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC1/AN5/C1IN1-/C2IN1-/T4IN/SMTSIG2	RC1	TTL/ST	CMOS/OD	General purpose I/O.
	AN5	AN	—	ADC Channel input.
	C1IN1-	AN	—	Comparator negative input.
	C2IN1-	AN	—	Comparator negative input.
	T4IN	TTL/ST	—	Timer4 input.
	SMTSIG2	TTL/ST	—	SMT2 signal input.
RC2/AN6/C1IN2-/C2IN2-/CWG1D	RC2	TTL/ST	CMOS/OD	General purpose I/O.
	AN6	AN	—	ADC Channel input.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
	CWG1D	—	CMOS/OD	CWG complementary output D.
RC3/AN7/C1IN3-/C2IN3-/CCP2(1)/CWG1C	RC3	TTL/ST	—	General purpose input with IOC and WPU.
	AN7	AN	—	ADC Channel input.
	C1IN3-	AN	—	Comparator negative input.
	C2IN3-	AN	—	Comparator negative input.
	CCP2	TTL/ST	CMOS/OD	Capture/Compare/PWM2.
	CWG1C	—	CMOS/OD	CWG complementary output C.
RC4/C2OUT/CWG1B	RC4	TTL/ST	CMOS/OD	General purpose I/O.
	C2OUT	—	CMOS/OD	Comparator output.
	CWG1B	—	CMOS/OD	CWG complementary output B.
RC5/CCP1/CWG1A	RC5	TTL/ST	CMOS/OD	General purpose I/O.
	CCP1	TTL/ST	CMOS/OD	Capture/Compare/PWM1.
	CWG1A	—	CMOS/OD	CWG complementary output A.
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^{2}C = Schmitt Trigger input with I^{2}C levels
HV = High Voltage
XTAL = Crystal

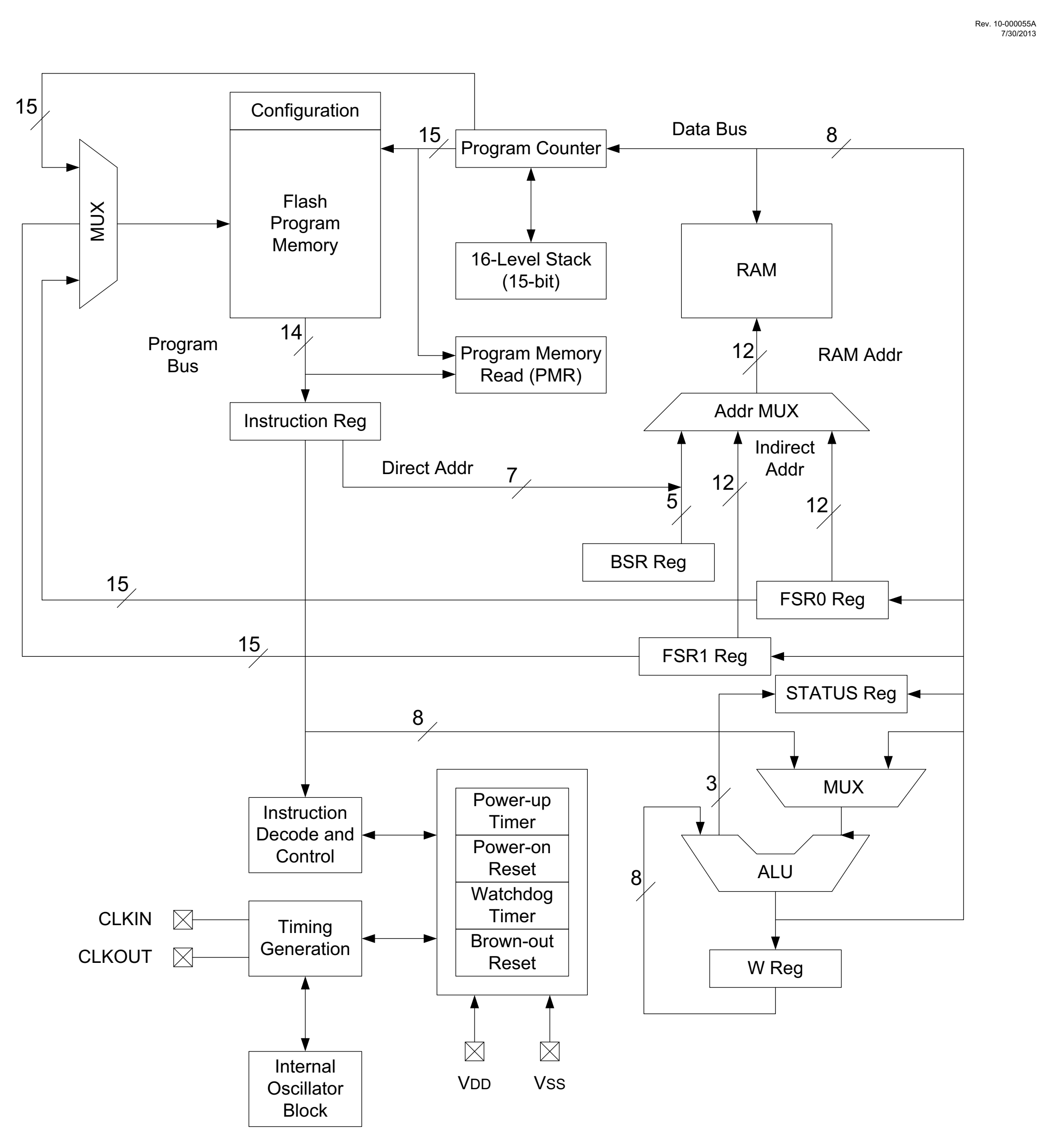
Note 1: Alternate pin function selected with the APFCON register (Register 12-1).

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 27.0 “Instruction Set Summary”** for more details.

3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

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

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

- PCL and PCLATH
- Stack
- Indirect Addressing

3.1 Program Memory Organization

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

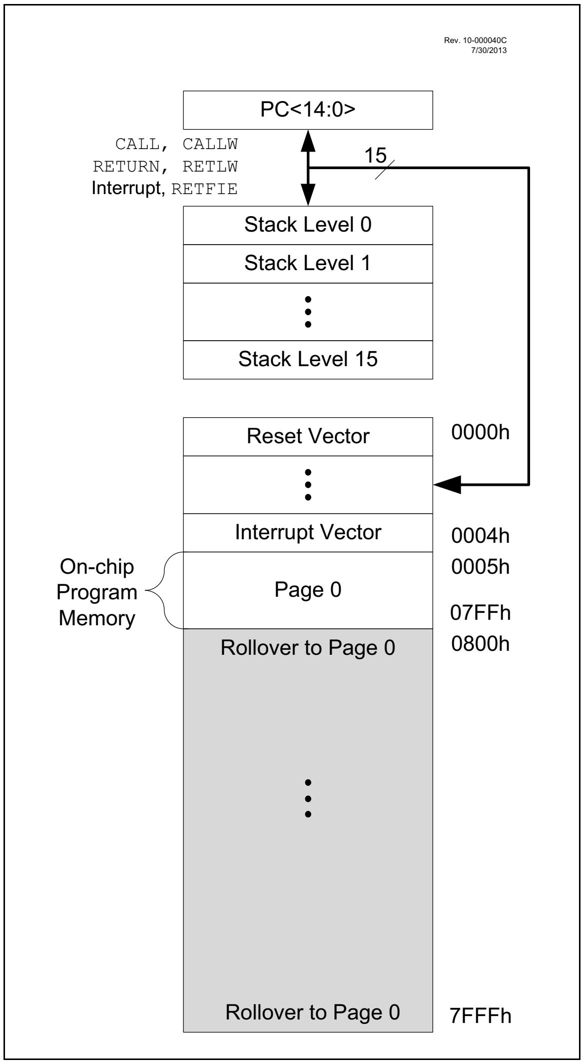
3.2 High-Endurance Flash

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

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range(1)
PIC12(L)F1612/16(L)F1613	2,048	07FFh	0780h-07FFh

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

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC12(L)F1612/16(L)F1613



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
  DW DATA0             ;First constant
  DW DATA1             ;Second constant
  DW DATA2
  DW DATA3
my_function
  ;… LOTS OF CODE…
  MOVLW   DATA_INDEX
  ADDLW   LOW constants
  MOVWF   FSR1L
  MOVLW   HIGH constants;MSb sets
                        automatically
  MOVWF   FSR1H
  BTFSC   STATUS, C    ;carry from ADDLW?
  INCF    FSR1h, f     ;yes
  MOVIW   0[FSR1]
;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 3-2):

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

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as ‘`0`’. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.6 “Indirect Addressing”** for more information.

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

3.3.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x80h through x0Bh/x8Bh). These registers are listed below in Table 3-1. For detailed

TABLE 3-1: 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 27.0 “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 shown in Table 3-2 through Table 3-7.

FIGURE 3-2: BANKED MEMORY PARTITIONING

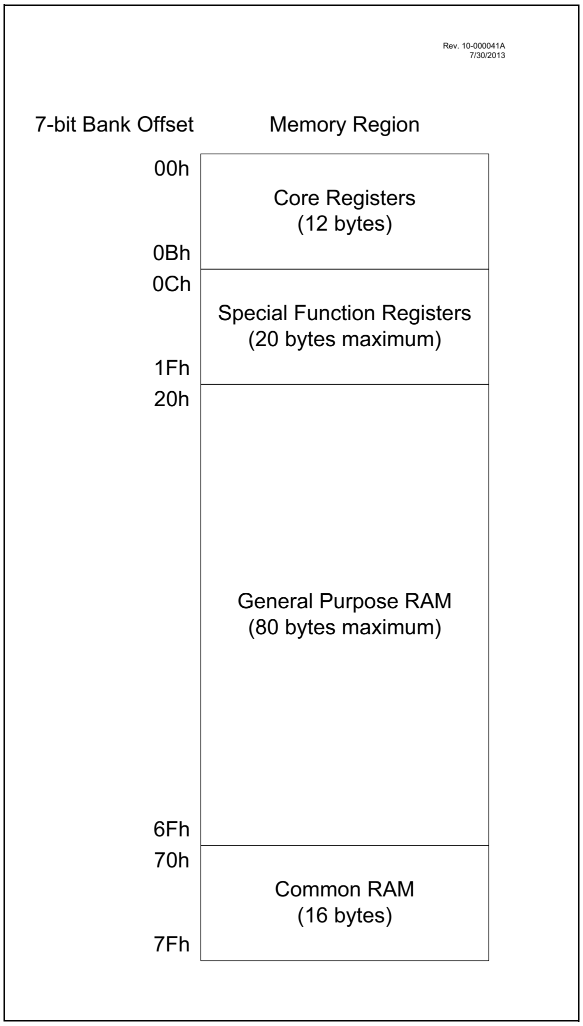


TABLE 3-2: PIC12(L)F1612 MEMORY MAP, BANK 0-7

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7
000h 00Bh	Core Registers (Table 3-1)	080h 08Bh	Core Registers (Table 3-1)	100h 10Bh	Core Registers (Table 3-1)	180h 18Bh	Core Registers (Table 3-1)	200h 20Bh	Core Registers (Table 3-1)	280h 28Bh	Core Registers (Table 3-1)	300h 30Bh	Core Registers (Table 3-1)	380h 38Bh	Core Registers (Table 3-1)
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	—	08Eh	—	10Eh	—	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
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	CCP1RL	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	CCP1RH	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	—	193h	PMDATL	213h	—	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	—	194h	PMDATH	214h	—	294h	CCP1CAP	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	—	117h	FVRCON	197h	VREGCON	217h	—	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCP2RL	318h	—	398h	—
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	—	219h	—	299h	CCP2RH	319h	—	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	—	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	—	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	—	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	APFCON	19Dh	—	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	—	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	—	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	Unimplemented Read as ‘`0`’	220h 26Fh	Unimplemented Read as ‘`0`’	2A0h 2EFh	Unimplemented Read as ‘`0`’	320h 36Fh	Unimplemented Read as ‘`0`’	3A0h 3EFh	Unimplemented Read as ‘`0`’
070h 07Fh	Common RAM	0F0h 0FFh	Common RAM (Accesses 70h – 7Fh)	170h 17Fh	Common RAM (Accesses 70h – 7Fh)	1F0h 1FFh	Common RAM (Accesses 70h – 7Fh)	270h 27Fh	Common RAM (Accesses 70h – 7Fh)	2F0h 2FFh	Common RAM (Accesses 70h – 7Fh)	370h 37Fh	Common RAM (Accesses 70h – 7Fh)	3F0h 3FFh	Common RAM (Accesses 70h – 7Fh)

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

TABLE 3-3: PIC16(L)F1613 MEMORY MAP, BANK 0-7

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7
000h 00Bh	Core Registers (Table 3-1)	080h 08Bh	Core Registers (Table 3-1)	100h 10Bh	Core Registers (Table 3-1)	180h 18Bh	Core Registers (Table 3-1)	200h 20Bh	Core Registers (Table 3-1)	280h 28Bh	Core Registers (Table 3-1)	300h 30Bh	Core Registers (Table 3-1)	380h 38Bh	Core Registers (Table 3-1)
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	CCPR1L	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	CCPR1H	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	—	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	CM2CON1	194h	PMDATH	214h	—	294h	CCP1CAP	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	—	117h	FVRCON	197h	VREGCON	217h	—	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCPR2L	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	—	219h	—	299h	CCPR2H	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	—	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	—	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	—	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	APFCON	19Dh	—	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	—	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	—	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	Unimplemented Read as ‘0’	220h 26Fh	Unimplemented Read as ‘0’	2A0h 2EFh	Unimplemented Read as ‘0’	320h 36Fh	Unimplemented Read as ‘0’	3A0h 3EFh	Unimplemented Read as ‘0’
070h 07Fh	Common RAM	0F0h 0FFh	Common RAM (Accesses 70h – 7Fh)	170h 17Fh	Common RAM (Accesses 70h – 7Fh)	1F0h 1FFh	Common RAM (Accesses 70h – 7Fh)	270h 27Fh	Common RAM (Accesses 70h – 7Fh)	2F0h 2FFh	Common RAM (Accesses 70h – 7Fh)	370h 37Fh	Common RAM (Accesses 70h – 7Fh)	3F0h 3FFh	Common RAM (Accesses 70h – 7Fh)

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

TABLE 3-4: PIC12(L)F1612/16(L)F1613 MEMORY MAP, BANK 8-23

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h 40Bh	Core Registers (Table 3-1)	480h 48Bh	Core Registers (Table 3-1)	500h 50Bh	Core Registers (Table 3-1)	580h 58Bh	Core Registers (Table 3-1)	600h 60Bh	Core Registers (Table 3-1)	680h 68Bh	Core Registers (Table 3-1)	700h 70Bh	Core Registers (Table 3-1)	780h 78Bh	Core Registers (Table 3-1)
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	WDTCON0	791h	CRCDATL
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	WDTCON1	792h	CRCDATH
413h	TMR4	493h	—	513h	—	593h	—	613h	—	693h	CWG1AS0	713h	WDTPSL	793h	CRCACCL
414h	PR4	494h	—	514h	—	594h	—	614h	—	694h	CWG1AS1	714h	WDTPSH	794h	CRCACCH
415h	T4CON	495h	—	515h	—	595h	—	615h	—	695h	CWG1OCON0	715h	WDTTMR	795h	CRCSHIFTL
416h	T4HLT	496h	—	516h	—	596h	—	616h	—	696h	CWG1CON0	716h	—	796h	CRCSHIFTH
417h	T4CLKCON	497h	—	517h	—	597h	—	617h	—	697h	CWG1CON1	717h	—	797h	CRCXORL
418h	T4RST	498h	—	518h	—	598h	—	618h	—	698h	CWG1OCON1	718h	SCANLADRL	798h	CRCXORH
419h	—	499h	—	519h	—	599h	—	619h	—	699h	CWG1CLKCON	719h	SCANLADRH	799h	CRCCON0
41Ah	TMR6	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	CWG1ISM	71Ah	SCANHADRL	79Ah	CRCCON1
41Bh	PR6	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	SCANHADRH	79Bh	—
41Ch	T6CON	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	SCANCON0	79Ch	—
41Dh	T6HLT	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	SCANTRIG	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	T6RST	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

BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h 80Bh	Core Registers (Table 3-1)	880h 88Bh	Core Registers (Table 3-1)	900h 90Bh	Core Registers (Table 3-1)	980h 98Bh	Core Registers (Table 3-1)	A00h A0Bh	Core Registers (Table 3-1)	A80h A8Bh	Core Registers (Table 3-1)	B00h B0Bh	Core Registers (Table 3-1)	B80h B8Bh	Core Registers (Table 3-1)
80Ch 86Fh	Unimplemented Read as ‘0’	88Ch 8EFh	Unimplemented Read as ‘0’	90Ch 96Fh	Unimplemented Read as ‘0’	98Ch 9EFh	Unimplemented Read as ‘0’	A0Ch A6Fh	Unimplemented Read as ‘0’	A8Ch AEFh	Unimplemented Read as ‘0’	B0Ch B6Fh	Unimplemented Read as ‘0’	B8Ch BEFh	Unimplemented Read as ‘0’
870h 87Fh	Accesses 70h – 7Fh	8F0h 8FFh	Accesses 70h – 7Fh	970h 97Fh	Accesses 70h – 7Fh	9F0h 9FFh	Accesses 70h – 7Fh	A70h A7Fh	Accesses 70h – 7Fh	AF0h AFFh	Accesses 70h – 7Fh	B70h B7Fh	Accesses 70h – 7Fh	BF0h BFFh	Accesses 70h – 7Fh

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

TABLE 3-5: PIC12(L)F1612/16(L)F1613 MEMORY MAP, BANK 24-31

	BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31
C00h C0Bh	Core Registers (Table 3-1)	C80h C8Bh	Core Registers (Table 3-1)	D00h D0Bh	Core Registers (Table 3-1)	D80h D8Bh	Core Registers (Table 3-1)	E00h E0Bh	Core Registers (Table 3-1)	E80h E8Bh	Core Registers (Table 3-1)	F00h F0Bh	Core Registers (Table 3-1)	F80h F8Bh	Core Registers (Table 3-1)
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	See Table 3-6 for register mapping details	E0Ch	—	E8Ch	—	F0Ch	—	F8Ch	See Table 3-7 for register mapping details
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh		E0Dh	—	E8Dh	—	F0Dh	—	F8Dh	
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh		E0Eh	—	E8Eh	—	F0Eh	—	F8Eh	
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh		E0Fh	—	E8Fh	—	F0Fh	—	F8Fh	
C10h	—	C90h	—	D10h	—	D90h		E10h	—	E90h	—	F10h	—	F90h	
C11h	—	C91h	—	D11h	—	D91h		E11h	—	E91h	—	F11h	—	F91h	
C12h	—	C92h	—	D12h	—	D92h		E12h	—	E92h	—	F12h	—	F92h	
C13h	—	C93h	—	D13h	—	D93h		E13h	—	E93h	—	F13h	—	F93h	
C14h	—	C94h	—	D14h	—	D94h		E14h	—	E94h	—	F14h	—	F94h	
C15h	—	C95h	—	D15h	—	D95h		E15h	—	E95h	—	F15h	—	F95h	
C16h	—	C96h	—	D16h	—	D96h		E16h	—	E96h	—	F16h	—	F96h	
C17h	—	C97h	—	D17h	—	D97h		E17h	—	E97h	—	F17h	—	F97h	
C18h	—	C98h	—	D18h	—	D98h		E18h	—	E98h	—	F18h	—	F98h	
C19h	—	C99h	—	D19h	—	D99h		E19h	—	E99h	—	F19h	—	F99h	
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah		E1Ah	—	E9Ah	—	F1Ah	—	F9Ah	
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh		E1Bh	—	E9Bh	—	F1Bh	—	F9Bh	
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch		E1Ch	—	E9Ch	—	F1Ch	—	F9Ch	
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh		E1Dh	—	E9Dh	—	F1Dh	—	F9Dh	
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh		E1Eh	—	E9Eh	—	F1Eh	—	F9Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh		E1Fh	—	E9Fh	—	F1Fh	—	F9Fh	
C20h C6Fh	Unimplemented Read as ‘0’	CA0h CEFh	Unimplemented Read as ‘0’	D20h D6Fh	Unimplemented Read as ‘0’	DA0h DEFh		E20h E6Fh	Unimplemented Read as ‘0’	EA0h EEFh	Unimplemented Read as ‘0’	F20h F6Fh	Unimplemented Read as ‘0’	FA0h FEFh	
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: ▨ = Unimplemented data memory locations, read as ‘0’.

TABLE 3-6: PIC12(L)F1612/16(L)F1613 MEMORY MAP, BANK 27

	Bank 27
D8Ch	SMT1TMRL
D8Dh	SMT1TMRH
D8Eh	SMT1TMRU
D8Fh	SMT1CPRL
D90h	SMT1CPRH
D91h	SMT1CPRU
D92h	SMT1CPWL
D93h	SMT1CPWH
D94h	SMT1CPWU
D95h	SMT1PRL
D96h	SMT1PRH
D97h	SMT1PRU
D98h	SMT1CON0
D99h	SMT1CON1
D9Ah	SMT1STAT
D9Bh	SMT1CLK
D9Ch	SMT1SIG
D9Dh	SMT1WIN
D9Eh	SMT2TMRL
D9Fh	SMT2TMRH
DA0h	SMT2TMRU
DA1h	SMT2CPRL
DA2h	SMT2CPRH
DA3h	SMT2CPRU
DA4h	SMT2CPWL
DA5h	SMT2CPWH
DA6h	SMT2CPWU
DA7h	SMT2PRL
DA8h	SMT2PRH
DA9h	SMT2PRU
DAAh	SMT2CON0
DABh	SMT2CON1
DACh	SMT2STAT
DADh	SMT2CLK
DAEh	SMT2SIG
DAFh	SMT2WIN
DB0h DEFh	—

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

TABLE 3-7: PIC12(L)F1612/16(L)F1613 MEMORY MAP, BANK 31

	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 3-8 can be addressed from any Bank.

TABLE 3-8: 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	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	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 3-9: SPECIAL FUNCTION REGISTER 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											
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	`--xx xxxx`	`--xx xxxx`
00Dh	—	Unimplemented								—	—
00Eh	PORTC(4)	—	—	RC5	RC4	RC3	RC2	RC1	RC0	`--xx xxxx`	`--xx xxxx`
00Fh	—	Unimplemented								—	—
010h	—	Unimplemented								—	—
011h	PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	`00-- -000`	`00-- -000`
012h	PIR2	—	C2IF(4)	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	`-00- -000`	`-00- -000`
013h	PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	`--00 ----`	`--00 ----`
014h	PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	`0000 0000`	`0000 0000`
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	ON	CKPS<2:0>			OUTPS<3:0>				`0000 0000`	`0000 0000`
01Dh	T2HLT	PSYNC	CKPOL	CKSYNC	—	MODE<3:0>				`000- 0000`	`000- 0000`
01Eh	T2CLKCON	—	—	—	—	—	T2CS<2:0>			`---- -000`	`---- -000`
01Fh	T2RST	—	—	—	—	RSEL<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: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 1											
08Ch	TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	`--11 1111`	`--11 1111`
08Dh	—	Unimplemented								—	—
08Eh	TRISC(4)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`--11 1111`	`--11 1111`
08Fh	—	Unimplemented								—	—
090h	—	Unimplemented								—	—
091h	PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	`00-- -000`	`00-- -000`
092h	PIE2	—	C2IE(4)	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	`-00- -000`	`-00- -000`
093h	PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	`--00 ----`	`--00 ----`
094h	PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	`0000 0000`	`0000 0000`
095h	OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			`1111 1111`	`1111 1111`
096h	PCON	STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	`00-1 11qq`	`qq-q qquu`
097h	—	Unimplemented								—	—
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	—	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	`-0-0 0000`	`-q-q 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: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 2											
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu
10Dh	—	Unimplemented								—	—
10Eh	LATC(4)	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	--xx xxxx	--uu uuuu
10Fh	—	Unimplemented								—	—
110h	—	Unimplemented								—	—
111h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	0000 -100	0000 -100
112h	CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			0000 -000	0000 -000
113h	CM2CON0(4)	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	0000 -100	0000 -100
114h	CM2CON1(4)	C2INTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			0000 -000	0000 -000
115h	CMOUT	—	—	—	—	—	—	MC2OUT(4)	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	DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	0-0- 00--	0-0- 00--
119h	DAC1CON1	DAC1R<7:0>								0000 0000	0000 0000
11Ah	—	Unimplemented								—	—
11Bh	—	Unimplemented								—	—
11Ch	ZCD1CON	ZCD1EN	ZCD1OE	ZCD1OUT	ZCD1POL	—	—	ZCD1INTP	ZCD1INTN	0000 --00	0000 --00
11Dh	APFCON	—	CWGASEL(3)	CWGBSEL(3)	—	T1GSEL	—	CCP2SEL(4)	CCP1SEL(3)	-00- 0-00	-00- 0-00
11Eh	—	Unimplemented								—	—
11Fh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 3											
18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	`---1 -111`	`---1 -111`
18Dh	—	Unimplemented								—	—
18Eh	ANSELC[4]	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	`---- 1111`	`---- 1111`
18Fh	—	Unimplemented								—	—
190h	—	Unimplemented								—	—
191h	PMADRL	Flash Program Memory Address Register Low Byte								`0000 0000`	`0000 0000`
192h	PMADRH	—[2]	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	—[2]	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[1]	—	—	—	—	—	—	VREGPM	Reserved	`---- --01`	`---- --01`
198h to 19Fh	—	Unimplemented								—	—
Bank 4											
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	`--11 1111`	`--11 1111`
20Dh	—	Unimplemented								—	—
20Eh	WPUC[4]	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	`--11 1111`	`--11 1111`
20Fh to 21Fh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 5											
28Ch	ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	`--00 -000`	`--00 -000`
28Dh	—	Unimplemented								—	—
28Eh	ODCONC[(4)]	—	—	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	`--00 0000`	`--00 0000`
28Fh	—	Unimplemented								—	—
290h	—	Unimplemented								—	—
291h	CCP1RL	Capture/Compare/PWM 1 Register (LSB)								`xxxx xxxx`	`uuuu uuuu`
292h	CCP1RH	Capture/Compare/PWM 1 Register (MSB)								`xxxx xxxx`	`uuuu uuuu`
293h	CCP1CON	EN	OE	OUT	FMT	MODE<3:0>				`0000 0000`	`0000 0000`
294h	CCP1CAP	—	—	—	—	—	—	CTS<1:0>		`---- --00`	`---- --00`
295h — 297h	—	Unimplemented								—	—
298h	CCP2RL	Capture/Compare/PWM 2 Register (LSB)								`xxxx xxxx`	`uuuu uuuu`
299h	CCP2RH	Capture/Compare/PWM 2 Register (MSB)								`xxxx xxxx`	`uuuu uuuu`
29Ah	CCP2CON	EN	OE	OUT	FMT	MODE<3:0>				`0000 0000`	`0000 0000`
29Bh	CCP2CAP	—	—	—	—	—	—	CTS<1:0>		`---- --00`	`---- --00`
29Ch	—	Unimplemented								—	—
29Dh	—	Unimplemented								—	—
29Eh	CCPTMRS	—	—	—	—	C2TSEL<1:0>		C1TSEL<1:0>		`---- 0000`	`---- 0000`
29Fh	—	Unimplemented								—	—
Bank 6											
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	`--00 -000`	`--00 -000`
30Dh	—	Unimplemented								—	—
30Eh	SLRCONC[(4)]	—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	`--00 0000`	`--00 0000`
30Fh — 31Fh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 7											
38Ch	INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	`--11 1111`	`--11 1111`
38Dh	—	Unimplemented								—	—
38Eh	INLVLC(4)	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	`--11 1111`	`--11 1111`
30Fh	—	Unimplemented								—	—
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	—	Unimplemented								—	—
395h	—	Unimplemented								—	—
396h	—	Unimplemented								—	—
397h	IOCCP(4)	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	`--00 0000`	`--00 0000`
398h	IOCCN(4)	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	`--00 0000`	`--00 0000`
399h	IOCCF(4)	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	`--00 0000`	`--00 0000`
39Ah to 39Fh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 8											
40Ch to 412h	—	Unimplemented								—	—
413h	TMR4	Timer4 Module Register								`0000 0000`	`0000 0000`
414h	PR4	Timer4 Period Register								`1111 1111`	`1111 1111`
415h	T4CON	ON	CKPS<2:0>			OUTPS<3:0>				`0000 0000`	`0000 0000`
416h	T4HLT	PSYNC	CKPOL	CKSYNC	—	MODE<3:0>				`000- 0000`	`000- 0000`
417h	T4CLKCON	—	—	—	—	—	T4CS<2:0>			`---- -000`	`---- -000`
418h	T4RST	—	—	—	—	RSEL<3:0>				`---- 0000`	`---- 0000`
419h	—	Unimplemented								—	—
41Ah	TMR6	Timer6 Module Register								`0000 0000`	`0000 0000`
41Bh	PR6	Timer6 Period Register								`1111 1111`	`1111 1111`
41Ch	T6CON	ON	CKPS<2:0>			OUTPS<3:0>				`0000 0000`	`0000 0000`
41Dh	T6HLT	PSYNC	CKPOL	CKSYNC	—	MODE<3:0>				`000- 0000`	`000- 0000`
41Eh	T6CLKCON	—	—	—	—	—	T6CS<2:0>			`---- -000`	`---- -000`
41Fh	T6RST	—	—	—	—	RSEL<3:0>				`---- 0000`	`---- 0000`
Bank 9											
48Ch to 49Fh	—	Unimplemented								—	—
Bank 10											
50Ch to 51Fh	—	Unimplemented								—	—
Bank 11											
58Ch to 59Fh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.
2: Unimplemented, read as ‘`1`’.
3: PIC12(L)F1612 only.
4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 12											
60Ch to 61Fh	—	Unimplemented								—	—
Bank 13											
68Ch to 690h	—	Unimplemented								—	—
691h	CWG1DBR	—	—	DBR<5:0>						`--00 0000`	`--00 0000`
692h	CWG1DBF	—	—	DBF<5:0>						`--xx xxxx`	`--xx xxxx`
693h	CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	`0000 00--`	`0000 00--`
694h	CWG1AS1	—	TMR6AS	TMR4AS	TMR2AS	—	C2AS(4)	C1AS	INAS	`-000 -000`	`-000 -000`
695h	CWG1OCON0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	`0000 0000`	`0000 0000`
696h	CWG1CON0	EN	LD	—	—	—	MODE<2:0>			`00-- -000`	`00-- -000`
697h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	`--x- 0000`	`--x- 0000`
698h	CWG1OCON1	—	—	—	—	OED	OEC	OEB	OEA	`---- 0000`	`---- 0000`
699h	CWG1CLKCON	—	—	—	—	—	—	—	CS	`---- ---0`	`---- ---0`
69Ah	CWG1ISM	—	—	—	—	—	IS<2:0>			`---- -000`	`---- -000`
69Bh to 6EFh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 14											
70Ch to 710h	—	Unimplemented								—	—
711h	WDTCON0	—	—	WDTPS<4:0>					SEN	--qq qqqq	--qq qqqq
712h	WDTCON1	—	WDTCS<2:0>			—	WINDOW<2:0>			-qqq -qqq	-qqq -qqq
713h	WDTPSL	PSCNT<7:0>								0000 0000	0000 0000
714h	WDTPSH	PSCNT<15:8>								0000 0000	0000 0000
715h	WDTTMR	WDTTMR<4:0>					STATE	PSCNT<17:16>		0000 0000	0000 0000
716h	—	Unimplemented								—	—
717h	—	Unimplemented								—	—
718h	SCANLADRL	LADR<7:0>								0000 0000	0000 0000
719h	SCANLADRH	LADR<15:8>								0000 0000	0000 0000
71Ah	SCANHADRL	HADR<7:0>								1111 1111	1111 1111
71Bh	SCANHADRH	HADR<15:8>								1111 1111	1111 1111
71Ch	SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		0000 0-00	0000 0-00
71Dh	SCANTRIG					—	—	TSEL<1:0>		---- --00	---- --00
71Eh	—	Unimplemented								—	—
71Fh	—	Unimplemented								—	—

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 15											
78Ch to 790h	—	Unimplemented								—	—
791h	CRCDATL	DATA<7:0>								`xxxx xxxx`	`xxxx xxxx`
792h	CRCDATH	DATA<15:8>								`xxxx xxxx`	`xxxx xxxx`
793h	CRCACCL	ACC<7:0>								`0000 0000`	`0000 0000`
794h	CRCACCH	ACC<15:8>								`0000 0000`	`0000 0000`
795h	CRCSHIFTL	SHIFT<7:0>								`0000 0000`	`0000 0000`
796h	CRCSHIFTH	SHIFT<15:8>								`0000 0000`	`0000 0000`
797h	CRCXORL	XOR<7:1>							—	`xxxx xxx-`	`xxxx xxx-`
798h	CRCXORH	XOR<15:8>								`xxxx xxxx`	`xxxx xxxx`
799h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	`0000 --00`	`0000 -00`
79Ah	CRCCON1	DLEN<3:0>				PLEN<3:0>				`0000 0000`	`0000 0000`
79Bh to 79Fh	—	Unimplemented								—	—
Bank 16-26											
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: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 27											
D80h to D8Bh	—	Unimplemented								—	—
D8Ch	SMT1TMRL	SMT1TMR<7:0>								`0000 0000`	`0000 0000`
D8Dh	SMT1TMRH	SMT1TMR<15:8>								`0000 0000`	`0000 0000`
D8Eh	SMT1TMRU	SMT1TMR<23:16>								`0000 0000`	`0000 0000`
D8Fh	SMT1CPRL	SMT1CPR<7:0>								`xxxx xxxx`	`xxxx xxxx`
D90h	SMT1CPRH	SMT1CPR<15:8>								`xxxx xxxx`	`xxxx xxxx`
D91h	SMT1CPRU	SMT1CPR<23:16>								`xxxx xxxx`	`xxxx xxxx`
D92h	SMT1CPWL	SMT1CPW<7:0>								`xxxx xxxx`	`xxxx xxxx`
D93h	SMT1CPWH	SMT1CPW<15:8>								`xxxx xxxx`	`xxxx xxxx`
D94h	SMT1CPWU	SMT1CPW<23:16>								`xxxx xxxx`	`xxxx xxxx`
D95h	SMT1PRL	SMT1PR<7:0>								`xxxx xxxx`	`xxxx xxxx`
D96h	SMT1PRH	SMT1PR<15:8>								`xxxx xxxx`	`xxxx xxxx`
D97h	SMT1PRU	SMT1PR<23:16>								`xxxx xxxx`	`xxxx xxxx`
D98h	SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMTxPS<1:0>		`0-00 0000`	`0-00 0000`
D99h	SMT1CON1	SMTxGO	REPEAT	—	—	MODE<3:0>				`00-- 0000`	`00-- 0000`
D9Ah	SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	`000- -000`	`000- -000`
D9Bh	SMT1CLK	—	—	—	—	—	CSEL<2:0>			`---- -000`	`---- -000`
D9Ch	SMT1SIG	—	—	—	—	SSEL<3:0>				`---- 0000`	`---- 0000`
D9Dh	SMT1WIN	—	—	—	—	—	WSEL<2:0>			`---- -000`	`---- -000`
D9Eh	SMT2TMRL	SMT2TMR<7:0>								`0000 0000`	`0000 0000`
D9Fh	SMT2TMRH	SMT2TMR<15:8>								`0000 0000`	`0000 0000`
DA0h	SMT2TMRU	SMT2TMR<23:16>								`0000 0000`	`0000 0000`
DA1h	SMT2CPRL	SMT2CPR<7:0>								`xxxx xxxx`	`xxxx xxxx`
DA2h	SMT2CPRH	SMT2CPR<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA3h	SMT2CPRU	SMT2CPR<23:16>								`xxxx xxxx`	`xxxx xxxx`
DA4h	SMT2CPWL	SMT2CPW<7:0>								`xxxx xxxx`	`xxxx xxxx`

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 27 (Continued)											
DA5h	SMT2CPWH	SMTxCPW<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA6h	SMT2CPWU	SMTxCPW<23:16>								`xxxx xxxx`	`xxxx xxxx`
DA7h	SMT2PRL	SMTxPR<7:0>								`xxxx xxxx`	`xxxx xxxx`
DA8h	SMT2PRH	SMTxPR<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA9h	SMT2PRU	SMTxPR<23:16>								`xxxx xxxx`	`xxxx xxxx`
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMTxPS<1:0>		`0-00 0000`	`0-00 0000`
DABh	SMT2CON1	SMTxGO	REPEAT	—	—	MODE<3:0>				`00-- 0000`	`00-- 0000`
DACh	SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	`000- -000`	`000- -000`
DADh	SMT2CLK	—	—	—	—	—	CSEL<2:0>			`---- -000`	`---- -000`
DAEh	SMT2SIG	—	—	—	—	SSEL<3:0>				`---- 0000`	`---- 0000`
DAFh	SMT2WIN	—	—	—	—	—	WSEL<2:0>			`---- -000`	`---- -000`

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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
Banks 28											
DA5h	SMT2CPWH	SMTxCPW<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA6h	SMT2CPWU	SMTxCPW<23:16>								`xxxx xxxx`	`xxxx xxxx`
DA7h	SMT2PRL	SMTxPR<7:0>								`xxxx xxxx`	`xxxx xxxx`
DA8h	SMT2PRH	SMTxPR<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA9h	SMT2PRU	SMTxPR<23:16>								`xxxx xxxx`	`xxxx xxxx`
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMTxPS<1:0>		`0-00 0000`	`0-00 0000`
DABh	SMT2CON1	SMTxGO	REPEAT	—	—	MODE<3:0>				`00-- 0000`	`00-- 0000`
DACh	SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	`000- -000`	`000- -000`
DADh	SMT2CLK	—	—	—	—	—	CSEL<2:0>			`---- -000`	`---- -000`
DAEh	SMT2SIG	—	—	—	—	SSEL<3:0>				`---- 0000`	`---- 0000`
DAFh	SMT2WIN	—	—	—	—	—	WSEL<2:0>			`---- -000`	`---- -000`
DA5h	SMT2CPWH	SMTxCPW<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA6h	SMT2CPWU	SMTxCPW<23:16>								`xxxx xxxx`	`xxxx xxxx`
DA7h	SMT2PRL	SMTxPR<7:0>								`xxxx xxxx`	`xxxx xxxx`
DA8h	SMT2PRH	SMTxPR<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA9h	SMT2PRU	SMTxPR<23:16>								`xxxx xxxx`	`xxxx xxxx`
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMTxPS<1:0>		`0-00 0000`	`0-00 0000`
DABh	SMT2CON1	SMTxGO	REPEAT	—	—	MODE<3:0>				`00-- 0000`	`00-- 0000`
DACh	SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	`000- -000`	`000- -000`
DADh	SMT2CLK	—	—	—	—	—	CSEL<2:0>			`---- -000`	`---- -000`
DAEh	SMT2SIG	—	—	—	—	SSEL<3:0>				`---- 0000`	`---- 0000`
DAFh	SMT2WIN	—	—	—	—	—	WSEL<2:0>			`---- -000`	`---- -000`
DA5h	SMT2CPWH	SMTxCPW<15:8>								`xxxx xxxx`	`xxxx xxxx`
DA6h	SMT2CPWU	SMTxCPW<23:16>								`xxxx xxxx`	`xxxx xxxx`
DA7h	SMT2PRL	SMTxPR<7:0>								`xxxx xxxx`	`xxxx xxxx`

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

Note 1: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘`1`’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 29-30											
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: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 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: PIC12F1612/16F1613 only.

2: Unimplemented, read as ‘1’.

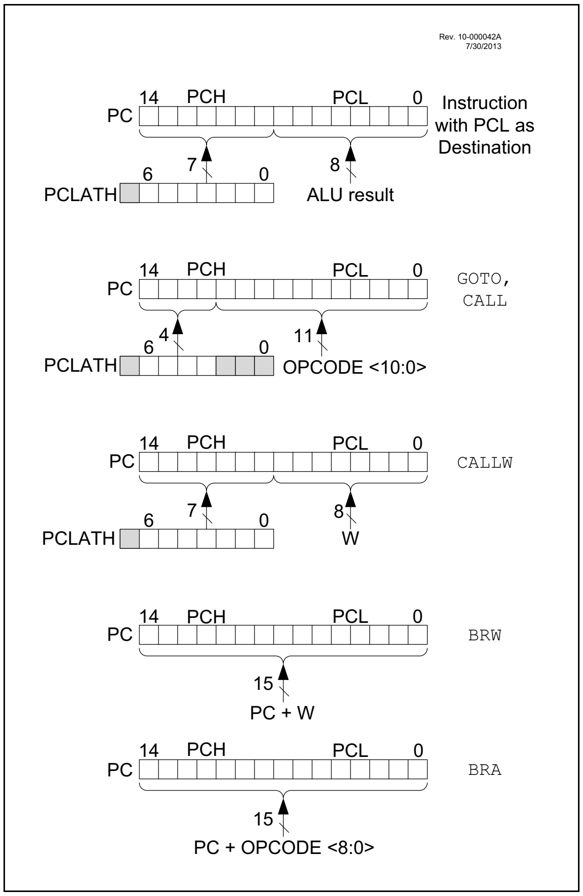
3: PIC12(L)F1612 only.

4: PIC16(L)F1613 only.

3.4 PCL and PCLATH

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

FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS



3.4.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the program counter will change to the values contained in the PCLATH register 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 3-4 through 3-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when `CALL` or `CALLW` instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

Note 1: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `CALLW`, `RETURN`, `RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

3.5.1 ACCESSING THE STACK

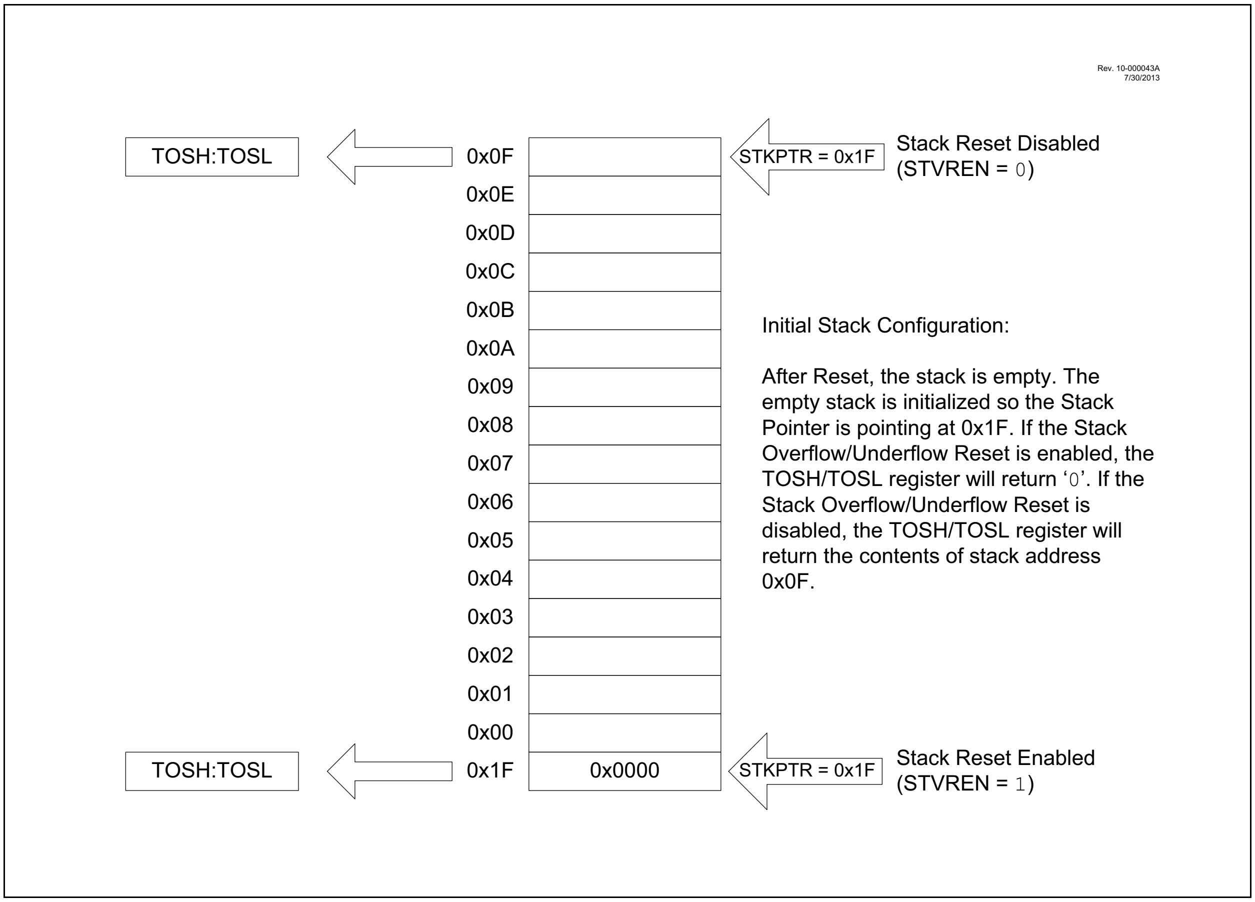
The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust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

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

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

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

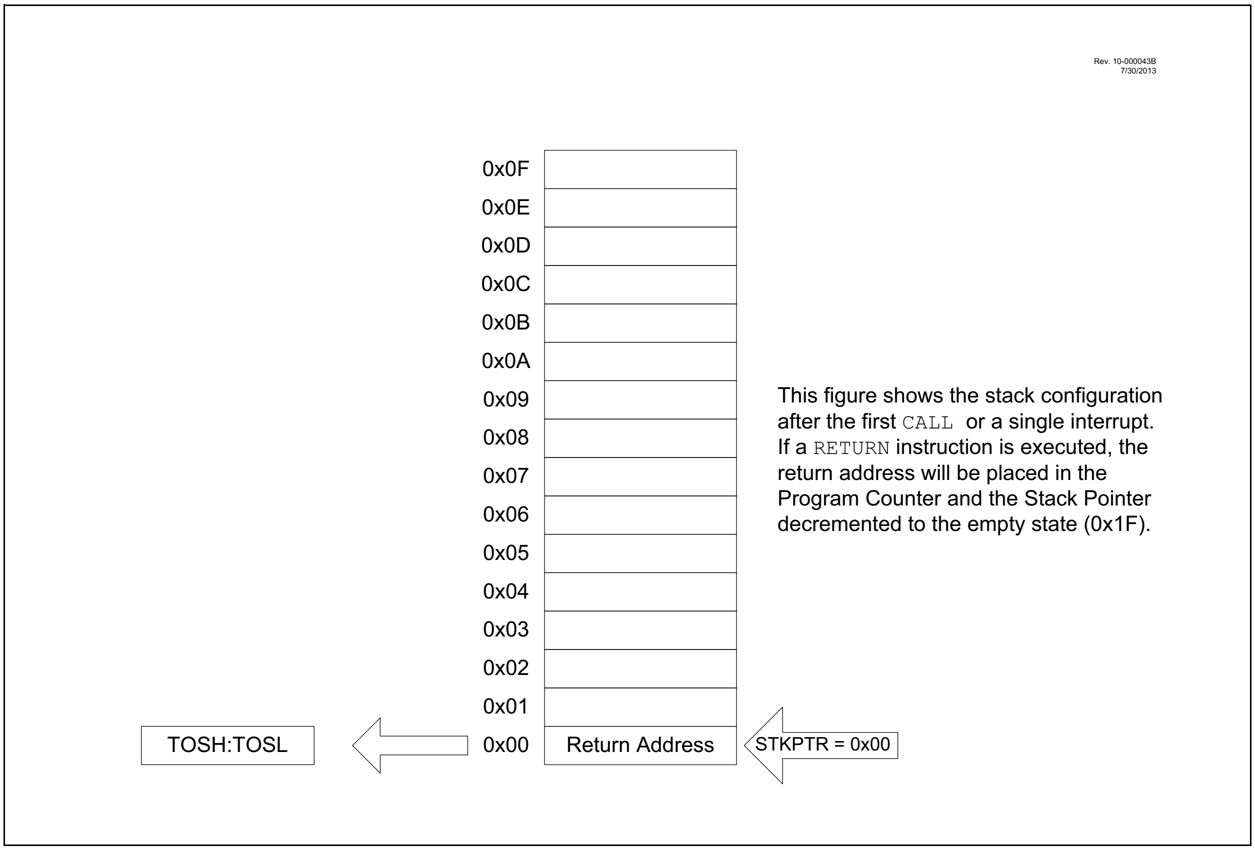
FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1



Initial Stack Configuration:

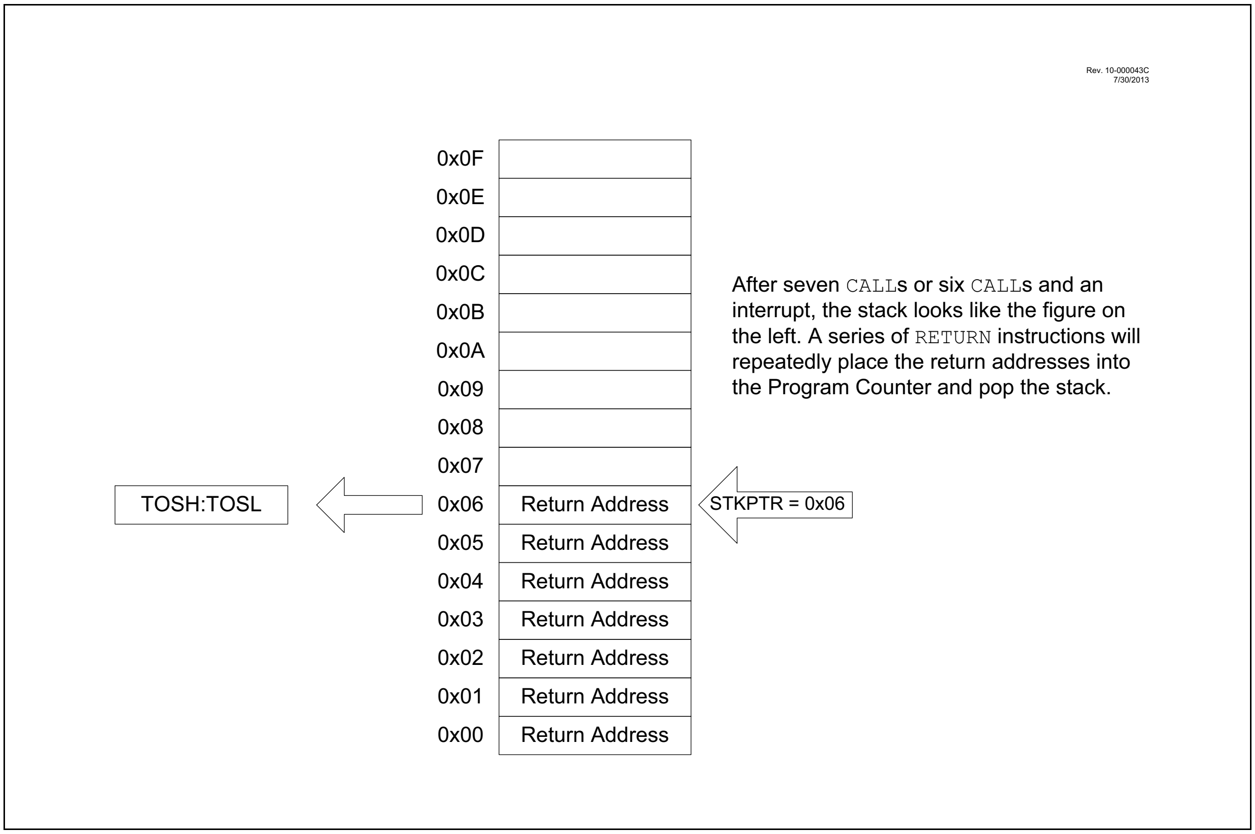
After Reset, the stack is empty. The empty stack is initialized so the Stack Pointer is pointing at 0x1F. If the Stack Overflow/Underflow Reset is enabled, the TOSH/TOSL register will return '`0`'. If the Stack Overflow/Underflow Reset is disabled, the TOSH/TOSL register will return the contents of stack address 0x0F.

FIGURE 3-5: ACCESSING THE STACK EXAMPLE 2



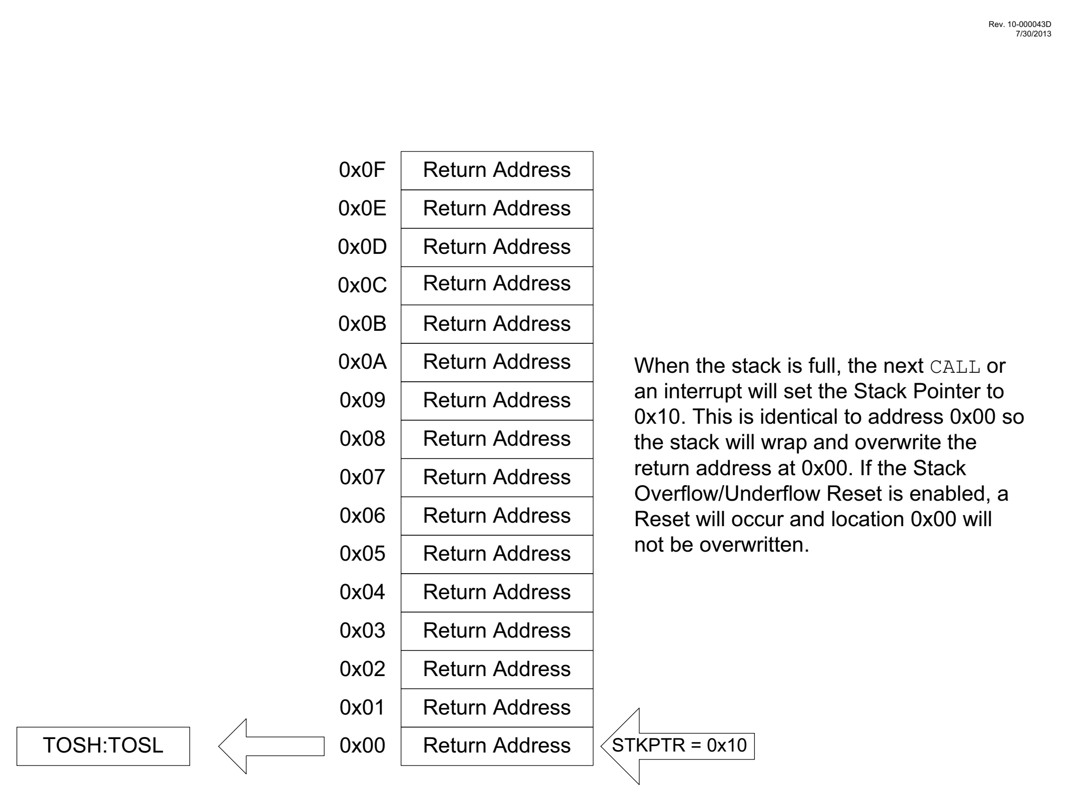
This figure shows the stack configuration after the first `CALL` or a single interrupt. If a `RETURN` instruction is executed, the return address will be placed in the Program Counter and the Stack Pointer decremented to the empty state (0x1F).

FIGURE 3-6: ACCESSING THE STACK EXAMPLE 3



After seven `CALL`s or six `CALL`s and an interrupt, the stack looks like the figure on the left. A series of `RETURN` instructions will repeatedly place the return addresses into the Program Counter and pop the stack.

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



When the stack is full, the next `CALL` or an interrupt will set the Stack Pointer to 0x10. This is identical to address 0x00 so the stack will wrap and overwrite the return address at 0x00. If the Stack Overflow/Underflow Reset is enabled, a Reset will occur and location 0x00 will not be overwritten.

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.

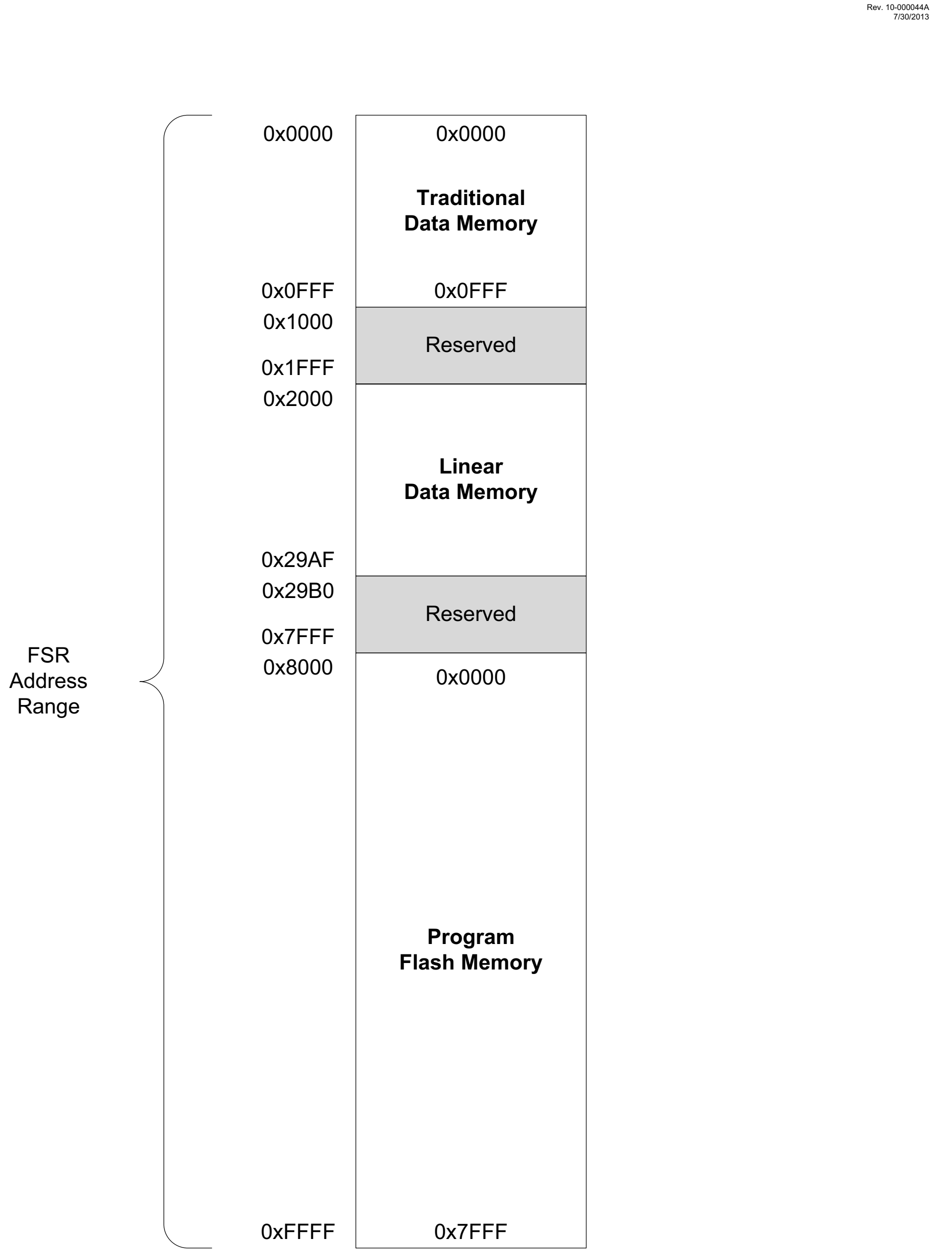
3.6 Indirect Addressing

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 3-8: INDIRECT ADDRESSING

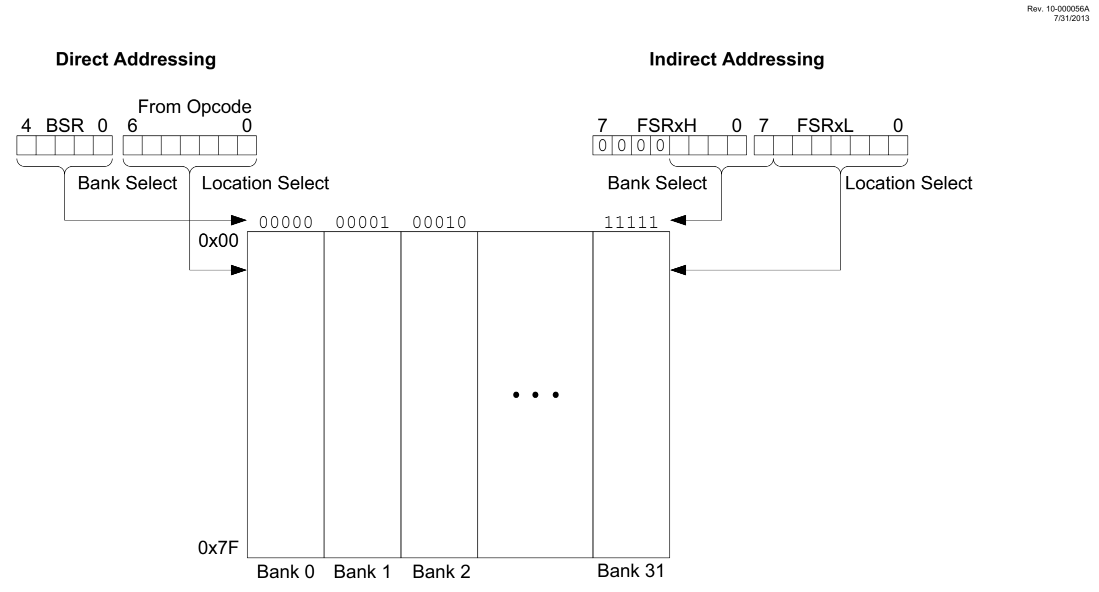


Note: Not all memory regions are completely implemented. Consult device memory tables for memory limits.

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 3-9: TRADITIONAL DATA MEMORY MAP



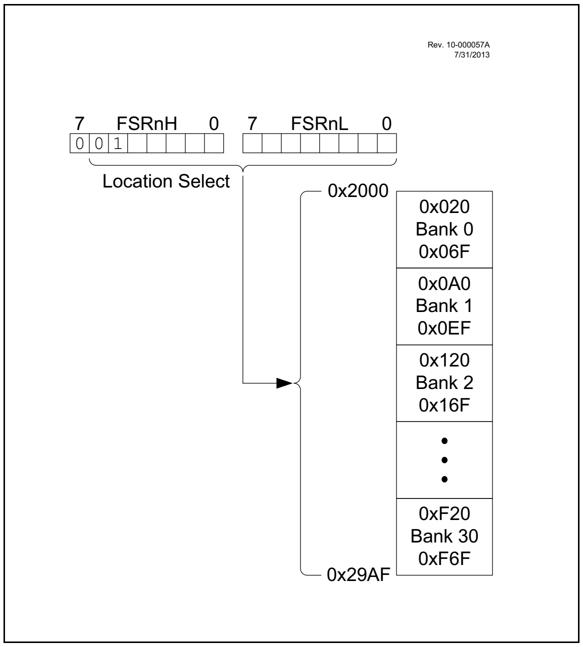
3.6.2 LINEAR DATA MEMORY

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

On the PIC12(L)F1612/16(L)F1613, linear data memory is implemented from FSR address 0x2000 to 0x20EF. Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

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

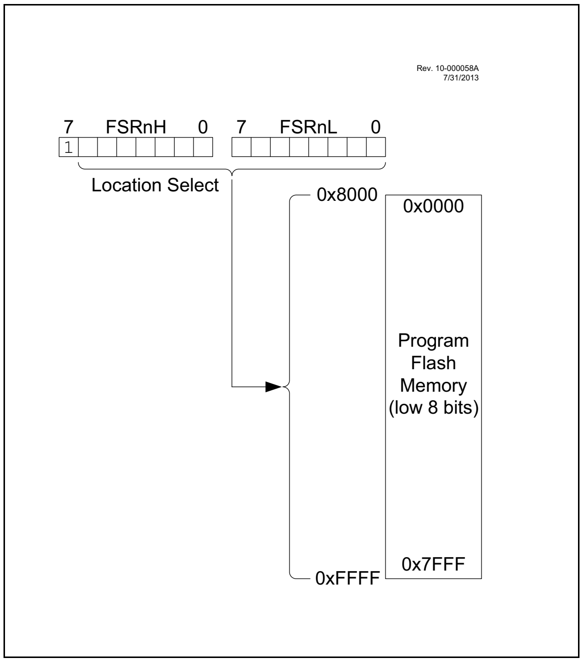
FIGURE 3-10: LINEAR DATA MEMORY MAP



3.6.3 PROGRAM FLASH MEMORY

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

FIGURE 3-11: PROGRAM FLASH MEMORY MAP



4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h, Configuration Word 2 at 8008h, and Configuration 3 at 8009h.

Note:	The $\overline{\text{DEBUG}}$ 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: CONFIG1: CONFIGURATION WORD 1

U-1	U-1	R/P-1	R/P-1	R/P-1	U-1
—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>(1)		—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	U-1	U-1	U-1	R/P-1	R/P-1
$\overline{\text{CP}}$(2)	MCLRE	$\overline{\text{PWRTE}}$	—	—	—	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 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
`1` = CLKOUT function is disabled. I/O function on the CLKOUT pin
`0` = CLKOUT function is enabled on the CLKOUT pin

bit 10-9 **BOREN<1:0>:** Brown-Out Reset Enable bits(1)
`11` = BOR enabled
`10` = BOR enabled during operation and disabled in Sleep
`01` = BOR controlled by SBOREN bit of the BORCON register
`00` = BOR disabled

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

bit 7 **$\overline{\text{CP}}$:** Code Protection bit(2)
`1` = Program memory code protection is disabled
`0` = Program memory code protection is enabled

bit 6 **MCLRE:** $\overline{\text{MCLR}}$/VPP Pin Function Select bit
If LVP bit = `1`:
This bit is ignored.
If LVP bit = `0`:
`1` = $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$; Weak pull-up enabled.
`0` = $\overline{\text{MCLR}}$/VPP pin function is digital input; $\overline{\text{MCLR}}$ internally disabled; Weak pull-up under control of WPUA3 bit.

bit 5 **$\overline{\text{PWRTE}}$:** Power-Up Timer Enable bit
`1` = PWRT disabled
`0` = PWRT enabled

bit 4-2 **Unimplemented:** Read as '`1`'

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

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: CONFIG2: CONFIGURATION WORD 2

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
LVP[1]	$\overline{\text{DEBUG}}$[3]	$\overline{\text{LPBOR}}$	BORV[2]	STVREN	PLLEN
bit 13					bit 8

R/P-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
$\overline{\text{ZCD}}$	—	—	—	—	—	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 = Low-voltage programming enabled 0 = High-voltage on $\overline{\text{MCLR}}$ must be used for programming
bit 12	**$\overline{\text{DEBUG}}$:** In-Circuit Debugger Mode bit[3] 1 = In-Circuit Debugger disabled, ICSPCLK and ICSPDAT are general purpose I/O pins 0 = In-Circuit Debugger enabled, ICSPCLK and ICSPDAT are dedicated to the debugger
bit 11	**$\overline{\text{LPBOR}}$:** Low-Power BOR Enable bit 1 = Low-Power Brown-out Reset is disabled 0 = Low-Power Brown-out Reset is enabled
bit 10	**BORV:** Brown-Out Reset Voltage Selection bit[2] 1 = Brown-out Reset voltage (V_{BOR}), low trip point selected 0 = Brown-out Reset voltage (V_{BOR}), high trip point selected
bit 9	**STVREN:** Stack Overflow/Underflow Reset Enable bit 1 = Stack Overflow or Underflow will cause a Reset 0 = Stack Overflow or Underflow will not cause a Reset
bit 8	**PLLEN:** PLL Enable bit 1 = 4xPLL enabled 0 = 4xPLL disabled
bit 7	**$\overline{\text{ZCD}}$:** ZCD Disable bit 1 = ZCD disabled. ZCD can be enabled by setting the ZCD1EN bit of ZCD1CON 0 = ZCD always enabled
bit 6-2	**Unimplemented:** Read as ‘1’
bit 1-0	**WRT<1:0>:** Flash Memory Self-Write Protection bits 2 kW Flash memory (PIC12(L)F1612/16(L)F1613): 11 = OFF - Write protection off 10 = BOOT - 000h to 1FFh write-protected, 200h to 7FFh may be modified by PMCON control 01 = HALF - 000h to 3FFh write-protected, 400h to 7FFh may be modified by PMCON control 00 = ALL - 000h to 7FFh write-protected, no addresses may be modified by PMCON control

Note 1: The LVP bit cannot be programmed to ‘0’ when Programming mode is entered via LVP.

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

3: 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’.

REGISTER 4-3: CONFIG3: CONFIGURATION WORD 3

R/P-0	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1
WDTCCS<2:0>			WDTCWS<2:0>		

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

bit 13	bit 8

U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	WDTE<1:0>		WDTCPS<4:0>				
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 13-11 **WDTCCS<2:0>:** WDT Configuration Clock Select bits

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

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

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

bit 7 **Unimplemented:** Read as ‘1’

bit 6-5 **WDTE<1:0>:** Watchdog Timer Enable bits

`11` =WDT enabled in all modes, the SEN bit in the WDTCON0 register is ignored
`10` =WDT enabled while running and disabled in Sleep
`01` =WDT controlled by the SEN bit in the WDTCON0 register
`00` = WDT disabled

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

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

WDTCPS <4:0>	WDTPS at POR				Software control of WDTPS	
	Value	Divider Ratio		Typical time out (F_{IN} = 31 kHz)		
`11111`	`01011`	1:65536	2^{16}	2 s	Yes	Default fuse = `11111`

`10011`
`...`
`11110`
	`10011`
	`...`
	`11110`
		1:32	2^{5}	1 ms	No	

`10010`	`10010`	1:8388608	2^{23}	256 s	No	
`10001`	`10001`	1:4194304	2^{22}	128 s		
`10000`	`10000`	1:2097152	2^{21}	64 s		
`01111`	`01111`	1:1048576	2^{20}	32 s		
`01110`	`01110`	1:524299	2^{19}	16 s		
`01101`	`01101`	1:262144	2^{18}	8 s		
`01100`	`01100`	1:131072	2^{17}	4 s		
`01011`	`01011`	1:65536	2^{16}	2 s		
`01010`	`01010`	1:32768	2^{15}	1 s		
`01001`	`01001`	1:16384	2^{14}	512 ms		
`01000`	`01000`	1:8192	2^{13}	256 ms		
`00111`	`00111`	1:4096	2^{12}	128 ms		
`00110`	`00110`	1:2048	2^{11}	64 ms		
`00101`	`00101`	1:1024	2^{10}	32 ms		
`00100`	`00100`	1:512	2^{9}	16 ms		
`00011`	`00011`	1:256	2^{8}	8 ms		
`00010`	`00010`	1:128	2^{7}	4 ms		
`00001`	`00001`	1:64	2^{6}	2 ms		
`00000`	`00000`	1:32	2^{5}	1 ms		

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

4.3 Code Protection

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

4.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{\text{CP}}$ 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 boot loader software, can be protected while allowing other regions of the program memory to be modified.

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

4.5 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 10.4 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the "*PIC12(L)F1612/16(L)F161X Memory Programming Specification*" (DS40001720).

4.6 Device ID and Revision ID

The 14-bit Device ID word is located at 8006h and the 14-bit Revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. See **Section 10.4 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

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

4.7 Register Definitions: Device ID

REGISTER 4-4: DEVID: DEVICE ID REGISTER

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

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

Legend:	
R = Readable bit	
'1' = Bit is set	'0' = Bit is cleared

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

Device	DEVID<13:0> Values
PIC12F1612	`11 0000 0101 1000` (3058h)
PIC12LF1612	`11 0000 0101 1001` (3059h)
PIC16F1613	`11 0000 0100 1100` (304Ch)
PIC16LF1613	`11 0000 0100 1101` (304Dh)

REGISTER 4-5: REVID: REVISION ID REGISTER

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

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

Legend:	
R = Readable bit	
'1' = Bit is set	'0' = Bit is cleared

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

5.0 OSCILLATOR MODULE

5.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 5-1 illustrates a block diagram of the oscillator module.

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

- Selectable system clock source between external or internal sources via software.

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

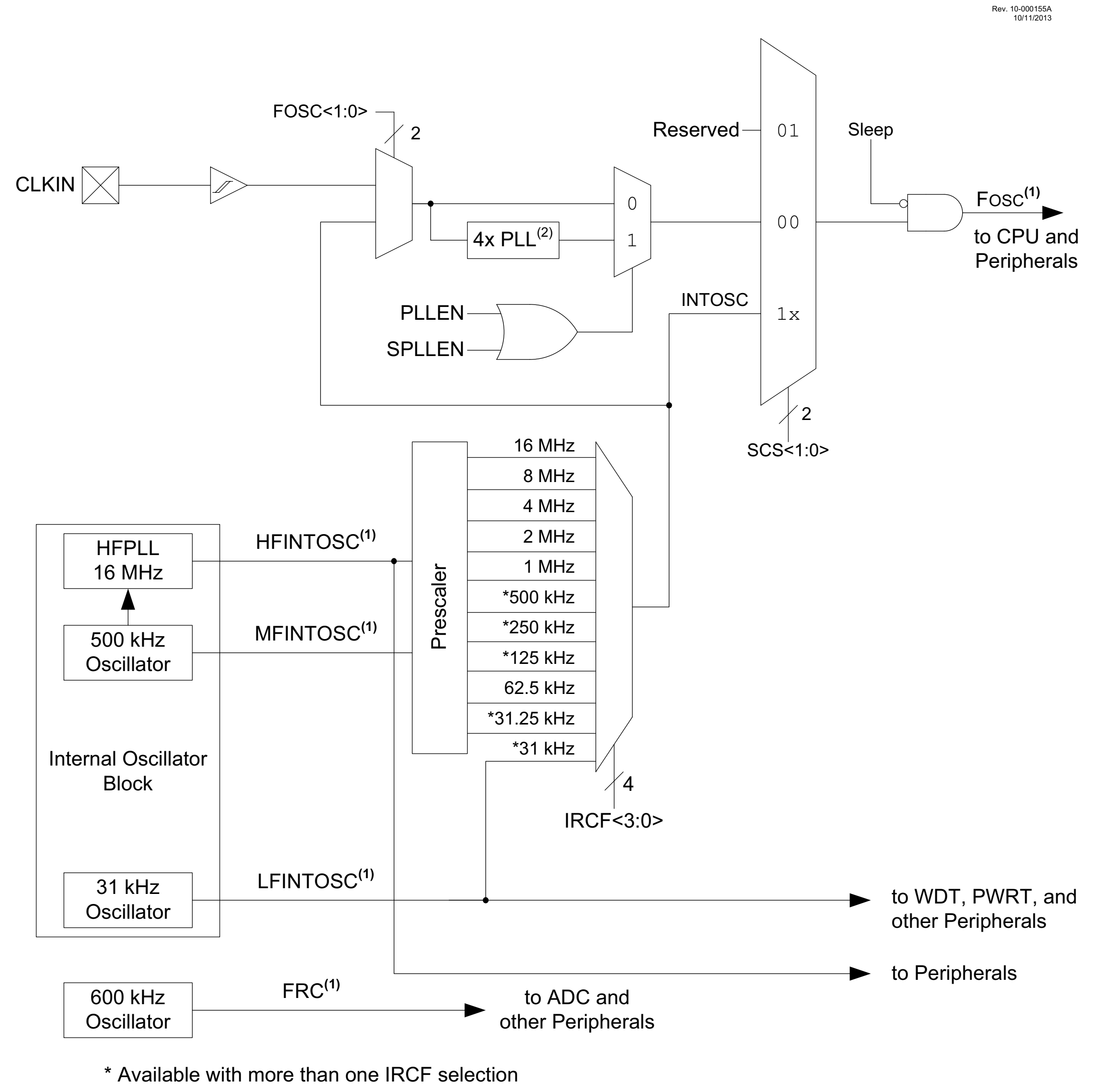
1. ECL – External Clock Low-Power mode (0 MHz to 0.5 MHz)
2. ECM – External Clock Medium-Power mode (0.5 MHz to 4 MHz)
3. ECH – External Clock High-Power mode (4 MHz to 32 MHz)
4. INTOSC – Internal oscillator (31 kHz to 32 MHz).

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: 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 **Section5.3 "Clock Switching"** for additional information.

5.2.1 EXTERNAL CLOCK SOURCES

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

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

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

5.2.1.1 EC Mode

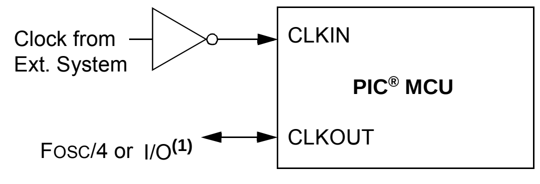
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. CLKOUT is available for general purpose I/O or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

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

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

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

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

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

The function of the OSC2/CLKOUT pin is determined by the $\overline{\text{CLKOUTEN}}$ 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 **Section5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information.

The HFINTOSC is enabled by:

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

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

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

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

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

5.2.2.2 MFINTOSC

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

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

The MFINTOSC is enabled by:

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

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

5.2.2.3 Internal Oscillator Frequency Adjustment

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

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

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

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

5.2.2.4 LFINTOSC

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

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

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

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

Peripherals that use the LFINTOSC are:

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

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

5.2.2.5 FRC

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

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

The FRC clock will continue to run during Sleep.

5.2.2.6 Internal Oscillator Frequency Selection

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

The postscaler outputs of the 16 MHz HFINTOSC, 500 kHz MFINTOSC, and 31 kHz LFINTOSC output connect to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

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

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

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

5.2.2.7 32 MHz Internal Oscillator Frequency Selection

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

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

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

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

5.2.2.8 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-3). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

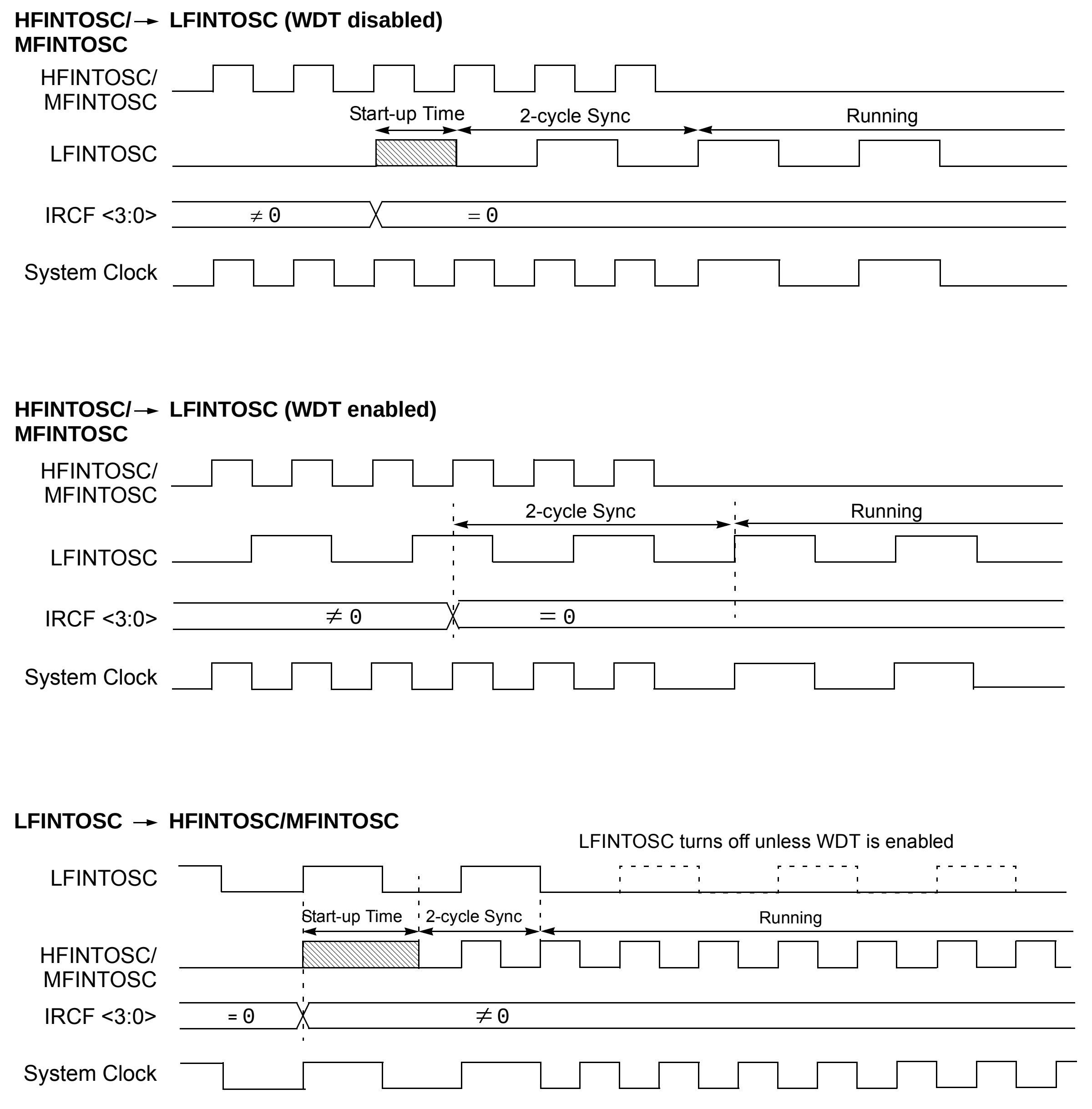
1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.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

See Figure 5-3 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 5-1.

Start-up delay specifications are located in the oscillator tables of **Section28.0 “Electrical Specifications”**.

FIGURE 5-3: INTERNAL OSCILLATOR SWITCH TIMING



5.3 Clock Switching

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

- Default system oscillator determined by FOSC bits in Configuration Words
- Internal Oscillator Block (INTOSC)

5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

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

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

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

TABLE 5-1: OSCILLATOR SWITCHING DELAYS

Switch From	Switch To	Frequency	Oscillator Delay
Sleep	LFINTOSC(1) MFINTOSC(1) HFINTOSC(1)	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	Oscillator Warm-up Delay (Tiosc st)
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

<u>If PLLEN in Configuration Words = 1:</u>
SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements)

<u>If PLLEN in Configuration Words = 0:</u>
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 HF
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 (defaults to internal oscillator block)
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	U-0	R-0/q	R-0/q	R-q/q	R-0/q	R-0/q
—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = 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<2:0> bits of the Configuration Words
`0` = Running from an internal oscillator (FOSC<2:0> = 100)

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 stable
`0` = HFINTOSC is not stable

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>		66
OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	67
OSCTUNE	—	—	TUN<5:0>						68

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>		—	52
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	—	FOSC<1:0>		

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

6.0 RESETS

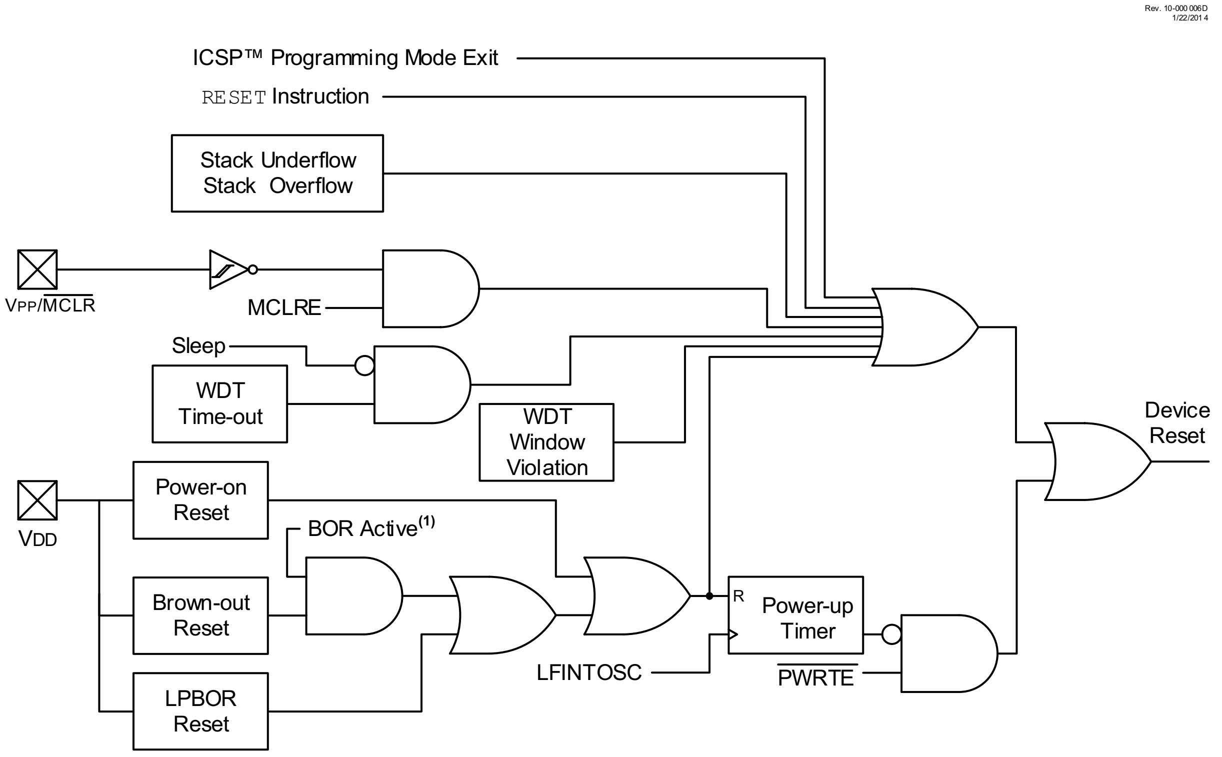
There are multiple ways to reset this device:

- 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.

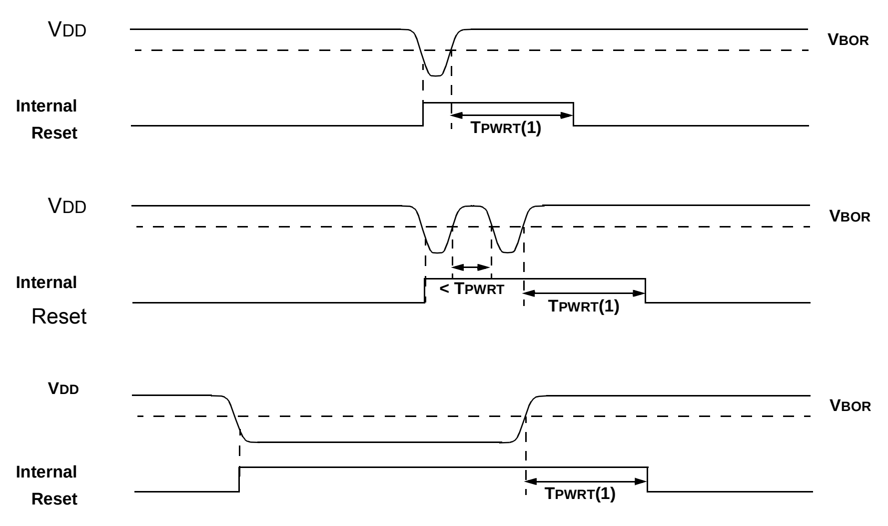
6.2.3 BOR CONTROLLED BY SOFTWARE

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 **Section12.3 “PORTA Registers”** for more information.

6.6 Watchdog Timer (WDT) Reset

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

6.7 RESET Instruction

A `RESET` instruction will cause a device Reset. The 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 **Section3.5.2 “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

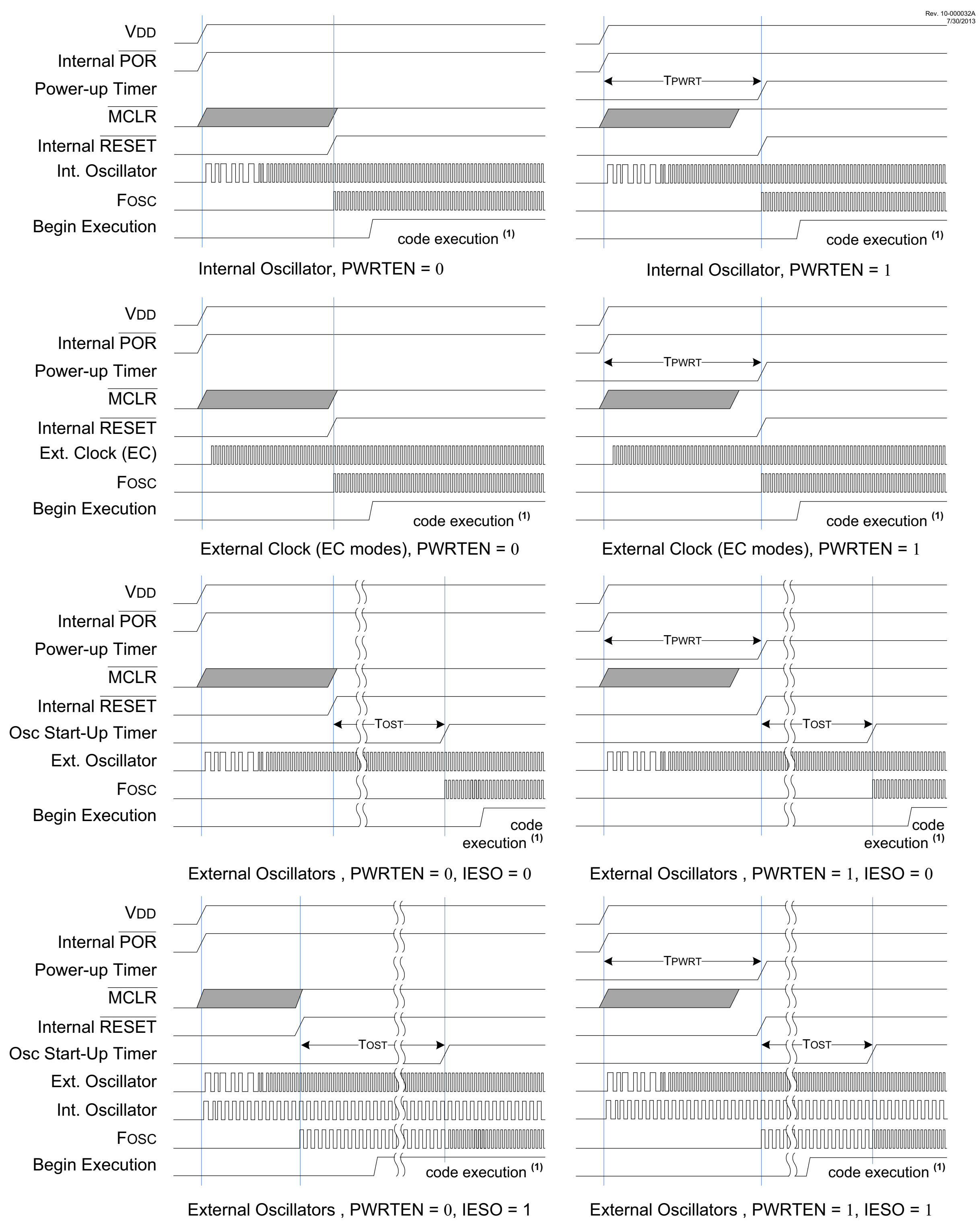
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

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 **Section5.0 “Oscillator Module”** for more information.

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

FIGURE 6-3: RESET START-UP SEQUENCE



Internal Oscillator, PWRTEN = 0

Internal Oscillator, PWRTEN = 1

External Clock (EC modes), PWRTEN = 0

External Clock (EC modes), PWRTEN = 1

External Oscillators , PWRTEN = 0, IESO = 0

External Oscillators , PWRTEN = 1, IESO = 0

External Oscillators , PWRTEN = 0, IESO = 1

External Oscillators , PWRTEN = 1, IESO = 1

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	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	0	1	1	1	0	x	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	1	1	1	0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
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	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	u	0	u	u	u	1	0	$\overline{\text{MCLR}}$ 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	0011 110x
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	---u uuuu	uuuu 0uuu
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	---1 0uuu	uuuu 0uuu
WDT Reset	0000h	---0 uuuu	uuu0 uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uuuu uuuu
Brown-out Reset	0000h	---1 1uuu	00uu 11u0
Interrupt Wake-up from Sleep	PC + 1[(1)]	---1 0uuu	uuuu uuuu
RESET Instruction Executed	0000h	---u uuuu	uuuu u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1uuu uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1uu uuuu
WDT Window Violation	0000h	---1 uuuu	uu0u uuuu

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

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

6.13 Power Control (PCON) Register

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

- Power-On Reset ($\overline{\text{POR}}$)
- 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	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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	**$\overline{\text{WDTWV}}$:** WDT Window Violation Flag bit `1` = A WDT Window Violation Reset has not occurred or set by firmware `0` = A WDT Window Violation Reset has occurred (a `CLRWDT` instruction was executed either without arming the window or outside the window (cleared by hardware)
bit 4	**$\overline{\text{RWDT}}$:** 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	71
PCON	STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	75
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	21
WDTCON0	—	—	WDTPS<4:0>					SEN	99

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>		—	52
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOR}}$	BORV	STVREN	PLLEN	53
	7:0	$\overline{\text{ZCD}}$	—	—	—	—	—	WRT<1:0>		
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			53
	7:0	—	WDTE<1:0>		WDTCPS<4:0>					

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.

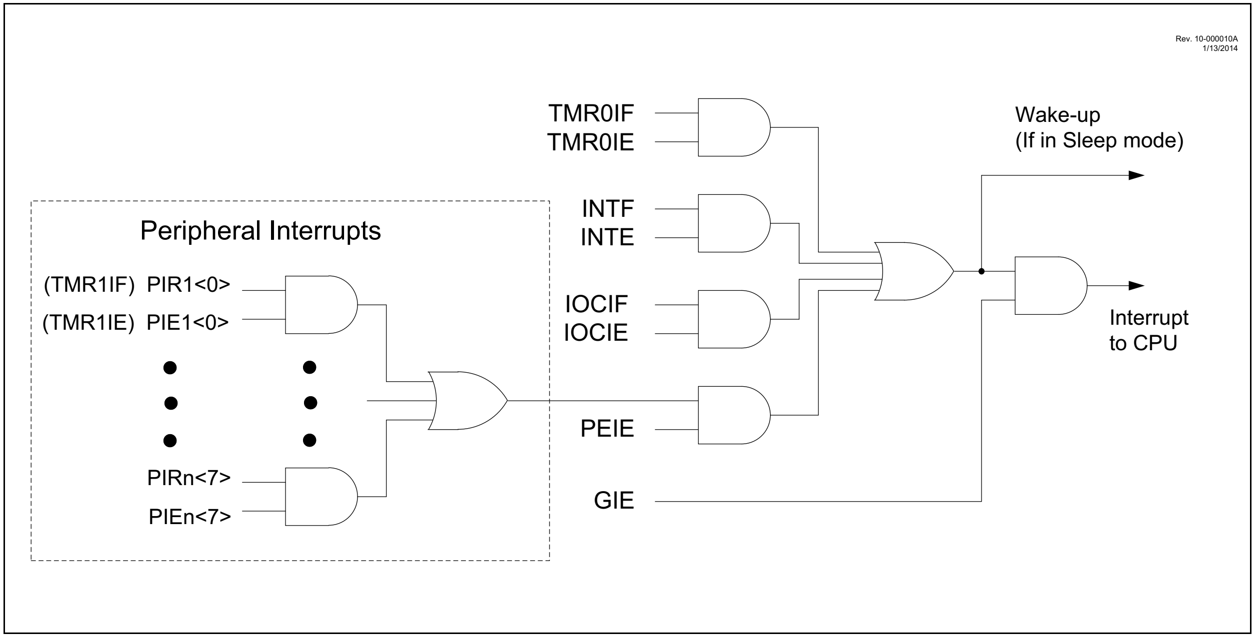
This chapter contains the following information for Interrupts:

- 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 **"Section7.5 "Automatic Context Saving"."**)
- PC is loaded with the interrupt vector 0004h

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

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

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

Note 1: Individual 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

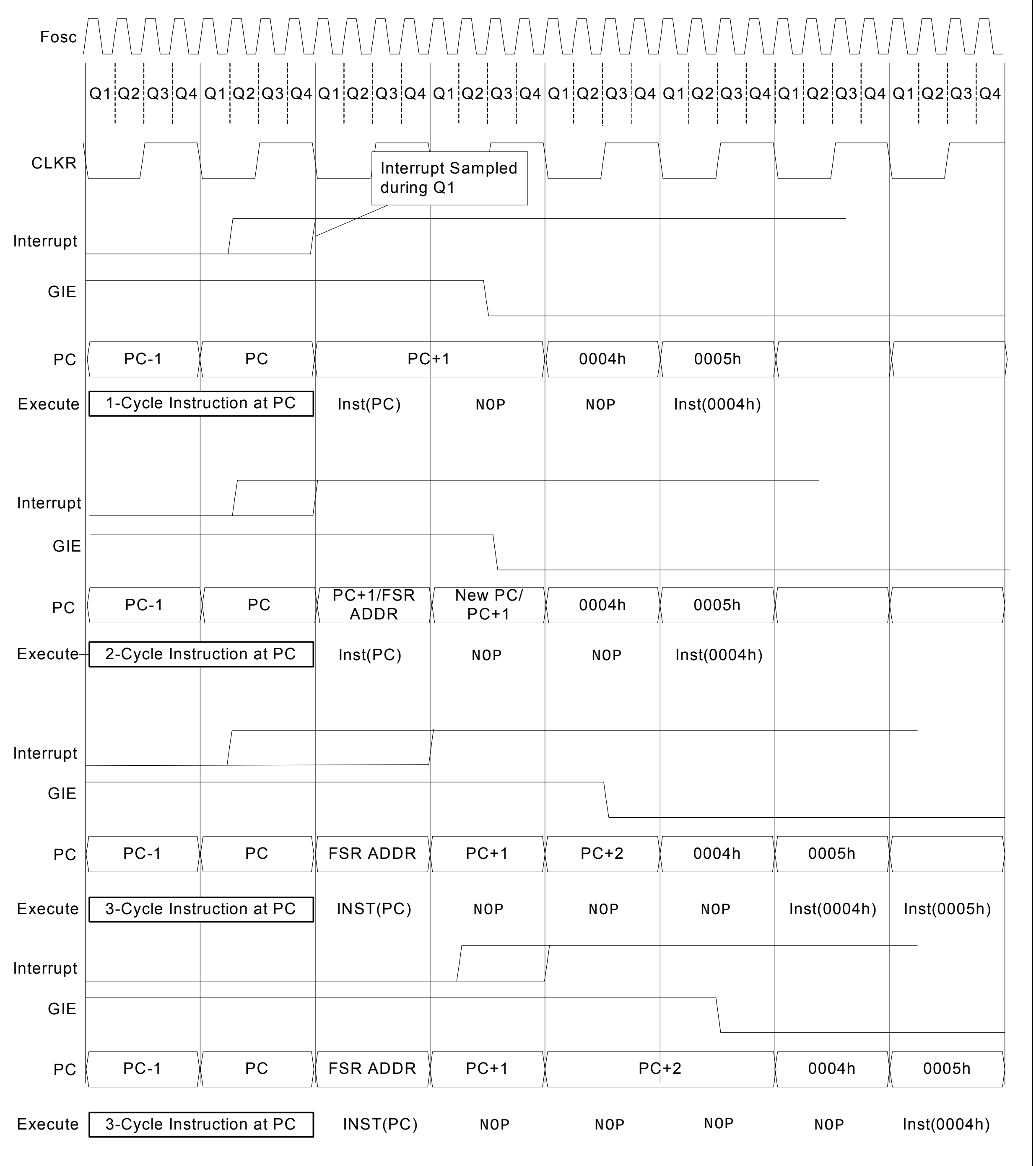
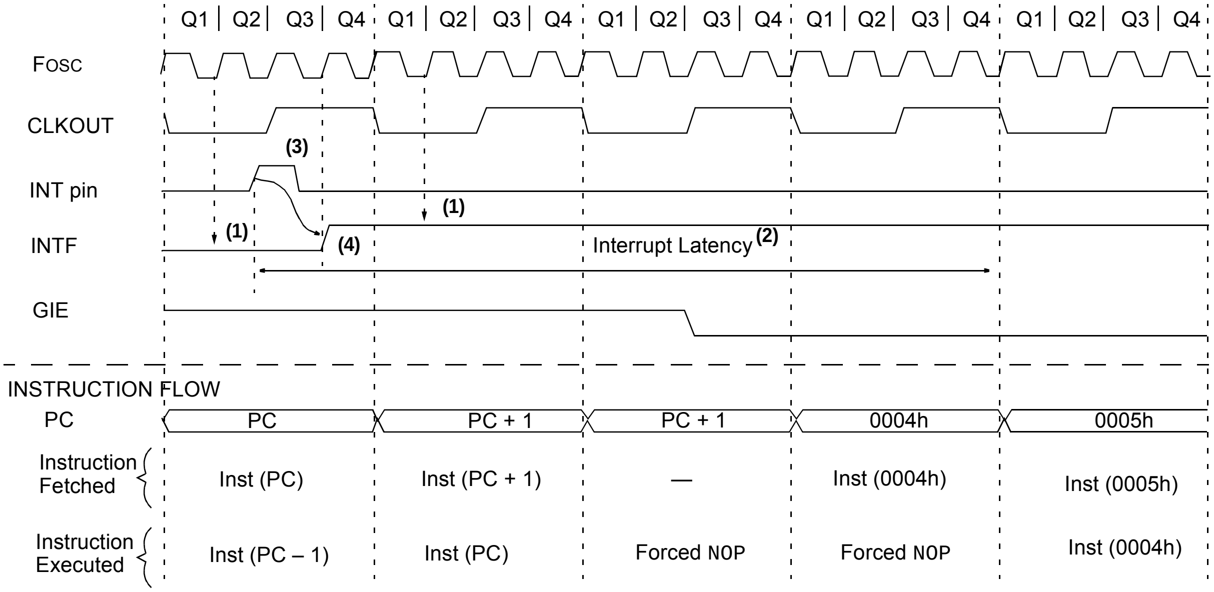


FIGURE 7-3: INT PIN INTERRUPT TIMING



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 **Section28.0 “Electrical Specifications”**.

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

7.3 Interrupts During Sleep

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

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the `SLEEP` instruction. The instruction directly after the `SLEEP` instruction will always be executed before branching to the ISR. Refer to **Section8.0 "Power-Down Mode (Sleep)"** for more details.

7.4 INT Pin

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

7.5 Automatic Context Saving

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

- W register
- STATUS register (except for $\overline{\text{TO}}$ 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	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	—	—	—	CCP1IE	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 5-3 **Unimplemented:** Read as ‘0’

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit

1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

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	R/W-0/0	R/W-0/0	R/W-0/0
—	C2IE[1]	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE
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]
`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 4-3 **Unimplemented:** Read as ‘`0`’

bit 2 **TMR6IE:** TMR6 to PR6 Match Interrupt Enable bit
`1` = Enables the Timer6 to PR6 match interrupt
`0` = Disables the Timer6 to PR6 match interrupt

bit 1 **TMR4IE:** TMR4 to PR4 Match Interrupt Enable bit
`1` = Enables the Timer4 to PR4 match interrupt
`0` = Disables the Timer4 to PR4 match interrupt

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
`1` = The CCP2 interrupt is enabled
`0` = The CCP2 interrupt is not enabled

Note 1: PIC16(L)F1613 only.
2: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

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

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	CWGIE	ZCDIE	—	—	—	—
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 **CWGIE**: Complementary Waveform Generator (CWG) Interrupt Enable bit

`1` = Enables the CWG interrupt
`0` = Disables the CWG interrupt

bit 4 **ZCDIE**: Zero-Cross Detection (ZCD) Interrupt Enable bit

`1` = Enables the ZCD interrupt
`0` = Disables the ZCD 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: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4

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
SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE
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 **SCANIE**: Scanner Interrupt Enable bit
`1` = Enables the scanner interrupt
`0` = Disables the scanner interrupt

bit 6 **CRCIE**: CRC Interrupt Enable bit
`1` = Enables the CRC interrupt
`0` = Disables the CRC interrupt

bit 5 **SMT2PWAIE**: SMT2 Pulse Width Acquisition Interrupt Enable bit
`1` = Enables the SMT2 acquisition interrupt
`0` = Disables the SMT2 acquisition interrupt

bit 4 **SMT2PRAIE**: SMT2 Period Acquisition Interrupt Enable bit
`1` = Enables the SMT2 acquisition interrupt
`0` = Disables the SMT2 acquisition interrupt

bit 3 **SMT2IE**: SMT2 Match Interrupt Enable bit
`1` = Enables the SMT2 period match interrupt
`0` = Disables the SMT2 period match interrupt

bit 2 **SMT1PWAIE**: SMT1 Pulse Width Acquisition Interrupt Enable bit
`1` = Enables the SMT1 acquisition interrupt
`0` = Disables the SMT1 acquisition interrupt

bit 1 **SMT1PRAIE**: SMT1 Period Acquisition Interrupt Enable bit
`1` = Enables the SMT1 acquisition interrupt
`0` = Disables the SMT1 acquisition interrupt

bit 0 **SMT1IE**: SMT1 Match Interrupt Enable bit
`1` = Enables the SMT1 period match interrupt
`0` = Disables the SMT1 period match interrupt

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

REGISTER 7-6: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	—	—	—	CCP1IF	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 5-3	**Unimplemented:** Read as ‘`0`’
bit 2	**CCP1IF:** CCP1 Interrupt Flag bit `1` = Interrupt is pending `0` = Interrupt is not pending
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.

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

U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	C2IF(1)	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF
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)
`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 4-3 **Unimplemented:** Read as ‘`0`’

bit 2 **TMR6IF:** Timer6 to PR6 Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 1 **TMR4IF:** Timer4 to PR4 Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 0 **CCP2IF**: CCP2 Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

Note 1: PIC16(L)F1613 only.

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

REGISTER 7-8: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	CWGIF	ZCDIF	—	—	—	—
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	**CWGIF**: CWG Interrupt Flag bit `1` = Interrupt is pending `0` = Interrupt is not pending
bit 4	**ZCDIF**: ZCD Interrupt Flag bit `1` = Interrupt is pending `0` = Interrupt is not pending
bit 3-0	**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 Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 7-9: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4

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
SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
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 **SCANIF**: Scanner Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 6 **CRCIF**: CRC Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 5 **SMT2PWAIF**: SMT2 Pulse Width Acquisition Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 4 **SMT2PRAIF**: SMT2 Period Acquisition Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 3 **SMT2IF**: SMT2 Match Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 2 **SMT1PWAIF**: SMT1 Pulse Width Acquisition Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 1 **SMT1PRAIF**: SMT1 Period Acquisition Interrupt Flag bit
`1` = Interrupt is pending
`0` = Interrupt is not pending

bit 0 **SMT1IF**: SMT1 Match 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 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	82
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			190
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	83
PIE2	—	C2IE[(1)]	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	84
PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	85
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IF	86
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	87
PIR2	—	C2IF[(1)]	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	88
PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	89
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	90

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by interrupts.
Note 1: PIC16(L)F1613 only.

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
   - Timer1 oscillator
7. ADC is unaffected, if the dedicated FRC oscillator is selected.
8. I/O ports maintain the status they had before `SLEEP` was executed (driving high, low or 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 modules 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.

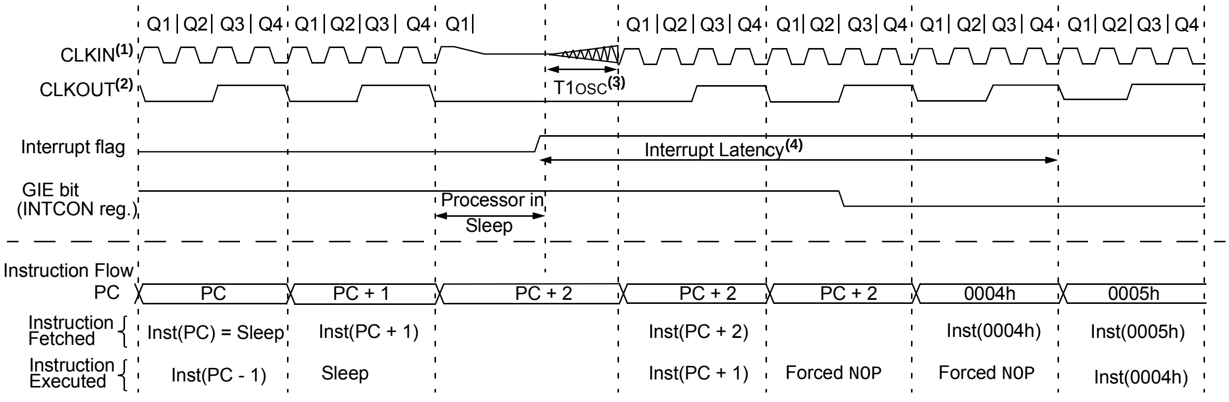
8.1.1 WAKE-UP USING INTERRUPTS

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; See **Section 28.0 “Electrical Specifications”**.
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) can utilize the HFINTOSC oscillator as either a clock source or as an input source. Under certain conditions, when the HFINTOSC is selected for use with the CWG modules, the HFINTOSC will remain active during Sleep. This will have a direct effect on the Sleep mode current.

Please refer to sections **Section 24.11 “Operation During Sleep”** for more information.

Note: The PIC12LF1612/16LF1613 does not have a configurable Low-Power Sleep mode. PIC12LF1612/16LF1613 is an unregulated device and is always in the lowest power state when in Sleep, with no wake-up time penalty. This device has a lower maximum V_{DD} and I/O voltage than the PIC12F1612/16F1613. See **Section 28.0 “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(2) Draws higher current in Sleep, faster wake-up
bit 0	**Reserved:** Read as ‘`1`’. Maintain this bit set.

Note 1: PIC12F1612/16F1613 only.

2: See **Section 28.0 “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	82
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	148
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	148
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	148
IOCCP(1)	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	148
IOCCN(1)	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	148
IOCCF(1)	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	148
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	83
PIE2	—	C2IE(1)	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	84
PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	85
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IF	86
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	87
PIR2	—	C2IF(1)	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	88
PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	89
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	90
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	21
WDTCON0	—	—	WDTPS<4:0>					SEN	99

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

Note 1: PIC16(L)F1613 only.

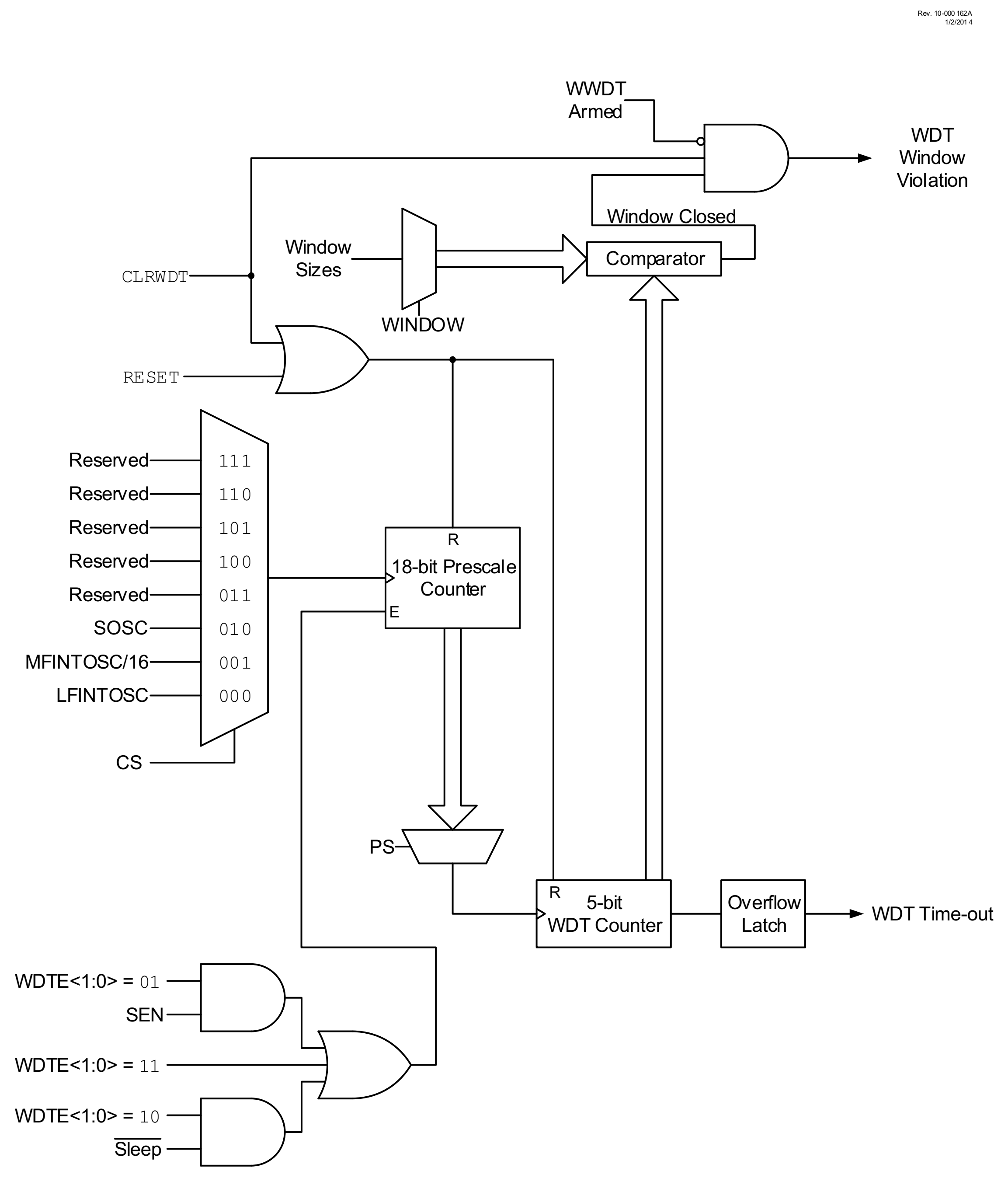
9.0 WINDOWED WATCHDOG TIMER (WDT)

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

The WDT has the following features:

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

FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM



9.1 Independent Clock Source

The WDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of either the WDTCCS<2:0> configuration bits or the WDTCS<2:0> bits of WDTCON1. Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section28.0 “Electrical Specifications”** for LFINTOSC and MFINTOSC tolerances.

9.2 WDT Operating Modes

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

9.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to ‘`11`’, the WDT is always on.

WDT protection is active during Sleep.

9.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to ‘`10`’, the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

9.2.3 WDT CONTROLLED BY SOFTWARE

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

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

TABLE 9-1: WDT OPERATING MODES

WDTE<1:0>	SEN	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 WDTCON0 register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is two seconds.

9.4 Watchdog Window

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

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

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

9.5 Clearing the WDT

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

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

9.5.1 `CLRWDT` CONSIDERATIONS (WINDOWED MODE)

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

See Table 9-2 for more information.

9.6 Operation During Sleep

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

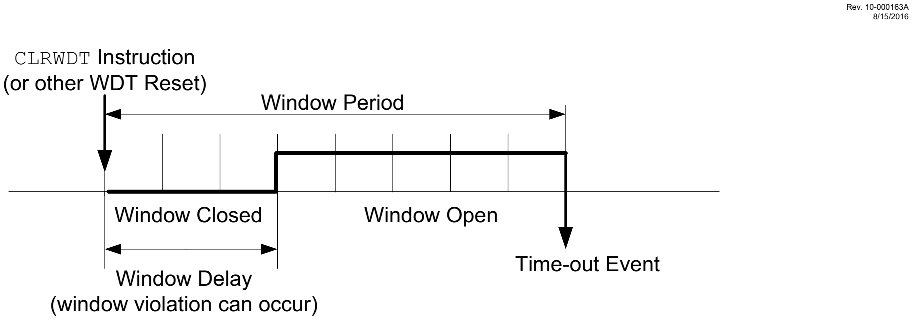
The WDT remains clear until the OST, if enabled, completes. See **Section5.0 “Oscillator Module”** for more information on the OST.

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

TABLE 9-2: WDT CLEARING CONDITIONS

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

FIGURE 9-2: WINDOW PERIOD AND DELAY



9.7 Register Definitions: Windowed Watchdog Timer Control

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

U-0	U-0	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W-0/0
—	—	WDTPS<4:0>(1)					SEN
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 5-1 **WDTPS<4:0>:** Watchdog Timer Prescale 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 **SEN:** 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.
2: When WDTCPS <4:0> in CONFIG3 = `11111`, the Reset value of WDTPS<4:0> is `01011`. Otherwise, the Reset value of WDTPS<4:0> is equal to WDTCPS<4:0> in CONFIG3.
3: When WDTCPS <4:0> in CONFIG3 ≠ `11111`, these bits are read-only.

REGISTER 9-2: WDTCON1: WATCHDOG TIMER CONTROL REGISTER 1

U-0	R/W(3)-q/q(1)	R/W(3)-q/q(1)	R/W(3)-q/q(1)	U-0	R/W(4)-q/q(2)	R/W(4)-q/q(2)	R/W(4)-q/q(2)
—	WDTCS<2:0>			—	WINDOW<2:0>		
bit 7							bit 0

Legend:

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

bit 7 **Unimplemented:** Read as ‘`0`’

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

`111` = Reserved
•
•
•
`010` = Reserved
`001` = MFINTOSC 31.25 kHz
`000` = LFINTOSC 31 kHz

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

bit 2-0 **WINDOW<2:0>:** Watchdog Timer Window Select bits

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

Note 1: If WDTCCS <2:0> in CONFIG3 = `111`, the Reset value of WDTCS<2:0> is `000`.

2: The Reset value of WINDOW<2:0> is determined by the value of WDTCWS<2:0> in the CONFIG3 register.

3: If WDTCCS<2:0> in CONFIG3 ≠ `111`, these bits are read-only.

4: If WDTCWS<2:0> in CONFIG3 ≠ `111`, these bits are read-only.

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

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<7:0>[(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 7-0 **PSCNT<7:0>**: Prescale Select Low Byte bits[(1)]

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

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

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<15:8>[(1)]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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 **PSCNT<15:8>**: Prescale Select High Byte bits[(1)]

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

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

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
WDTTMR<3:0>					STATE	PSCNT<17:16>[(1)]	
bit 7							bit 0

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

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

bit 2 **STATE**: WDT Armed Status bit
`1` = WDT is armed
`0` = WDT is not armed

bit 1-0 **PSCNT<17:16>**: Prescale Select Upper Byte bits[(1)]

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

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>		66
PCON	STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	75
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	21
WDTCON0	—	—	WDTPS<4:0>					SEN	99
WDTCON1	—	WDTCS<2:0>			—	WINDOW<2:0>			99
WDTPSL	PSCNT<7:0>								99
WDTPSH	PSCNT<15:8>								99
WDTTMR	—	WDTTMR<4:0>				STATE	PSCNT<17:16>		99

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>		—	52
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	—	FOSC<1:0>		
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			53
	7:0	—	WDTE<1:0>		WDTCPS<4:0>					

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 VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

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

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

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

The Flash program memory can be protected in two ways; by code protection ($\overline{\text{CP}}$ 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)
PIC12(L)F1612	16	16
PIC16(L)F1613		

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

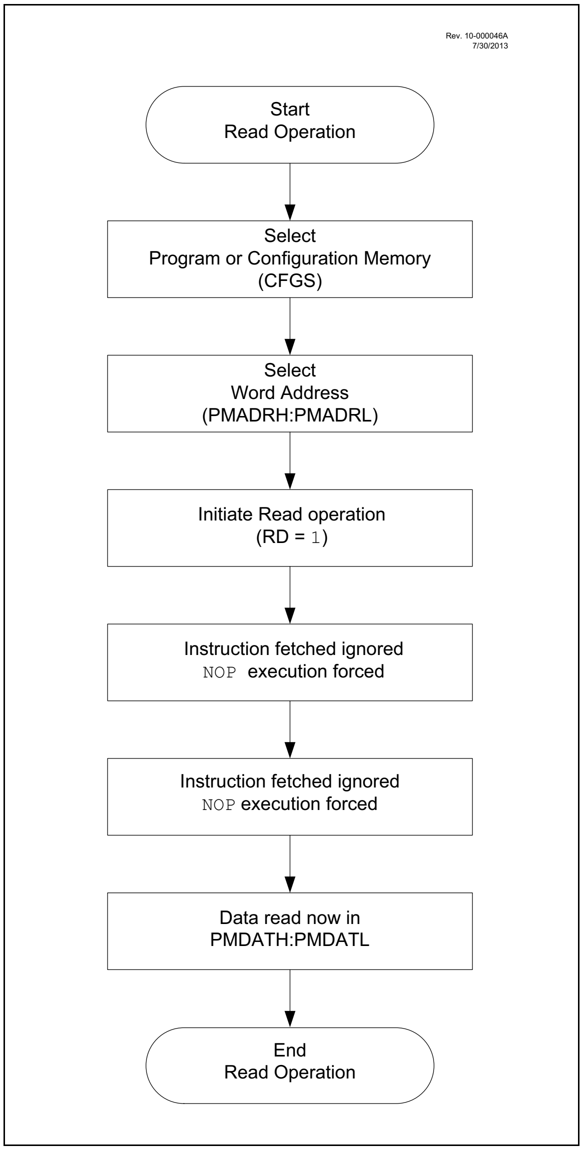
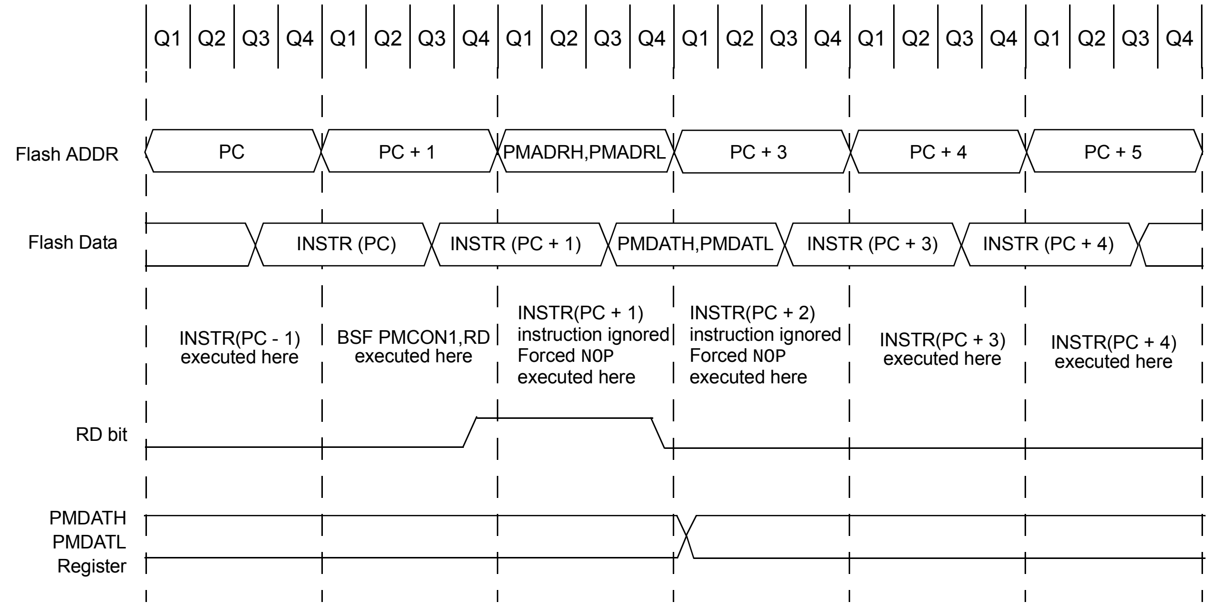


FIGURE 10-2: FLASH PROGRAM MEMORY READ CYCLE EXECUTION



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

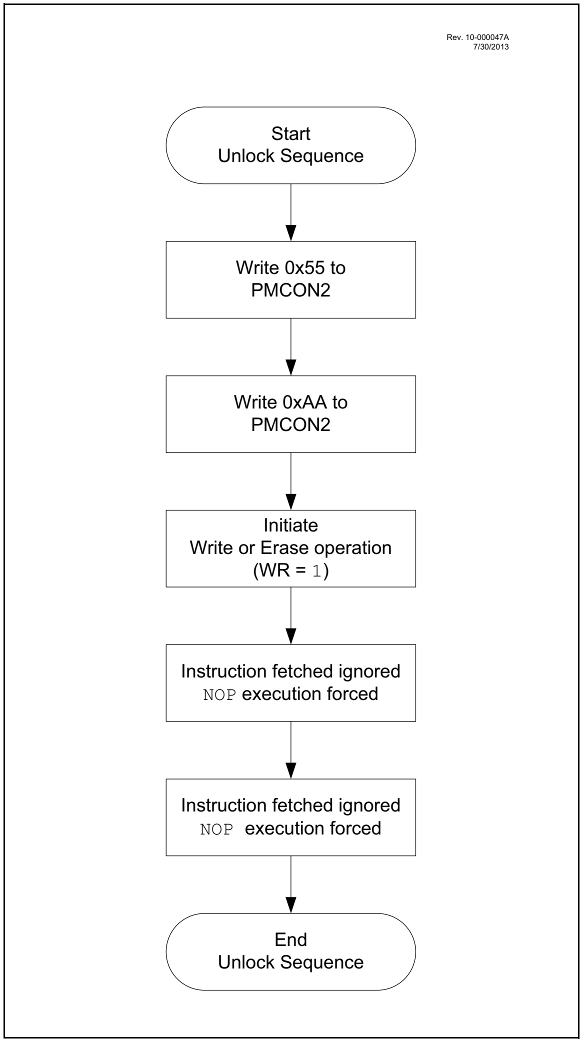
The unlock sequence consists of the following steps:

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

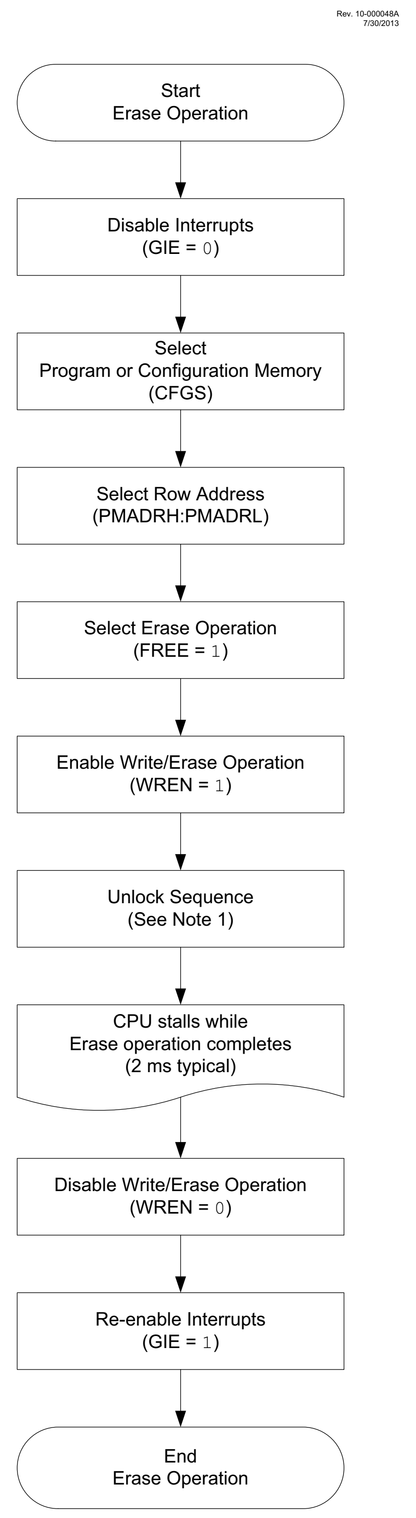
While executing code, program memory can only be erased by rows. To erase a row:

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 immediately following the WR bit set instruction. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction.

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

       MOVLW       55h             ; Start of required sequence to initiate erase
       MOVWF       PMCON2          ; Write 55h
       MOVLW       AAh             ;
       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
```

Required Sequence

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 four 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 16 WRITE LATCHES

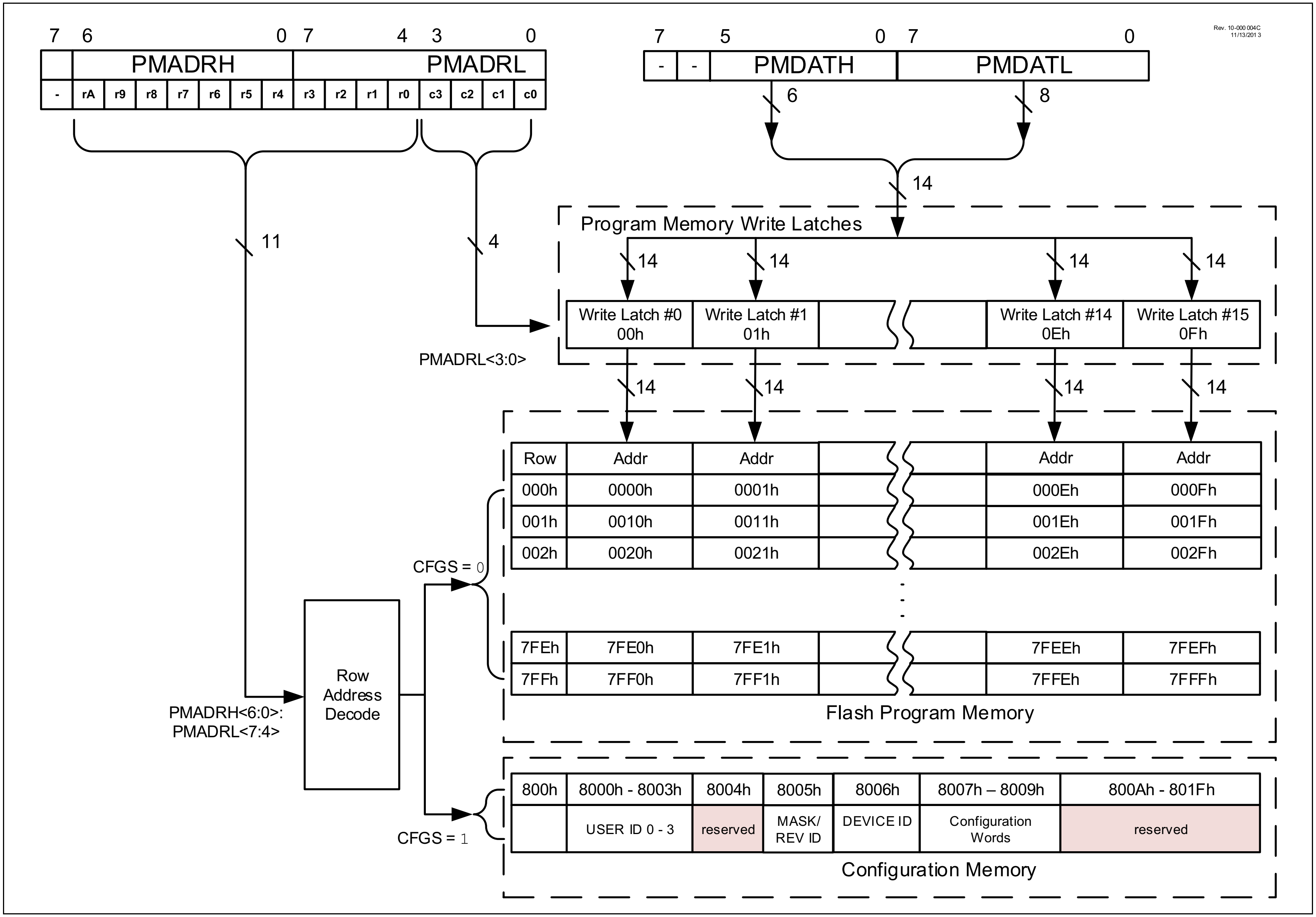
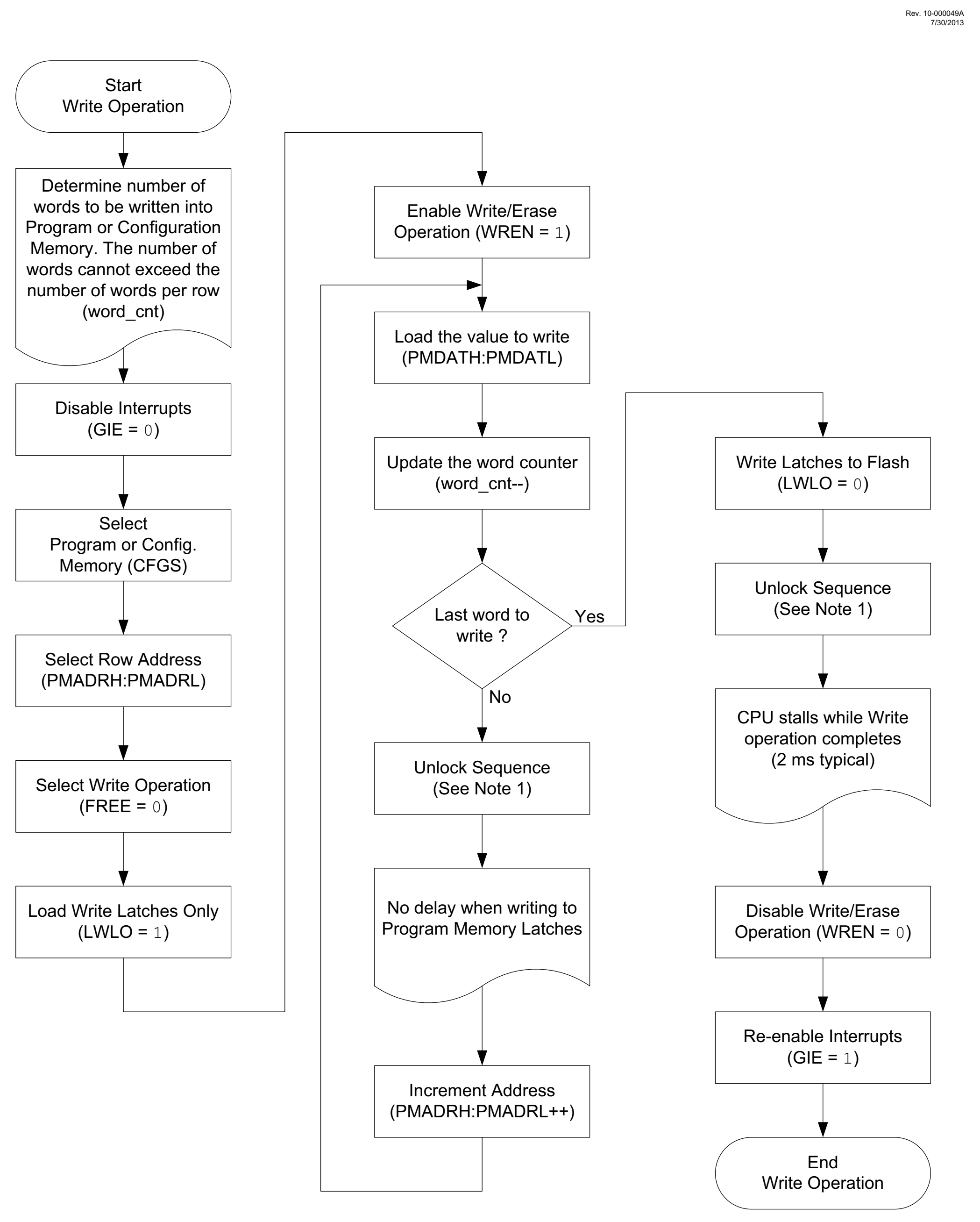


FIGURE 10-6: FLASH PROGRAM MEMORY WRITE FLOWCHART



Note 1: See Figure 10-3.

EXAMPLE 10-3: WRITING TO FLASH PROGRAM MEMORY (16 WRITE LATCHES)

```
; This write routine assumes the following:
; 1. 32 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       0x0F            ; Check if we're on the last of 16 addresses
        ANDLW       0x0F            ;
        BTFSC       STATUS,Z        ; Exit if last of 16 words,
        GOTO        START_WRITE     ;
```

Required Sequence

```
        MOVLW       55h             ; Start of required write sequence:
        MOVWF       PMCON2          ; Write 55h
        MOVLW       AAh             ;
        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

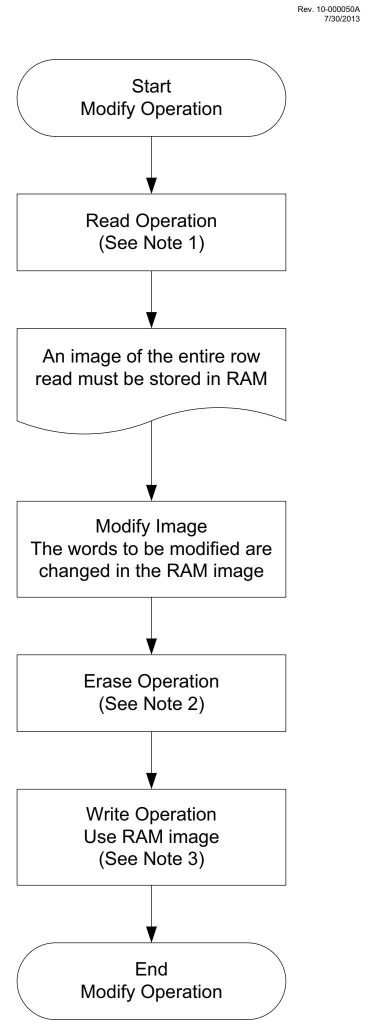
```
        MOVLW       55h             ; Start of required write sequence:
        MOVWF       PMCON2          ; Write 55h
        MOVLW       AAh             ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin write
        NOP                         ; 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 10-5.

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-8009h	Configuration Words 1, 2, and 3	Yes	No

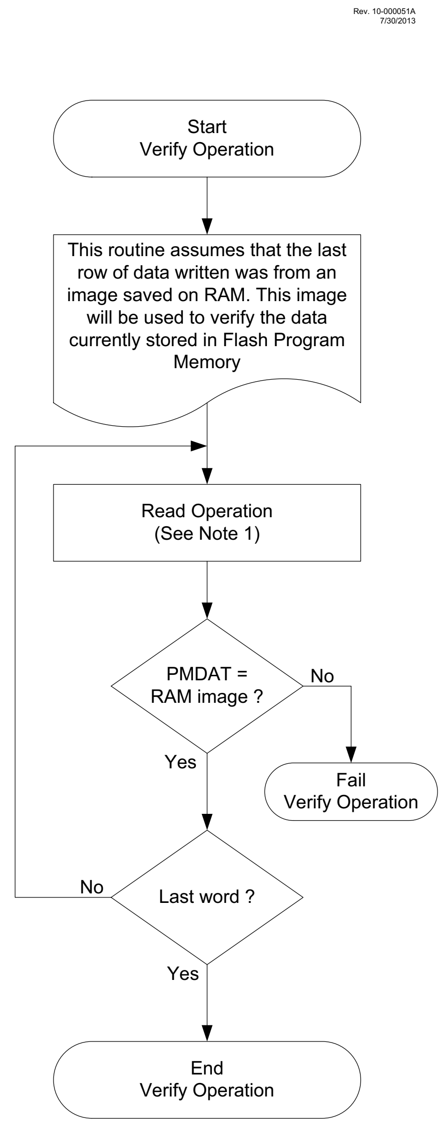
EXAMPLE 10-4: CONFIGURATION WORD AND DEVICE ID ACCESS

```
* 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 10-2.

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	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 a write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit
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	82
PMCON1	—[1]	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	117
PMCON2	Program Memory Control Register 2								118
PMADRL	PMADRL<7:0>								116
PMADRH	—[1]	PMADRH<6:0>							116
PMDATL	PMDATL<7:0>								116
PMDATH	—	—	PMDATH<5:0>						116

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>		—	52
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOR}}$	BORV	STVREN	PLLEN	53
	7:0	$\overline{\text{ZCD}}$	—	—	—	—	—	WRT<1:0>		
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			53
	7:0	—	WDTE<1:0>		WDTCPS<4:0>					

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

11.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE

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

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

11.1 CRC Module Overview

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

11.2 CRC Functional Overview

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

EXAMPLE 11-1:

Rev. 10-000206A
1/8/2014

CRC-16-ANSI

$x^{16} + x^{15} + x^{2} + 1$ (17 bits)

Standard 16-bit representation = 0x8005

`CRCXORH = 0b10000000`
`CRCXORL = 0b0000010-` (1)

Data Sequence:
0x55, 0x66, 0x77, 0x88

`DLEN = 0b0111`
`PLEN = 0b1111`

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

SHIFTM = 1:
`10101010 01100110 11101110 00010001`

Check Value (ACCM = 1):

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

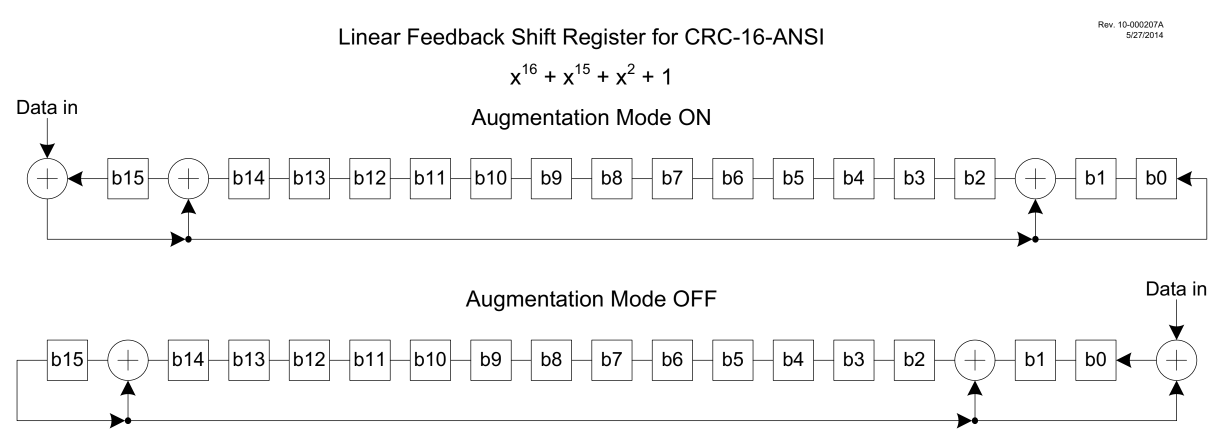
SHIFTM = 1: `0x6BA2`
`CRCACCH = 0b01101011`
`CRCACCL = 0b10100010`

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

11.3 CRC Polynomial Implementation

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

EXAMPLE 11-2: CRC LFSR EXAMPLE



11.4 CRC Data Sources

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

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

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

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

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

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

11.4.1 CRC FROM USER DATA

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

11.4.2 CRC FROM FLASH

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

11.5 CRC Check Value

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

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

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

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

11.6 CRC Interrupt

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

11.7 Configuring the CRC

The following steps illustrate how to properly configure the CRC.

1. Determine if the automatic Program Memory scan will be used with the Scanner or manual calculation through the SFR interface and perform the actions specified in **Section11.4 "CRC Data Sources"**, depending on which decision was made.
2. If desired, seed a starting CRC value into the CRCACCH/L registers.
3. Program the CRCXORH/L registers with the desired generator polynomial.
4. Program the DLEN<3:0> bits of the CRCCON1 register with the length of the data word - 1 (refer to Example 11-1). This determines how many times the shifter will shift into the accumulator for each data word.
5. Program the PLEN<3:0> bits of the CRCCON1 register with the length of the polynomial - 2 (refer to Example 11-1).
6. Determine whether shifting in trailing zeros is desired and set the ACCM bit of CRCCON0 register appropriately.
7. Likewise, determine whether the MSb or LSb should be shifted first and write the SHIFTM bit of CRCCON0 register appropriately.
8. Write the CRCGO bit of the CRCCON0 register to begin the shifting process.

9a. If manual SFR entry is used, monitor the FULL bit of CRCCON0 register. When FULL = `0`, another word of data can be written to the CRCDATH/L registers, keeping in mind that CRCDATH should be written first if the data has >8 bits, as the shifter will begin upon the CRCDATL register being written.

9b. If the scanner is used, the scanner will automatically stuff words into the CRCDATH/L registers as needed, as long as the SCANGO bit is set.

10a. If using the Flash memory scanner, monitor the SCANIF (or the SCANGO bit) for the scanner to finish pushing information into the CRCDATA registers. After the scanner is completed, monitor the CRCIF (or the BUSY bit) to determine that the CRC has been completed and the check value can be read from the CRCACC registers. If both the interrupt flags are set (or both BUSY and SCANGO bits are cleared), the completed CRC calculation can be read from the CRCACCH/L registers.

10b. If manual entry is used, monitor the CRCIF (or BUSY bit) to determine when the CRCACC registers will hold the check value.

11.8 Program Memory Scan Configuration

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

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

11.9 Scanner Interrupt

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

11.10 Scanning Modes

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

11.10.1 BURST MODE

When MODE = `01`, the scanner is in Burst mode. In Burst mode, CPU operation is stalled beginning with the operation after the one that sets the SCANGO bit, and the scan begins, using the instruction clock to execute.

The CPU is held until the scan stops. Note that because the CPU is not executing instructions, the SCANGO bit cannot be cleared in software, so the CPU will remain stalled until one of the hardware end-conditions occurs. Burst mode has the highest throughput for the scanner, but has the cost of stalling other execution while it occurs.

11.10.2 CONCURRENT MODE

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

11.10.3 TRIGGERED MODE

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

11.10.4 PEEK MODE

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

TABLE 11-1: SUMMARY OF SCANNER MODES

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

11.10.5 INTERRUPT INTERACTION

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

TABLE 11-2: SCAN INTERRUPT MODES

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

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

11.10.6 WDT INTERACTION

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

11.10.7 IN-CIRCUIT DEBUG (ICD) INTERACTION

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

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

TABLE 11-3: ICD AND SCANNER INTERACTIONS

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

11.11 Register Definitions: CRC and Scanner Control

REGISTER 11-1: CRCCON0: CRC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R-0	R/W-0/0	U-0	U-0	R/W-0/0	R-0
EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL
bit 7							bit 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 **EN:** CRC Enable bit
`1` = CRC module is released from Reset
`0` = CRC is disabled and consumes no operating current

bit 6 **CRCGO:** CRC Start bit
`1` = Start CRC serial shifter
`0` = CRC serial shifter turned off

bit 5 **BUSY:** CRC Busy bit
`1` = Shifting in progress or pending
`0` = All valid bits in shifter have been shifted into accumulator and EMPTY = `1`

bit 4 **ACCM:** Accumulator Mode bit
`1` = Data is augmented with zeros
`0` = Data is not augmented with zeros

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

bit 1 **SHIFTM:** Shift Mode bit
`1` = Shift right (LSb)
`0` = Shift left (MSb)

bit 0 **FULL:** Data Path Full Indicator bit
`1` = CRCDATH/L registers are full
`0` = CRCDATH/L registers have shifted their data into the shifter

REGISTER 11-2: CRCCON1: CRC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
DLEN<3:0>				PLEN<3:0>			
bit 7							bit 0

Legend:		
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 **DLEN<3:0>:** Data Length bits
Denotes the length of the data word -1 (See Example 11-1)

bit 3-0 **PLEN<3:0>:** Polynomial Length bits
Denotes the length of the polynomial -1 (See Example 11-1)

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

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
DAT<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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	**DAT<15:8>**: CRC Input/Output Data bits

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

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
DAT<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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	**DAT<7:0>**: CRC Input/Output Data bits Writing to this register fills the shifter.

REGISTER 11-5: CRCACCH: CRC ACCUMULATOR HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACC<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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	**ACC<15:8>**: CRC Accumulator Register bits Writing to this register writes to the CRC accumulator register. Reading from this register reads the CRC accumulator.

REGISTER 11-6: CRCACCL: CRC ACCUMULATOR LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACC<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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	**ACC<7:0>**: CRC Accumulator Register bits Writing to this register writes to the CRC accumulator register through the CRC write bus. Reading from this register reads the CRC accumulator.

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

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

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

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<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	**SHIFT<7:0>**: CRC Shifter Register bits Reading from this register reads the CRC Shifter.

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

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
XOR<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = 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	**XOR<15:8>**: XOR of Polynomial Term X_N Enable bits

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

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

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

bit 7-1	**XOR<7:1>**: XOR of Polynomial Term X_N Enable bits
bit 0	**Unimplemented:** Read as ‘`0`’

REGISTER 11-11: SCANCON0: SCANNER ACCESS CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	R-0	R-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
EN(1)	SCANGO(2, 3)	BUSY(4)	INVALID	INTM	—	MODE<1:0>(5)	
bit 7							bit 0

Legend:		
R = 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 **EN:** Scanner Enable bit(1)

`1` = Scanner is enabled
`0` = Scanner is disabled, internal states are reset

bit 6 **SCANGO:** Scanner GO bit(2, 3)

`1` = When the CRC sends a ready signal, NVM will be accessed according to MDx and data passed to the client peripheral.
`0` = Scanner operations will not occur

bit 5 **BUSY:** Scanner Busy Indicator bit(4)

`1` = Scanner cycle is in process
`0` = Scanner cycle is complete (or never started)

bit 4 **INVALID:** Scanner Abort signal bit

`1` = SCANLADRL/H has incremented or contains an invalid address(6)
`0` = SCANLADRL/H points to a valid address

bit 3 **INTM:** NVM Scanner Interrupt Management Mode Select bit

If MODE = `10`:

This bit is ignored

If MODE = `01` (CPU is stalled until all data is transferred):

`1` = SCANGO is overridden (to zero) during interrupt operation; scanner resumes after returning from interrupt
`0` = SCANGO is not affected by interrupts, the interrupt response will be affected

If MODE = `00` or `11`:

`1` = SCANGO is overridden (to zero) during interrupt operation; scan operations resume after returning from interrupt
`0` = Interrupts do not prevent NVM access

bit 2 **Unimplemented:** Read as ‘`0`’

bit 1-0 **MODE<1:0>:** Memory Access Mode bits(5)

`11` = Triggered mode
`10` = Peek mode
`01` = Burst mode
`00` = Concurrent mode

Note 1: Setting EN = `0` (SCANCON0 register) does not affect any other register content.
2: This bit is cleared when LADR > HADR (and a data cycle is not occurring).
3: If INTM = `1`, this bit is overridden (to zero, but not cleared) during an interrupt response.
4: BUSY = `1` when the NVM is being accessed, or when the CRC sends a ready signal.
5: See Table 11-1 for more detailed information.
6: An invalid address happens when the entire range of the PFM is scanned and completed, i.e., device memory is 0x4000 and SCANHADR = 0x3FFF, after the last scan SCANLADR increments to 0x4000, the address is invalid.

REGISTER 11-12: SCANLADRH: SCAN LOW ADDRESS HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LADR<15:8>[(1, 2)]							
bit 7							bit 0

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

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

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

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

REGISTER 11-13: SCANLADRL: SCAN LOW ADDRESS LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LADR<7:0>[(1, 2)]							
bit 7							bit 0

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

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

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

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

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

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
HADR<15:8>[(1, 2)]							
bit 7							bit 0

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

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

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

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

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

REGISTER 11-15: SCANHADRL: SCAN HIGH ADDRESS LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
HADR<7:0>[(1, 2)]							
bit 7							bit 0

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

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

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

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

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

REGISTER 11-16: SCANTRIG: SCAN TRIGGER SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	TSEL<3:0>			
bit 7							bit 0

Legend:

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

bit 7-4 **Unimplemented:** Read as ‘`0`’

bit 3-0 **TSEL<3:0>**: Scanner Data Trigger Input Selection bits
`1111-1010` = Reserved
`1001` = SMT2_Match
`1000` = SMT1_Match
`0111` = TMR0_Overflow
`0110` = TMR5_Overflow
`0101` = TMR3_Overflow
`0100` = TMR1_Overflow
`0011` = TMR6_postscaled
`0010` = TMR4_postscaled
`0001` = TMR2_postscaled
`0000` = LFINTOSC

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH CRC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CRCACCH	ACC<15:8>								125
CRCACCL	ACC<7:0>								125
CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	124
CRCCON1	DLEN<3:0>				PLEN<3:0>				124
CRCDATH	DAT<15:8>								125
CRCDATL	DAT<7:0>								125
CRCSHIFTH	SHIFT<15:8>								126
CRCSHIFTL	SHIFT<7:0>								126
CRCXORH	XOR<15:8>								126
CRCXORL	XOR<7:1>							—	126
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	90
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	86
SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		127
SCANHADRH	HADR<15:8>								129
SCANHADRL	HADR<7:0>								129
SCANLADRH	LADR<15:8>								128
SCANLADRL	LADR<7:0>								128
SCANTRIG					TSEL<3:0>				130

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

12.0 I/O PORTS

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

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- INLVLx (input level control)
- ODCONx registers (open-drain)
- SLRCONx registers (slew rate)

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

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

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

TABLE 12-1: PORT AVAILABILITY PER DEVICE

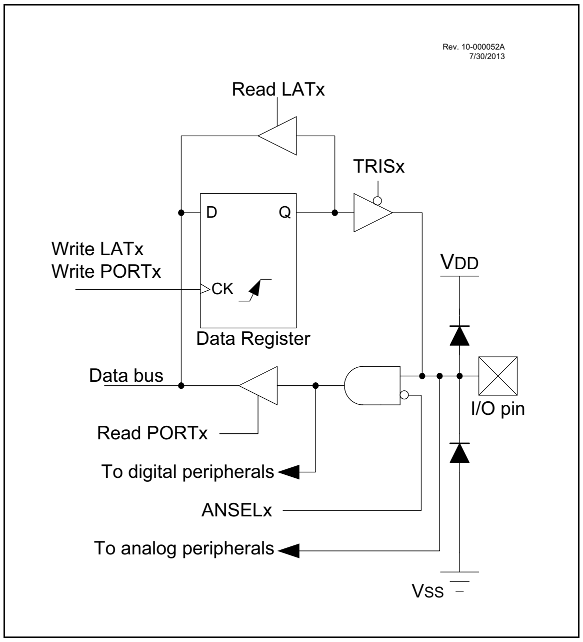
Device	PORTA	PORTC
PIC16(L)F1613	●	●
PIC12(L)F1612	●	

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 12-1.

FIGURE 12-1: GENERIC I/O PORT OPERATION



12.1 Alternate Pin Function

The Alternate Pin Function Control (APFCON) register is used to steer specific peripheral input and output functions between different pins. The APFCON register is shown in Register 12-1. For this device family, the following functions can be moved between different pins.

- CWGA
- CWGB
- T1G
- CCP1
- CCP2

These bits have no effect on the values of any TRIS register. PORT and TRIS overrides will be routed to the correct pin. The unselected pin will be unaffected.

12.2 Register Definitions: Alternate Pin Function Control

REGISTER 12-1: APFCON: ALTERNATE PIN FUNCTION CONTROL REGISTER

U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
—	CWGASEL[1]	CWGBSEL[1]	—	T1GSEL	—	CCP2SEL[2]	CCP1SEL[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 **Unimplemented:** Read as ‘0’

bit 6 **CWGASEL:** Pin Selection bit[1]
1 = CWGA function is on RA5
0 = CWGA function is on RA2

bit 5 **CWGBSEL:** Pin Selection bit[1]
1 = CWGB function is on RA4
0 = CWGB function is on RA0

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **T1GSEL:** Pin Selection bit
1 = T1G function is on RA3
0 = T1G function is on RA4

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **CCP2SEL:** Pin Selection bit[2]
1 = CCP2 function is on RA5
0 = CCP2 function is on RC3

bit 0 **CCP1SEL:** Pin Selection bit[1]
1 = CCP1 function is on RA5
0 = CCP1 function is on RA2

Note 1: PIC12(L)F1612 only.
2: PIC16(L)F1613 only.

12.3 PORTA Registers

12.3.1 DATA REGISTER

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 12-3). Setting a TRISA bit (= `1`) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= `0`) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). The exception is RA3, which is input-only and its TRIS bit will always read as ‘`1`’. Example 12-1 shows how to initialize an I/O port.

Reading the PORTA register (Register 12-2) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATA).

12.3.2 DIRECTION CONTROL

The TRISA register (Register 12-3) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read ‘`0`’.

12.3.3 OPEN-DRAIN CONTROL

The ODCONA register (Register 12-7) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONA bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONA bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

12.3.4 SLEW RATE CONTROL

The SLRCONA register (Register 12-8) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONA bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONA bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

12.3.5 INPUT THRESHOLD CONTROL

The INLVLA register (Register 12-9) 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 **28.3 “DC Characteristics”** for more information on threshold levels.

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

12.3.6 ANALOG CONTROL

The ANSELA register (Register 12-5) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as ‘`0`’ and allow analog functions on the pin to operate correctly.

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

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

EXAMPLE 12-1: INITIALIZING PORTA

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

12.3.7 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 12-2.

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

Analog input functions, such as ADC and comparator inputs, are not shown in the priority lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below in Table 12-2.

TABLE 12-2: PORTA OUTPUT PRIORITY (PIC12(L)F1612 ONLY)

Pin Name	Function Priority(1)
RA0	DAC1OUT1 CWG1B(2) CCP2 RA0
RA1	ZCD1OUT RA1
RA2	CWG1A(2) C1OUT CCP1 RA2(2)
RA3	RA3
RA4	CLKOUT CWG1B(3) RA4
RA5	CWG1A(3) CCP1(3) RA5

Note 1: Priority listed from highest to lowest.
2: Default pin (see APFCON register).
3: Alternate pin (see APFCON register).

TABLE 12-3: PORTA OUTPUT PRIORITY (PIC16(L)F1613 ONLY)

Pin Name	Function Priority(1)
RA0	DAC1OUT1 RA0
RA1	ZCD1OUT RA1
RA2	C1OUT RA2(2)
RA3	RA3
RA4	CLKOUT RA4
RA5	CCP2(3) RA5

Note 1: Priority listed from highest to lowest.
2: Default pin (see APFCON register).
3: Alternate pin (see APFCON register).

12.4 Register Definitions: PORTA

REGISTER 12-2: 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 $\geq$ VIH
0 = Port pin is $\leq$ VIL

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

REGISTER 12-3: 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 12-4: 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 12-5: 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 12-6: 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 12-7: 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 12-8: 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 12-9: INLVLA: PORTA INPUT LEVEL CONTROL 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
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
bit 7							bit 0

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

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

bit 5-0 **INLVLA<5:0>:** PORTA Input Level Select bits
For RA<5:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

TABLE 12-4: 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	136
APFCON	—	CWGASEL(2)	CWGBSEL(2)	—	T1GSEL	—	CCP2SEL(3)	CCP1SEL(2)	132
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	138
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	136
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	137
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			190
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	135
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	138
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	135
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	137

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

Note 1: Unimplemented, read as ‘`1`’.

2: PIC12(L)F1612 only.

3: PIC16(L)F1613 only.

TABLE 12-5: 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>		—	52
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	—	FOSC<1:0>		

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

12.5 PORTC Registers (PIC16(L)F1613 only)

12.5.1 DATA REGISTER

PORTC is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 12-11). Setting a TRISC bit (= `1`) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= `0`) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 12-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 12-10) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATC).

12.5.2 DIRECTION CONTROL

The TRISC register (Register 12-11) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read ‘`0`’.

12.5.3 OPEN-DRAIN CONTROL

The ODCONC register (Register 12-15) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONC bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONC bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

12.5.4 SLEW RATE CONTROL

The SLRCONC register (Register 12-16) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONC bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONC bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

12.5.5 INPUT THRESHOLD CONTROL

The INLVLC register (Register 12-17) 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 **28.3 “DC Characteristics”** for more information on threshold levels.

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

12.5.6 ANALOG CONTROL

The ANSELC register (Register 12-13) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC bit high will cause all digital reads on the pin to be read as ‘`0`’ and allow analog functions on the pin to operate correctly.

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

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

12.5.7 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 12-6.

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

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

TABLE 12-6: PORTC OUTPUT PRIORITY

Pin Name	Function Priority[(1)]
RC0	RC0
RC1	RC1
RC2	CWG1D RC2
RC3	CWG1C CCP2[(2)] RC3
RC4	CWG1B C2OUT RC4
RC5	CWG1A CCP1 RC5

Note 1: Priority listed from highest to lowest.
2: Default pin (see APFCON register).

12.6 Register Definitions: PORTC (PIC16(L)F1613 ONLY)

REGISTER 12-10: PORTC: PORTC 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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **RC<5:0>**: PORTC General Purpose I/O Pin bits
`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.

REGISTER 12-11: TRISC: PORTC TRI-STATE 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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **TRISC<5:0>**: PORTC Tri-State Control bits
`1` = PORTC pin configured as an input (tri-stated)
`0` = PORTC pin configured as an output

REGISTER 12-12: LATC: PORTC DATA LATCH 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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **LATC<5:0>**: PORTC Tri-State Control bits
`1` = PORTC pin configured as an input (tri-stated)
`0` = PORTC pin configured as an output

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

REGISTER 12-13: ANSELC: PORTC ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as ‘`0`’

bit 3-0 **ANSC<3:0>**: Analog Select between Analog or Digital Function on pins RC<3:0>, respectively
`1` = Analog input. Pin is assigned as analog input[(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 12-14: WPUC: WEAK PULL-UP PORTC REGISTER(1),(2)

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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **WPUC<5:0>**: Weak Pull-up Register bits
`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.

REGISTER 12-15: ODCONC: PORTC OPEN-DRAIN CONTROL 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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **ODC<5:0>:** PORTC Open Drain Enable bits
For RC<5:0> 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 12-16: SLRCONC: PORTC SLEW RATE 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
—	—	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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **SLRC<5:0>:** PORTC Slew Rate Enable bits
For RC<5:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 12-17: INLVLC: PORTC INPUT LEVEL CONTROL 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
—	—	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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **INLVLC<5:0>:** PORTC Input Level Select bits
For RC<5:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

TABLE 12-7: 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	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	143
APFCON	—	CWGASEL$^{(1)}$	CWGBSEL$^{(1)}$	—	T1GSEL	—	CCP2SEL$^{(2)}$	CCP1SEL$^{(1)}$	132
INLVLC	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	145
LATC	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	143
ODCONC	—	—	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	144
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			190
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	142
SLRCONC	—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	145
TRISC$^{(2)}$	—	—	**TRISC5**	**TRISC4**	**TRISC3**	**TRISC2**	**TRISC1**	**TRISC0**	142
WPUC	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	144

Legend: x = unknown, u = unchanged, - = unimplemented locations read as ‘0’. Shaded cells are not used by PORTC.
Note 1: PIC12(L)F1612 only.
2: PIC16(L)F1613 only.

13.0 INTERRUPT-ON-CHANGE

The PORTA 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.

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 and IOCCFx bits located in the IOCAF 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 and IOCCFx bits.

13.4 Clearing Interrupt Flags

The individual status flags, (IOCAFx 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)

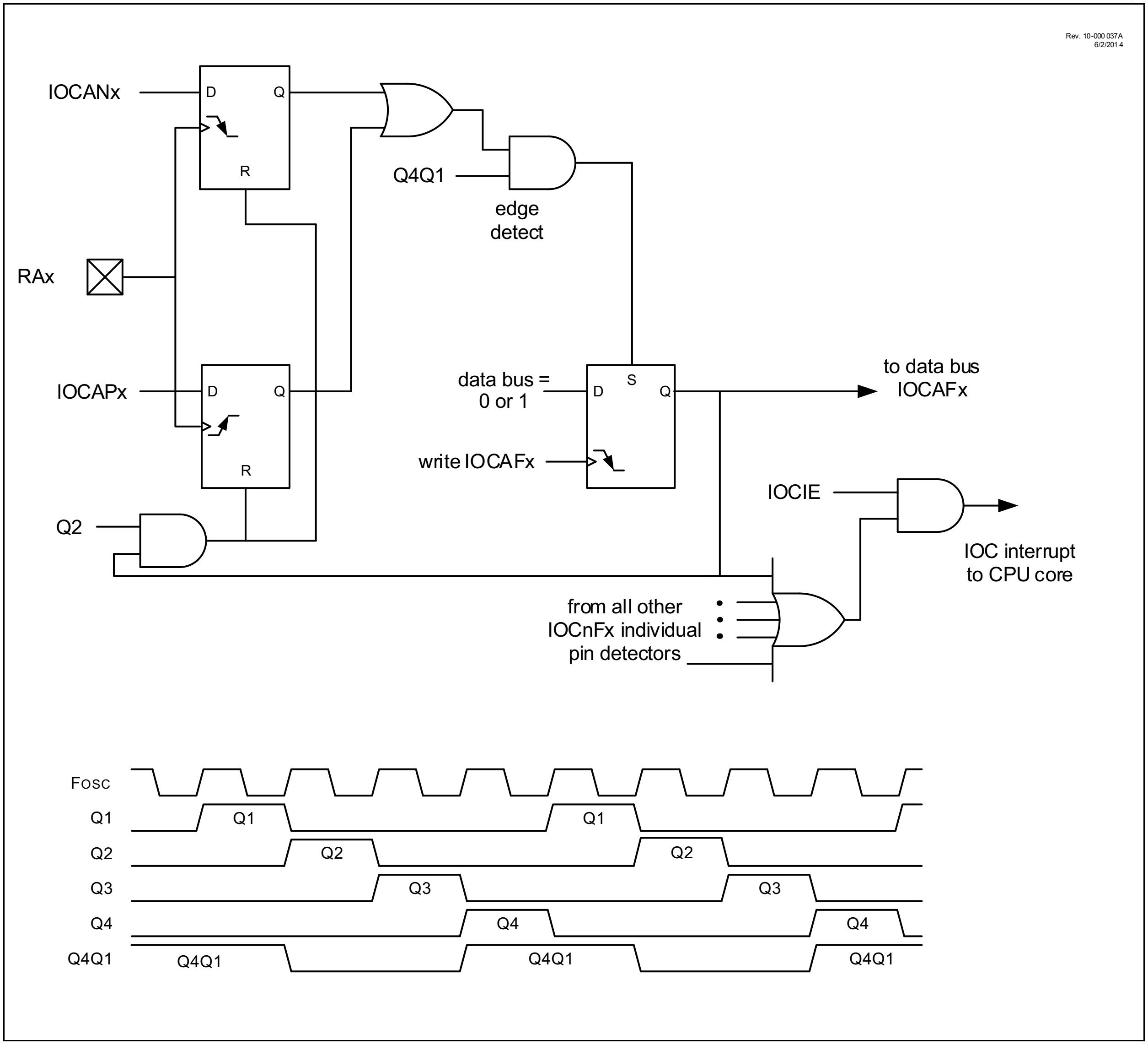
```
MOVLW   0xff
XORWF   IOCAF, W
ANDWF   IOCAF, F
```

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: IOCCP: INTERRUPT-ON-CHANGE PORTC POSITIVE EDGE REGISTER(1)

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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **IOCCP<5:0>: Interrupt-on-Change PORTC Positive Edge Enable bits**

`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: PIC16(L)F1613 only.

REGISTER 13-5: IOCCN: INTERRUPT-ON-CHANGE PORTC NEGATIVE EDGE REGISTER(1)

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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **IOCCN<5:0>: Interrupt-on-Change PORTC Negative Edge Enable bits**

`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: PIC16(L)F1613 only.

REGISTER 13-6: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER(1)

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
—	—	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 7-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **IOCCF<5:0>: Interrupt-on-Change PORTC Flag bits**

`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: PIC16(L)F1613 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	136
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	148
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	148
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	148
IOCCF(2)	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	149
IOCCN(2)	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	149
IOCCP(2)	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	149
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	135
TRISC(2)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	142

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

Note 1: Unimplemented, read as ‘`1`’.

2: 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 **Section16.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

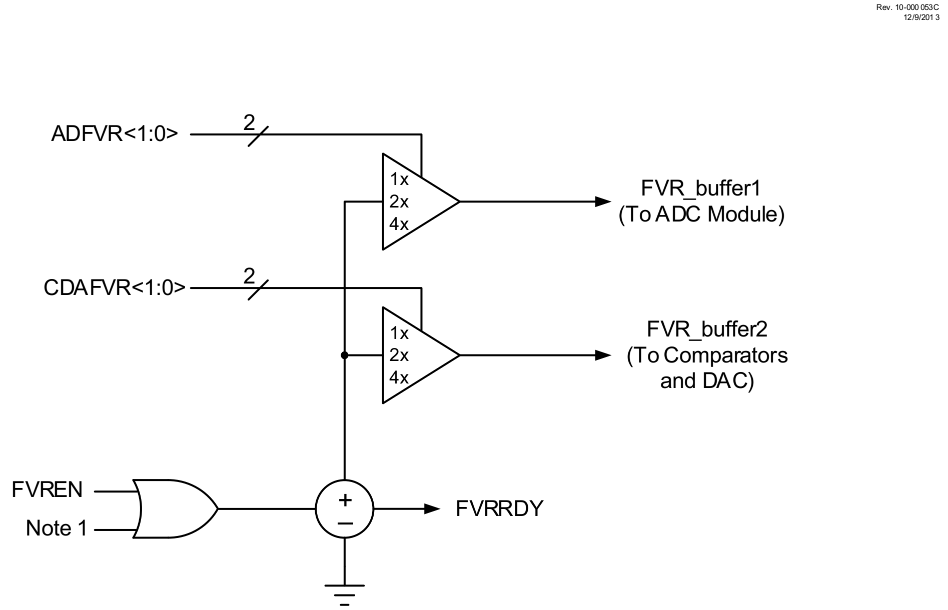
The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the comparator modules. Reference **Section18.0 "Comparator Module"** for additional information.

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

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See Figure 36-64: FVR Stabilization Period, Only.

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 PIC12F1612/16F1613 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 PIC12F1612/16F1613 devices.
3: See **Section15.0 “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>		153

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

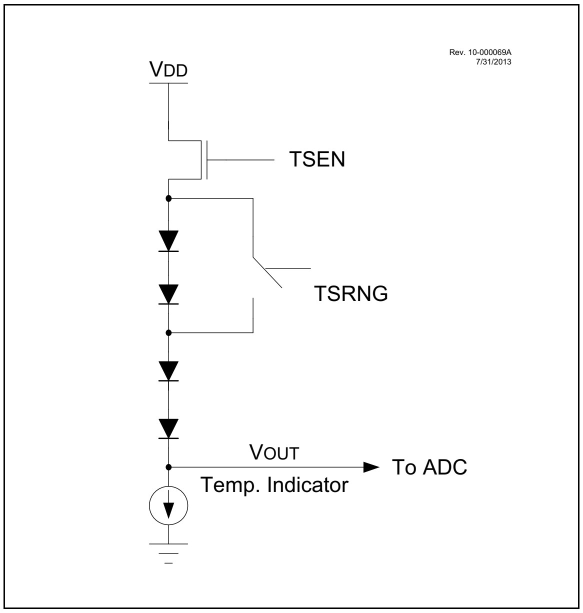
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section14.0 "Fixed Voltage Reference (FVR)"** for more information.

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 **Section16.0 "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.

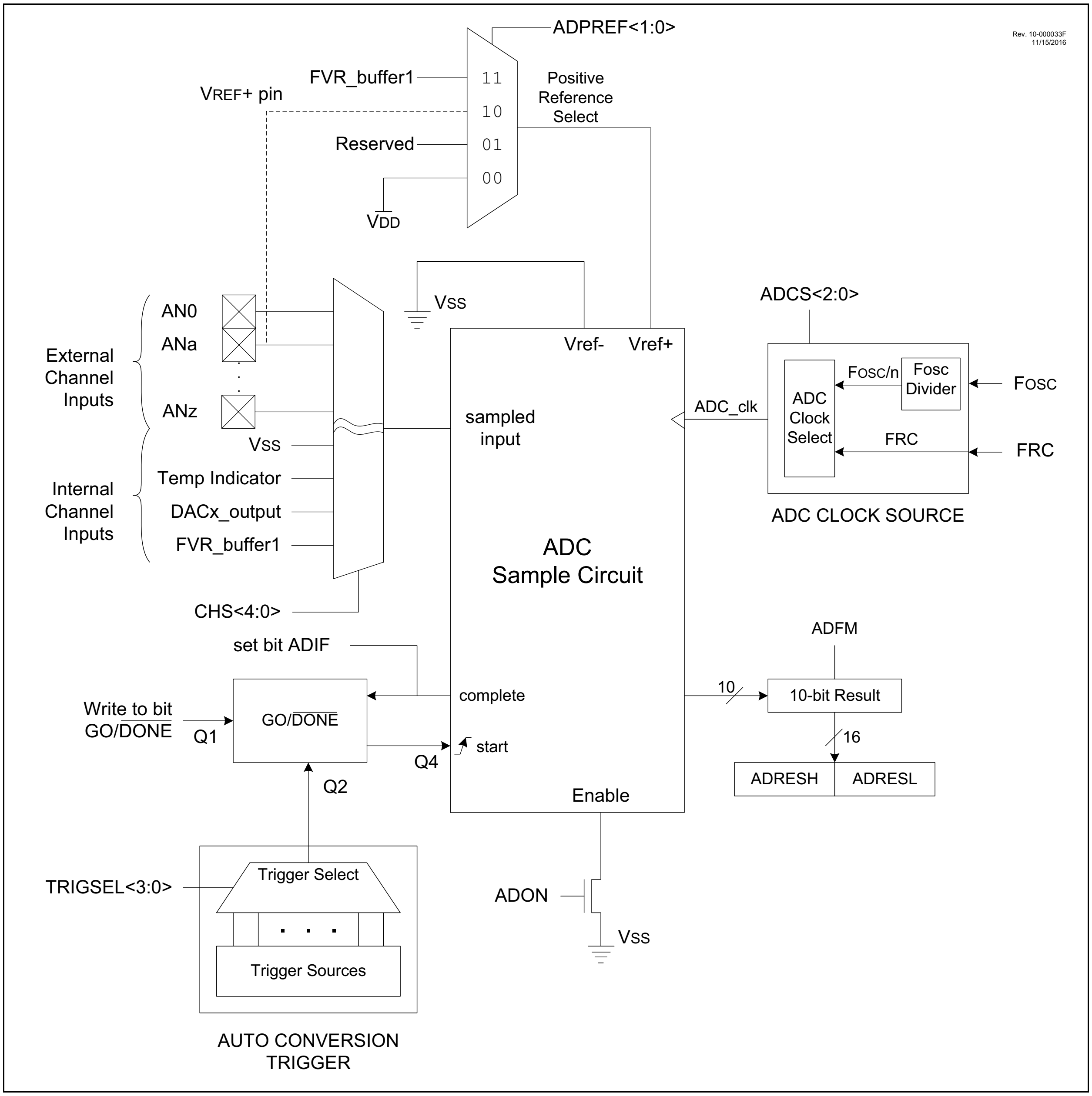
16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

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 **Section12.0 “I/O Ports”** for more information.

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

16.1.2 CHANNEL SELECTION

There are up to 11 channel selections available:

- AN<7:0> pins (PIC16(L)F1613 only)
- AN<3:0> pins (PIC12(L)F1612 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 **Section16.2.6 “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 **Section28.0 “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

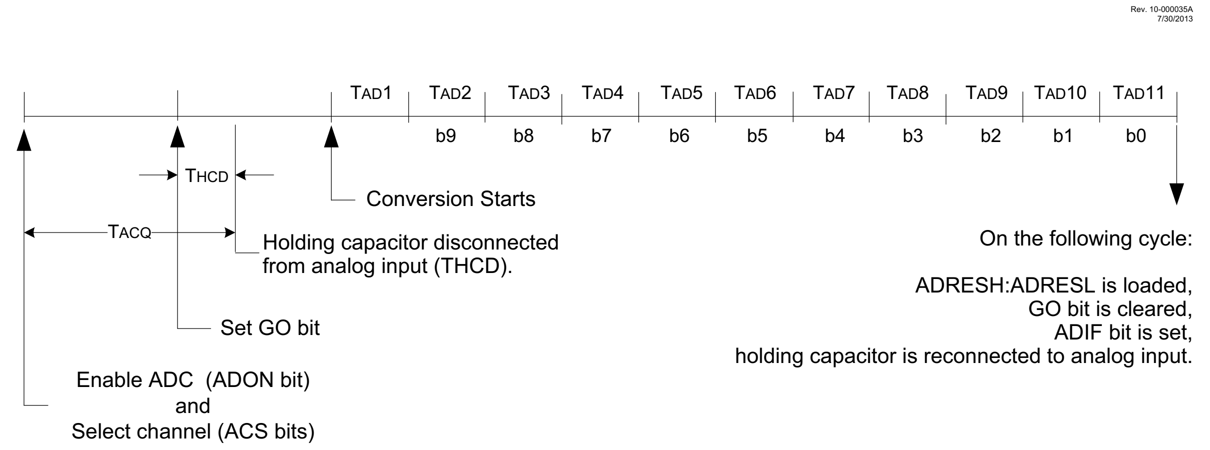
Legend: Shaded cells are outside of recommended range.

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

2: When the device frequency is greater than 1 MHz, the FRC clock source is only recommended if the conversion will be performed during Sleep.

3: The TAD period when using the FRC clock source can fall within a specified range, (see TAD parameter). The TAD period when using the FOSC-based clock source can be configured for a more precise TAD period. However, the FRC clock source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 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.

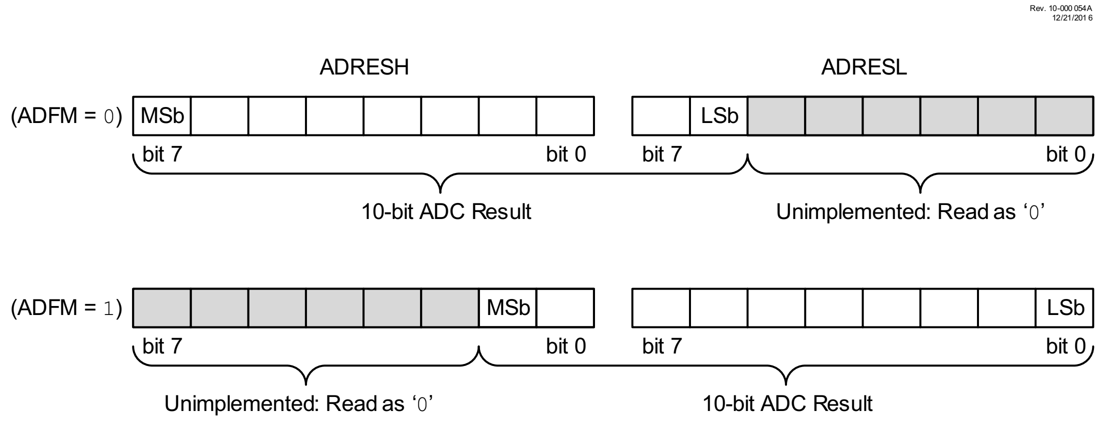
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume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

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/$\overline{\text{DONE}}$ bit of the ADCON0 register to a ‘1’ will start the Analog-to-Digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section16.2.6 “ADC Conversion Procedure”**.

16.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ 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/$\overline{\text{DONE}}$ 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/$\overline{\text{DONE}}$ bit is set by hardware.

The auto-conversion trigger source is selected with the TRIGSEL<4:0> bits of the ADCON2 register.

Using the auto-conversion trigger does not assure proper ADC timing. It is the user’s responsibility to ensure that the ADC timing requirements are met.

See Table 16-2 for auto-conversion sources.

TABLE 16-2: AUTO-CONVERSION SOURCES

Source Peripheral	Signal Name
Timer0	T0_overflow
Timer1	T1_overflow
Timer2	TMR2_postscaled
Timer4	TMR4_postscaled
Timer6	TMR6_postscaled
Comparator C1	C1_OUT_sync
Comparator C2(1)	C2_OUT_sync
SMT1	SMT1_CPW
SMT1	SMT1_CPR
SMT1	SMT1_PR
SMT2	SMT2_CPW
SMT2	SMT2_CPR
SMT2	SMT2_PR
CCP1	CCP1_out
CCP2	CCP2_out

Note 1: PIC16(L)F1613 only.

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 **Section16.4 “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/$\overline{\text{DONE}}$	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

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

bit 1 **GO/$\overline{\text{DONE}}$:** 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 **Section15.0 “Temperature Indicator Module”**.
2: See **Section17.0 “8-bit Digital-to-Analog Converter (DAC1) Module”** for more information.
3: See **Section14.0 “Fixed Voltage Reference (FVR)”** for more information.
4: AN<7:4> available on PIC16(L)F1613 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

`111` = FRC (clock supplied from an internal RC oscillator)
`110` = FOSC/64
`101` = FOSC/16
`100` = FOSC/4
`011` = FRC (clock supplied from an internal RC oscillator)
`010` = FOSC/32
`001` = FOSC/8
`000` = FOSC/2

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

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits

`11` = VRPOS is connected to internal Fixed Voltage Reference (FVR)
`10` = VRPOS is connected to external VREF+ pin(1)
`01` = Reserved
`00` = VRPOS is connected to VDD

Note 1: When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **SectionTABLE 28-13: “Analog-to-Digital Converter (ADC) Characteristics(1,2,3)”** 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]

`1111` = SMT2_PR
`1110` = SMT1_PR
`1101` = TMR6_postscaled
`1100` = TMR4_postscaled
`1011` = SMT2_CPR
`1010` = SMT2_CPW
`1001` = SMT1_CPR
`1000` = SMT1_CPW
`0111` = C2_OUT_sync[3]
`0110` = C1_OUT_sync
`0101` = TMR2_postscaled
`0100` = T1_overflow[2]
`0011` = T0_overflow[2]
`0010` = CCP2_out
`0001` = CCP1_out
`0000` = No auto-conversion trigger selected

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.
3: PIC16(L)F1613 only. Reserved on PIC12(L)F1612.

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 (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 16-4. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), refer to Figure 16-4. **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* $10k\Omega$ $5.0V$ V_{DD}

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

The value for T_C *can be approximated with the following equations:*

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

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

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

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

Solving for T_C:

$$
\begin{aligned}
T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ \ln(1/2047) \\
&= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega)\ \ln(0.0004885) \\
&= 1.12\mu s
\end{aligned}
$$

Therefore:

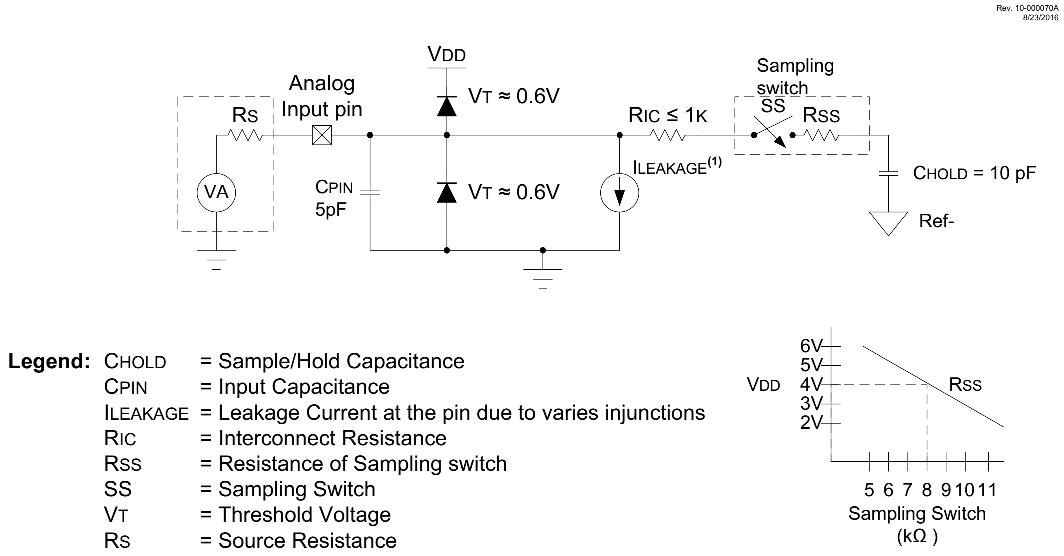
$$
\begin{aligned}
T_{ACQ} &= 2\mu s + 1.12\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\
&= 4.37\mu s
\end{aligned}
$$

Note 1: The reference voltage (V_{RPOS}) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (C_{HOLD}) 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 16-4: ANALOG INPUT MODEL



Legend:
CHOLD = Sample/Hold Capacitance
CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to varies injunctions
RIC = Interconnect Resistance
RSS = Resistance of Sampling switch
SS = Sampling Switch
VT = Threshold Voltage
RS = Source Resistance

Note 1: Refer to **Section28.0 "Electrical Specifications"**.

FIGURE 16-5: ADC TRANSFER FUNCTION

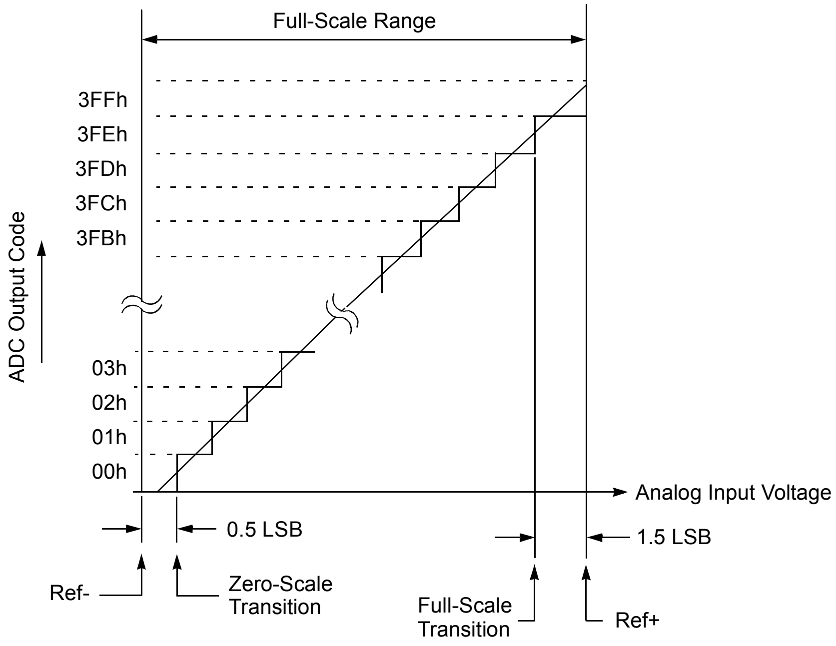


TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

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/$\overline{\text{DONE}}$	ADON	162
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		163
ADCON2	TRIGSEL<4:0>					—	—	—	164
ADRESH	ADC Result Register High								165, 166
ADRESL	ADC Result Register Low								165, 166
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	136
ANSELC(2)	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	143
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	83
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	87
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	135
TRISC(2)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	142
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		153

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

Note 1: Unimplemented, read as ‘`1`’.

2: PIC16(L)F1613 only.

17.0 8-BIT DIGITAL-TO-ANALOG CONVERTER (DAC1) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 256 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACxOUT1 pin

The Digital-to-Analog Converter (DAC) is enabled by setting the DAC1EN bit of the DAC1CON0 register.

17.1 Output Voltage Selection

The DAC has 256 voltage level ranges. The 256 levels are set with the DAC1R<7:0> bits of the DAC1CON1 register.

The DAC output voltage is determined by Equation 17-1:

EQUATION 17-1: DAC OUTPUT VOLTAGE

$$\underline{IF\ DAC1EN = 1}$$

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DAC1R[7:0]}{2^8}\right) + V_{SOURCE-}$$

$$V_{SOURCE+} = V_{DD},\ V_{REF},\ or\ FVR\ BUFFER\ 2$$

$$V_{SOURCE-} = V_{SS}$$

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.

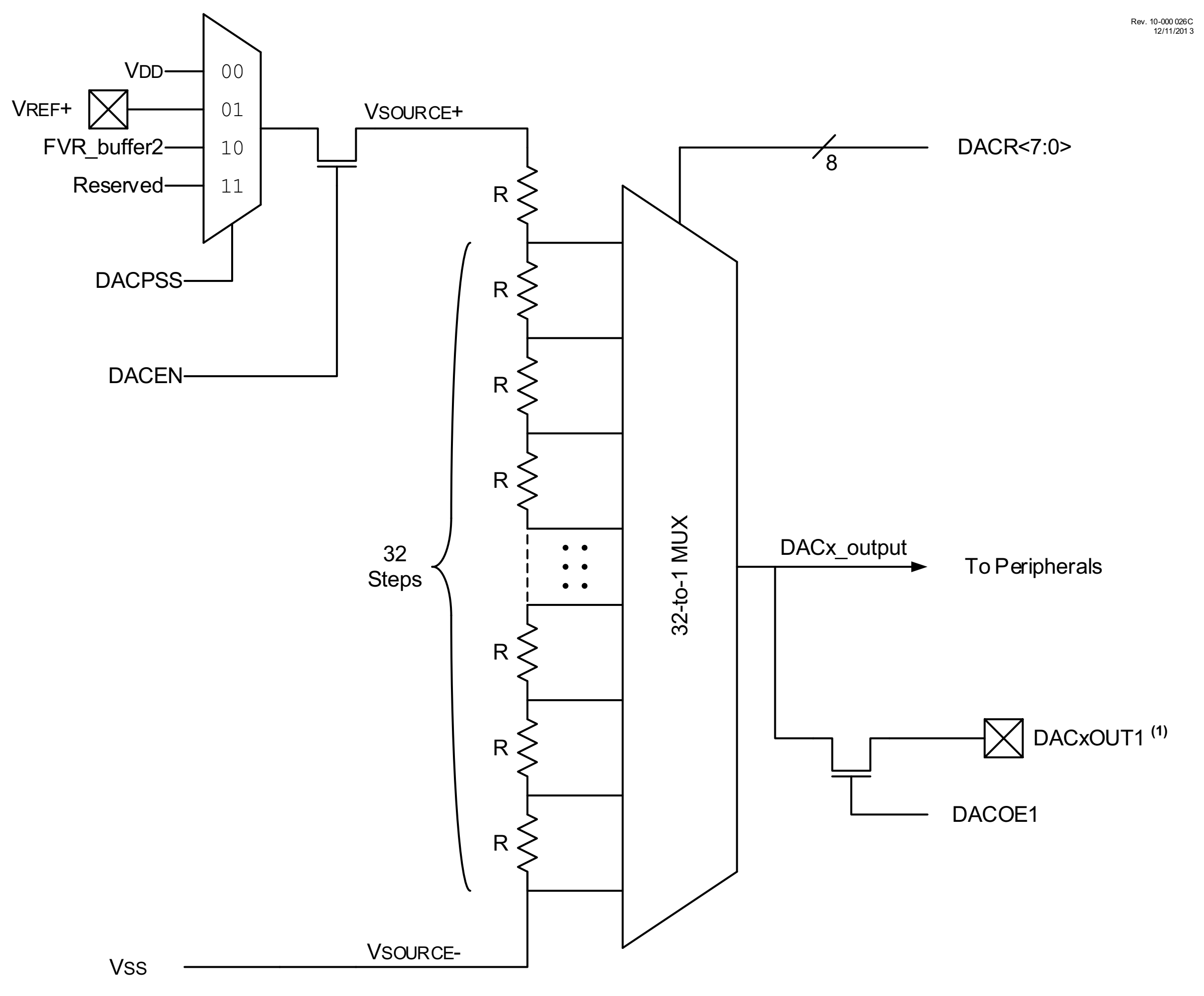
The value of the individual resistors within the ladder can be found in **Section28.0 “Electrical Specifications”**.

17.3 DAC Voltage Reference Output

The DAC voltage can be output to the DACxOUT1 pin by setting the DAC1OE1 bit of the DAC1CON0 register. Selecting the DAC reference voltage for output on the DACxOUT1 pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACxOUT1 pin when it has been configured for DAC reference voltage output will always return a ‘`0`’.

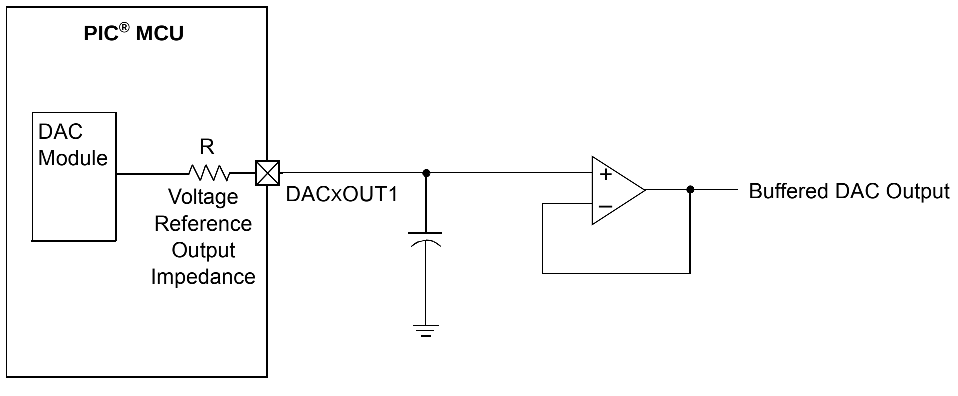
Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to either DACxOUT1 pin. Figure 17-2 shows an example buffering technique.

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



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

FIGURE 17-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



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 DAC1CON0 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:

- DAC is disabled.
- DAC output voltage is removed from the DACxOUT1 pin.
- The DAC1R<7:0> range select bits are cleared.

17.6 Register Definitions: DAC Control

REGISTER 17-1: DAC1CON0: DAC1 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
DAC1EN	—	DAC1OE1	—	DAC1PSS<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 **DAC1EN:** DAC1 Enable bit
`1` = DAC is enabled
`0` = DAC is disabled

bit 6 **Unimplemented:** Read as ‘`0`’

bit 5 **DAC1OE1:** DAC1 Voltage Output 1 Enable bit
`1` = DAC voltage level is also an output on the DACxOUT1 pin
`0` = DAC voltage level is disconnected from the DACxOUT1 pin

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

bit 3-2 **DAC1PSS<1:0>:** DAC1 Positive Source Select bits
`11` = Reserved, do not use
`10` = FVR Buffer2 output
`01` = VREF+ pin
`00` = VDD

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

REGISTER 17-2: DAC1CON1: DAC1 CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
DAC1R<7:0>							
bit 7							bit 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 **DAC1R<7:0>:** DAC1 Voltage Output Select bits

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC1 MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		153
DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	173
DAC1CON1	DAC1R<7:0>								173

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

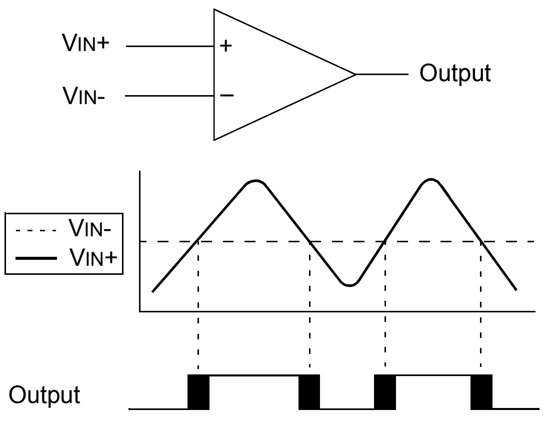
A single comparator is shown in Figure 18-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

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

TABLE 18-1: COMPARATOR AVAILABILITY PER DEVICE

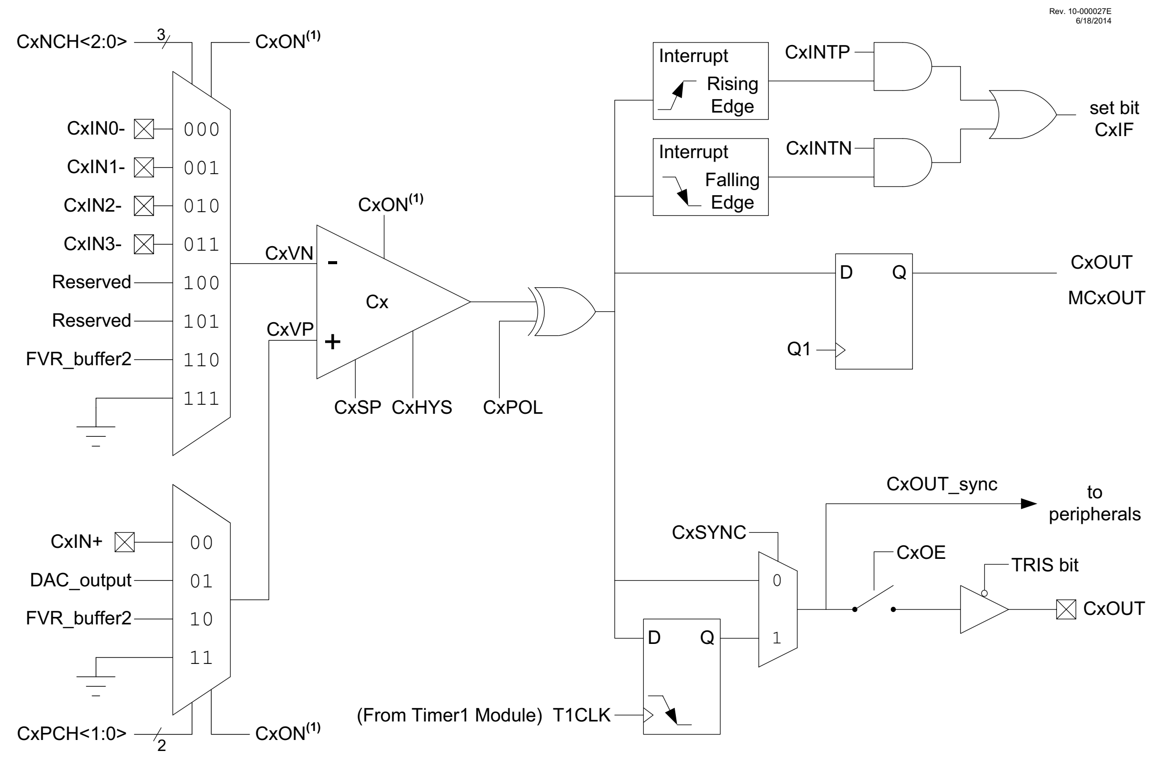
Device	C1	C2
PIC16(L)F1613	●	●
PIC12(L)F1612	●	

FIGURE 18-1: SINGLE COMPARATOR



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

FIGURE 18-2: 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.

18.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

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

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

The CMxCON1 registers (see Register 18-2) contain 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 OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

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

Note 1: The CxOE bit of the CMxCON0 register overrides the PORT data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override.

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

18.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 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`

18.2.4 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is ‘`1`’ which selects the Normal Speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to ‘`0`’.

18.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See **Section28.0 "Electrical Specifications"** for more information.

18.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section21.5 "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.

18.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from a comparator can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 18-2) and the Timer1 Block Diagram (Figure 21-1) for more information.

18.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

Note:	Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register.

18.6 Comparator Positive Input Selection

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN+ analog pin
- DAC output
- FVR (Fixed Voltage Reference)
- Vss (Ground)

See **Section14.0 "Fixed Voltage Reference (FVR)"** for more information on the Fixed Voltage Reference module.

See **Section17.0 "8-bit Digital-to-Analog Converter (DAC1) Module"** for more information on the DAC input signal.

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

18.7 Comparator Negative Input Selection

The CxNCH<2:0> bits of the CMxCON1 register direct an analog input pin or analog ground to the inverting input of the comparator:

- CxIN0- pin
- CxIN1- pin
- CxIN2- pin
- CxIN3- pin
- Analog Ground
- FVR_buffer2

Some inverting input selections share a pin with the operational amplifier output function. Enabling both functions at the same time will direct the operational amplifier output to the comparator inverting input.

Note:	To use CxINy+ and CxINy- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

18.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section28.0 "Electrical Specifications"** for more details.

18.9 Analog Input Connection Considerations

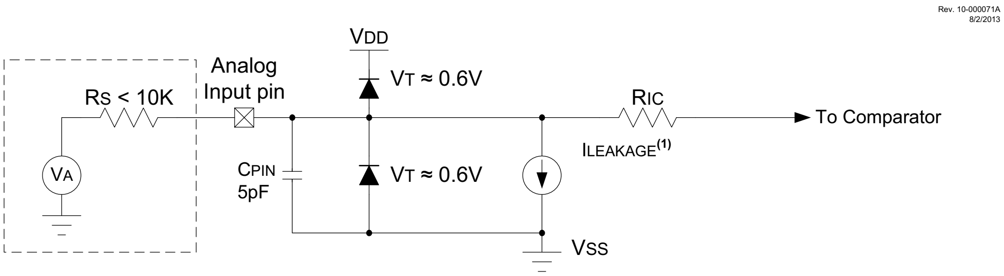
A simplified circuit for an analog input is shown in Figure 18-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range 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 **Section28.0 "Electrical Specifications"**.

18.10 Register Definitions: Comparator Control

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

R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	CxOE	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 **CxOE:** Comparator Output Enable bit

`1` = CxOUT is present on the CxOUT pin. Requires that the associated TRIS bit be cleared to drive the pin. Not affected by CxON.
`0` = CxOUT is internal only

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 operates in normal power, higher speed mode
`0` = Comparator operates in Low-power, Low-speed mode

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	R/W-0/0	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 AGND
`10` = CxVP connects to FVR Buffer 2
`01` = CxVP connects to VDAC
`00` = CxVP connects to CxIN+ pin

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

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

Note 1: PIC16(L)F1613 only.

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(1)	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(1)

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

Note 1: PIC16(L)F1613 only.

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	136
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	179
CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			180
CM2CON0(2)	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	179
CM2CON1(2)	C2INTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			180
CMOUT	—	—	—	—	—	—	MC2OUT(2)	MC1OUT	180
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		153
DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	173
DAC1CON1	DAC1R<7:0>								173
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	84
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	88
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	135
TRISC(2)	TRISC7(2)	TRISC6(2)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	142

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

Note 1: Unimplemented, read as ‘1’.

2: PIC16(L)F1613 only.

19.0 ZERO-CROSS DETECTION (ZCD) MODULE

The ZCD module detects when an A/C signal crosses through the ground potential. The actual zero crossing threshold is the zero crossing reference voltage, VCPINV, which is typically 0.75V above ground.

The connection to the signal to be detected is through a series current limiting resistor. The module applies a current source or sink to the ZCD pin to maintain a constant voltage on the pin, thereby preventing the pin voltage from forward biasing the ESD protection diodes. When the applied voltage is greater than the reference voltage, the module sinks current. When the applied voltage is less than the reference voltage, the module sources current. The current source and sink action keeps the pin voltage constant over the full range of the applied voltage. The ZCD module is shown in the simplified block diagram Figure 19-2.

The ZCD module is useful when monitoring an A/C waveform for, but not limited to, the following purposes:

- A/C period measurement
- Accurate long term time measurement
- Dimmer phase delayed drive
- Low EMI cycle switching

19.1 External Resistor Selection

The ZCD module requires a current limiting resistor in series with the external voltage source. The impedance and rating of this resistor depends on the external source peak voltage. Select a resistor value that will drop all of the peak voltage when the current through the resistor is nominally 300 μA. Refer to Equation 19-1 and Figure 19-1. Make sure that the ZCD I/O pin internal weak pull-up is disabled so it does not interfere with the current source and sink.

EQUATION 19-1: EXTERNAL RESISTOR

$$R_{SERIES} = \frac{V_{PEAK}}{3 \times 10^{-4}}$$

FIGURE 19-1: EXTERNAL VOLTAGE

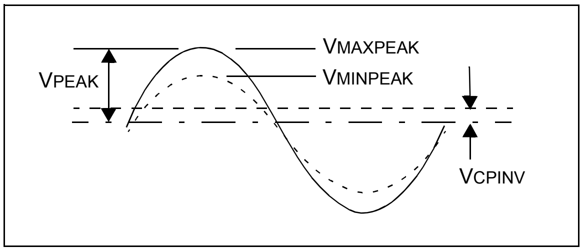
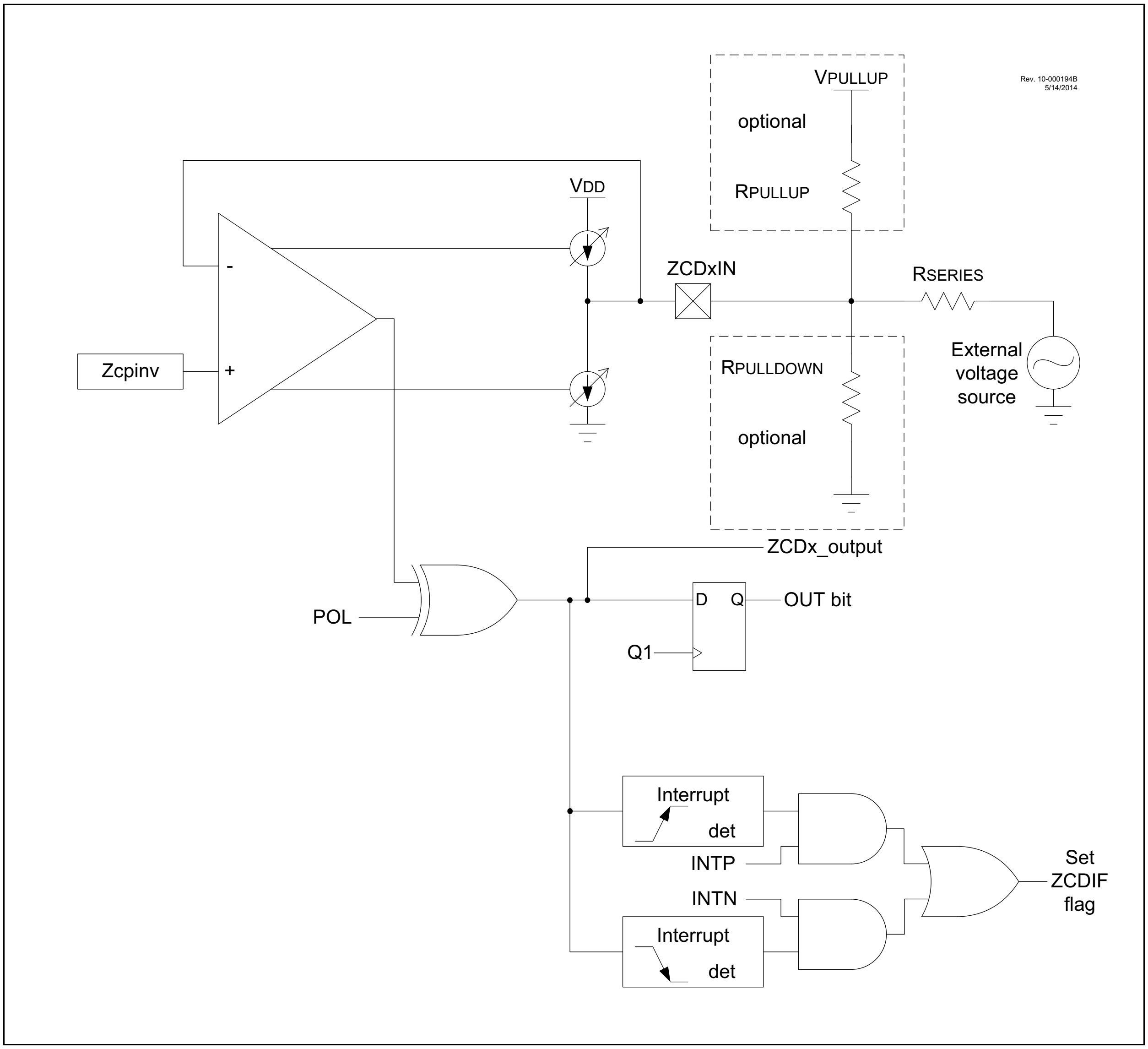


FIGURE 19-2: SIMPLIFIED ZCD BLOCK DIAGRAM



19.2 ZCD Logic Output

The ZCD module includes a Status bit, which can be read to determine whether the current source or sink is active. The ZCDxOUT bit of the ZCDxCON register is set when the current sink is active, and cleared when the current source is active. The ZCDxOUT bit is affected by the polarity bit.

19.3 ZCD Logic Polarity

The ZCDxPOL bit of the ZCDxCON register inverts the ZCDxOUT bit relative to the current source and sink output. When the ZCDxPOL bit is set, a ZCDxOUT high indicates that the current source is active, and a low output indicates that the current sink is active.

The ZCDxPOL bit affects the ZCD interrupts. See **Section19.4 "ZCD Interrupts"**.

19.4 ZCD Interrupts

An interrupt will be generated upon a change in the ZCD logic output when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in the ZCD for this purpose.

The ZCDIF bit of the PIR3 register will be set when either edge detector is triggered and its associated enable bit is set. The ZCDxINTP enables rising edge interrupts and the ZCDxINTN bit enables falling edge interrupts. Both are located in the ZCDxCON register.

To fully enable the interrupt, the following bits must be set:

- ZCDIE bit of the PIE3 register
- ZCDxINTP bit of the ZCDxCON register (for a rising edge detection)
- ZCDxINTN bit of the ZCDxCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

Changing the ZCDxPOL bit will cause an interrupt, regardless of the level of the ZCDxEN bit.

The ZCDIF bit of the PIR3 register must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

19.5 Correcting for VCPINV offset

The actual voltage at which the ZCD switches is the reference voltage at the non-inverting input of the ZCD op amp. For external voltage source waveforms other than square waves, this voltage offset from zero causes the zero-cross event to occur either too early or too late. When the waveform is varying relative to VSS, then the zero cross is detected too early as the waveform falls and too late as the waveform rises. When the waveform is varying relative to VDD, then the zero cross is detected too late as the waveform rises and too early as the waveform falls. The actual offset time can be determined for sinusoidal waveforms with the corresponding equations shown in Equation 19-2.

EQUATION 19-2: ZCD EVENT OFFSET

When External Voltage Source is relative to Vss:

$$T_{OFFSET} = \frac{\operatorname{asin}\left(\frac{V_{cpinv}}{V_{PEAK}}\right)}{2\pi \bullet Freq}$$

When External Voltage Source is relative to VDD:

$$T_{OFFSET} = \frac{\operatorname{asin}\left(\frac{V_{DD} - V_{cpinv}}{V_{PEAK}}\right)}{2\pi \bullet Freq}$$

This offset time can be compensated for by adding a pull-up or pull-down biasing resistor to the ZCD pin. A pull-up resistor is used when the external voltage source is varying relative to VSS. A pull-down resistor is used when the voltage is varying relative to VDD. The resistor adds a bias to the ZCD pin so that the target external voltage source must go to zero to pull the pin voltage to the VCPINV switching voltage. The pull-up or pull-down value can be determined with the equations shown in Equation 19-3 or Equation 19-4.

EQUATION 19-3: ZCD PULL-UP/DOWN

When External Signal is relative to Vss:

$$R_{PULLUP} = \frac{R_{SERIES}(V_{PULLUP} - V_{cpinv})}{V_{cpinv}}$$

When External Signal is relative to VDD:

$$R_{PULLDOWN} = \frac{R_{SERIES}(V_{cpinv})}{(V_{DD} - V_{cpinv})}$$

The pull-up and pull-down resistor values are significantly affected by small variations of VCPINV. Measuring VCPINV can be difficult, especially when the waveform is relative to VDD. However, by combining Equations 19-2 and 19-3, the resistor value can be determined from the time difference between the ZCDx_output high and low periods. Note that the time difference, ΔT, is 4*TOFFSET. The equation for determining the pull-up and pull-down resistor values from the high and low ZCDx_output periods is shown in Equation 19-4. The ZCDx_output signal can be directly observed on the ZCDxOUT pin by setting the ZCDxOE bit.

EQUATION 19-4:

$$R = R_{SERIES}\left(\frac{V_{BIAS}}{V_{PEAK}\left(\sin\left(\pi Freq\frac{(\Delta T)}{2}\right)\right)} - 1\right)$$

R is pull-up or pull-down resistor.

VBIAS is VPULLUP when R is pull-up or VDD when R is pull-down.

ΔT is the ZCDxOUT high and low period difference.

19.6 Handling VPEAK variations

If the peak amplitude of the external voltage is expected to vary, the series resistor must be selected to keep the ZCD current source and sink below the design maximum range of ± 600 μA and above a reasonable minimum range. A general rule of thumb is that the maximum peak voltage can be no more than six times the minimum peak voltage. To ensure that the maximum current does not exceed ± 600 μA and the minimum is at least ± 100 μA, compute the series resistance as shown in Equation 19-5. The compensating pull-up for this series resistance can be determined with Equation 19-3 because the pull-up value is independent from the peak voltage.

EQUATION 19-5: SERIES R FOR V RANGE

$$R_{SERIES} = \frac{V_{MAXPEAK} + V_{MINPEAK}}{7 \times 10^{-4}}$$

19.7 Operation During Sleep

The ZCD current sources and interrupts are unaffected by Sleep.

19.8 Effects of a Reset

The ZCD circuit can be configured to default to the active or inactive state on Power-On-Reset (POR). When the $\overline{\text{ZCD}}$ Configuration bit is cleared, the ZCD circuit will be active at POR. When the $\overline{\text{ZCD}}$ Configuration bit is set, the ZCDxEN bit of the ZCDxCON register must be set to enable the ZCD module.

19.9 Register Definitions: ZCD Control

REGISTER 19-1: ZCDxCON: ZERO-CROSS DETECTION CONTROL REGISTER

R/W-q/q	R/W-0/0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ZCDxEN	ZCDxOE	ZCDxOUT	ZCDxPOL	—	—	ZCDxINTP	ZCDxINTN
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 configuration bits

bit 7 **ZCDxEN**: Zero-Cross Detection Enable bit
`1` = Zero-cross detect is enabled. ZCD pin is forced to output to source and sink current.
`0` = Zero-cross detect is disabled. ZCD pin operates according to TRIS controls.

bit 6 **ZCDxOE:** Zero-Cross Detection Output Enable bit
`1` = ZCD pin output is enabled
`0` = ZCD pin output is disabled

bit 5 **ZCDxOUT:** Zero-Cross Detection Logic Level bit
ZCDxPOL bit = `0`:
`1` = ZCD pin is sinking current
`0` = ZCD pin is sourcing current
ZCDxPOL bit = `1`:
`1` = ZCD pin is sourcing current
`0` = ZCD pin is sinking current

bit 4 **ZCDxPOL:** Zero-Cross Detection Logic Output Polarity bit
`1` = ZCD logic output is inverted
`0` = ZCD logic output is not inverted

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

bit 1 **ZCDxINTP:** Zero-Cross Positive Edge Interrupt Enable bit
`1` = ZCDIF bit is set on low-to-high ZCDx_output transition
`0` = ZCDIF bit is unaffected by low-to-high ZCDx_output transition

bit 0 **ZCDxINTN:** Zero-Cross Negative Edge Interrupt Enable bit
`1` = ZCDIF bit is set on high-to-low ZCDx_output transition
`0` = ZCDIF bit is unaffected by high-to-low ZCDx_output transition

TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	85
PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	89
ZCD1CON	ZCD1EN	ZCD1OE	ZCD1OUT	ZCD1POL	—	—	ZCD1INTP	ZCD1INTN	186

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

TABLE 19-2: SUMMARY OF CONFIGURATION WORD WITH THE ZCD MODULE

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOR}}$	BORV	STVREN	PLLEN	53
	7:0	$\overline{\text{ZCD}}$	—	—	—	—	—	WRT<1:0>		

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

20.0 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 20-1 is a block diagram of the Timer0 module.

20.1 Timer0 Operation

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

20.1.1 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.

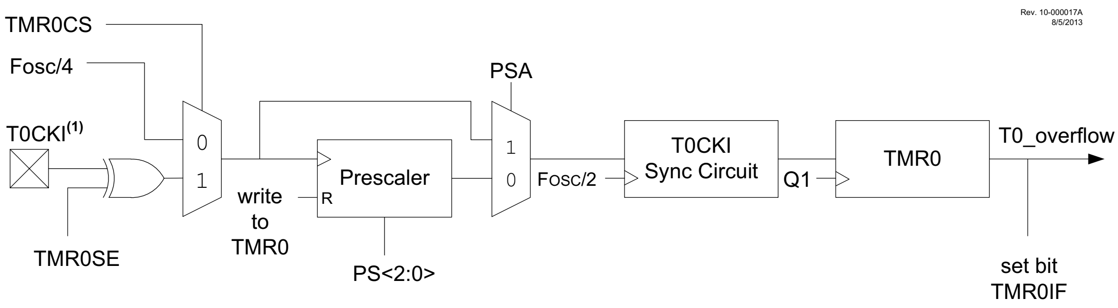
20.1.2 8-BIT COUNTER MODE

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 20-1: TIMER0 BLOCK DIAGRAM



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

20.1.3 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.

20.1.4 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.

20.1.5 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 **Section28.0 “Electrical Specifications”**.

20.1.6 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.

20.2 Register Definitions: Option Register

REGISTER 20-1: 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 20-1: 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>				—	—	—	—	164
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			190
TMR0	Holding Register for the 8-bit Timer0 Count								188*
TRISA	—	—	TRISA5	TRISA4	—[(1)]	TRISA2	TRISA1	TRISA0	135

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`’.

21.0 TIMER1/3/5 MODULE WITH GATE CONTROL

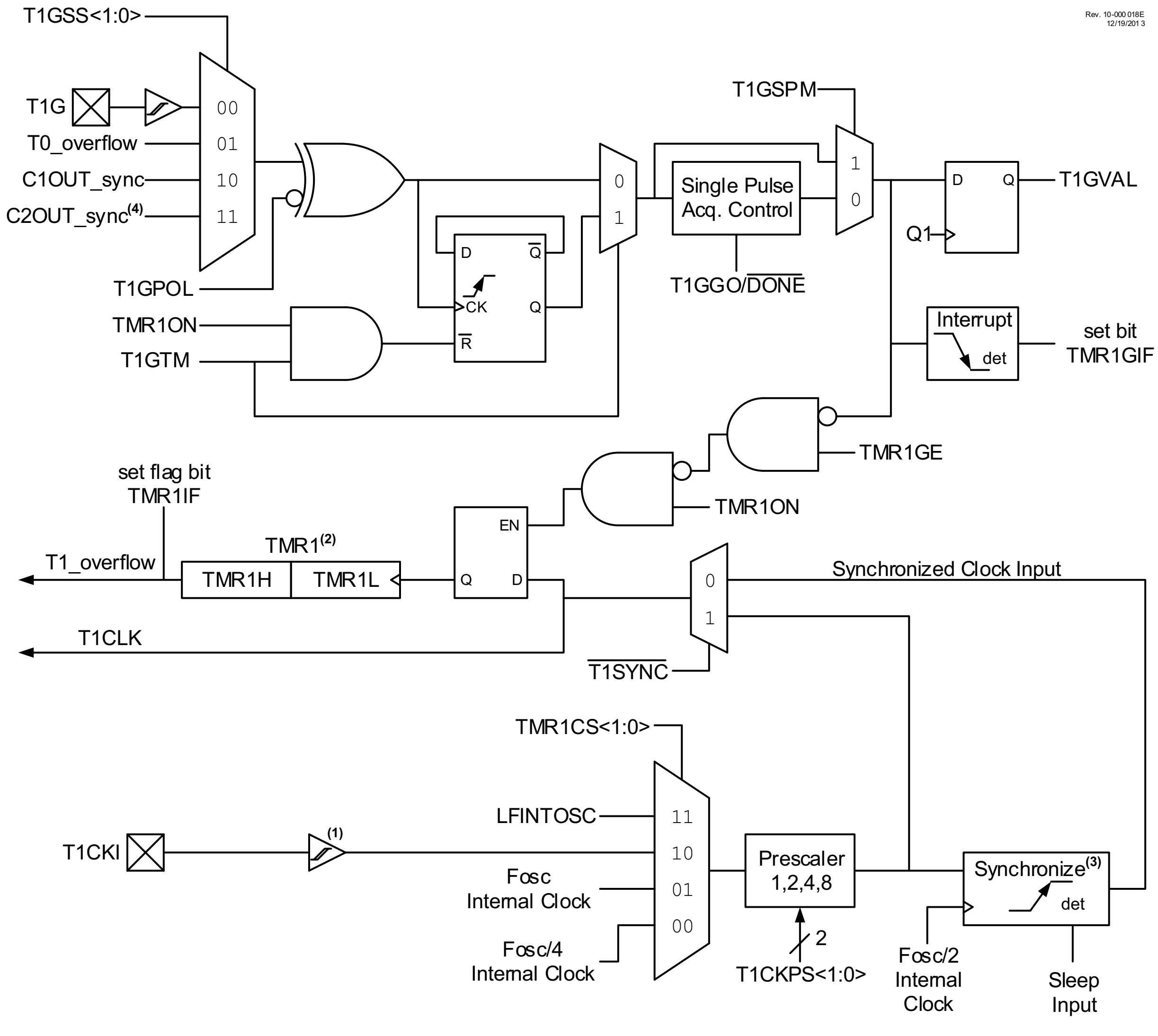
The Timer1/3/5 modules are a 16-bit timers/counters with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Optionally synchronized comparator out
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- ADC Auto-Conversion Trigger(s)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-Pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 21-1 is a block diagram of the Timer1 module.

Note: Three identical Timer1 modules are implemented on this device. The timers are named Timer1, Timer3, and Timer5. All references to Timer1 apply as well to Timer3 and Timer5, as well as references to their associated registers.

FIGURE 21-1: 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.

4: PIC16(L)F1613 only

21.1 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 21-1 displays the Timer1 enable selections.

TABLE 21-1: TIMER1 ENABLE SELECTIONS

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

21.2 Clock Source Selection

The TMR1CS<1:0> bits of the T1CON register are used to select the clock source for Timer1. Table 21-2 displays the clock source selections.

21.2.1 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 (PIC16(L)F1613 only) comparator input to Timer1 gate

21.2.2 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 21-2: CLOCK SOURCE SELECTIONS

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

21.3 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.

21.4 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 **Section21.4.1 "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.

21.4.1 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.

TABLE 21-4: 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 (C1_OUT_sync)(1)
11	Comparator 2 Output (C2_OUT_sync)(1,2)

Note 1: Optionally synchronized comparator output.
2: PIC16(L)F1613 only.

21.5 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.

21.5.1 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 21-3 for timing details.

TABLE 21-3: TIMER1 GATE ENABLE SELECTIONS

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

21.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 21-4. 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.

21.5.2.1 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.

21.5.2.2 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.

21.5.3 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 21-4 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.

21.5.4 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 21-5 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 21-6 for timing details.

21.5.5 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).

21.5.6 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).

21.6 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.

21.7 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

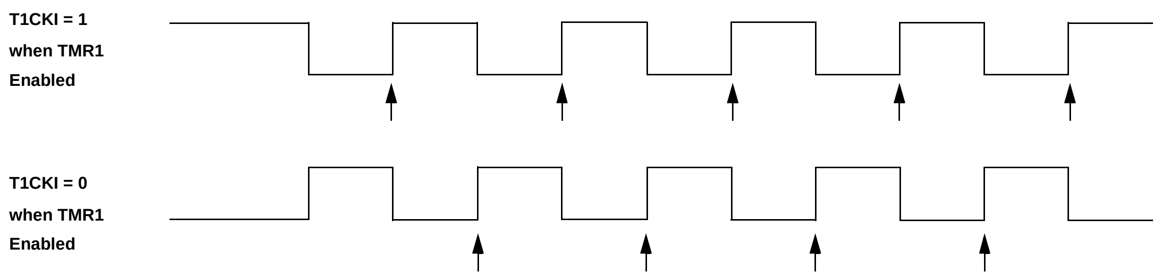
The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

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

21.7.1 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function register, APFCON. To determine which pins can be moved and what their default locations are upon a Reset, see **Section12.1 “Alternate Pin Function”** for more information.

FIGURE 21-2: 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 21-3: TIMER1 GATE ENABLE MODE

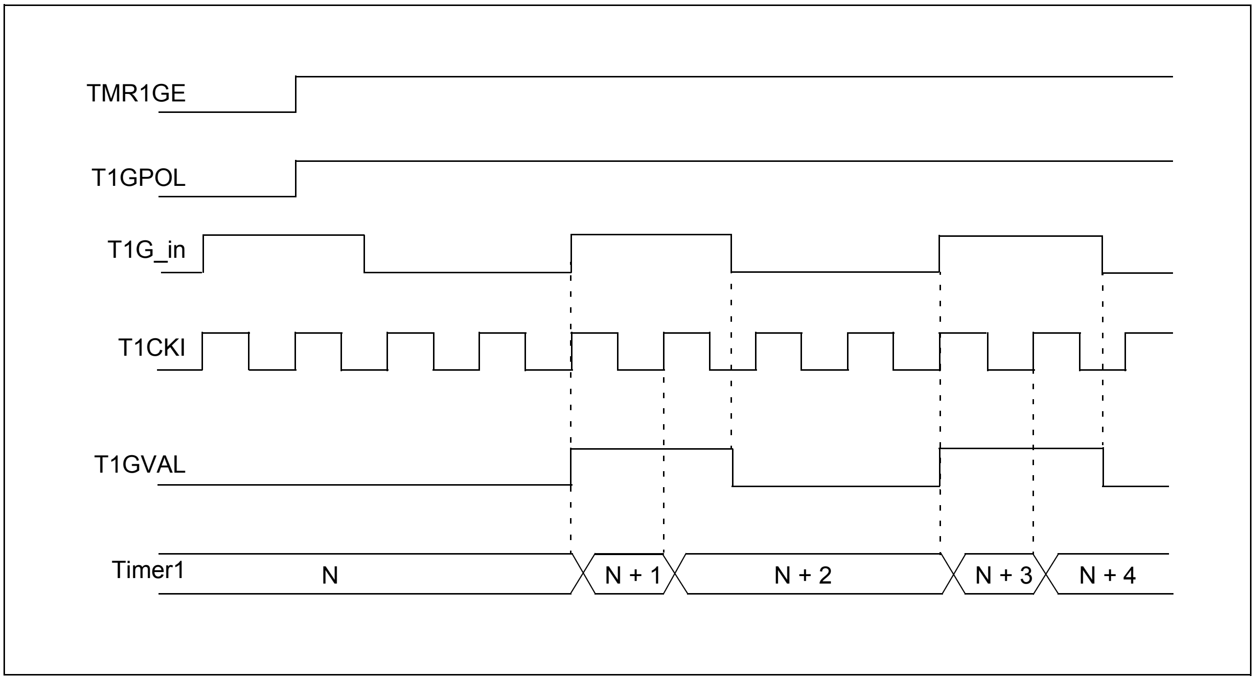


FIGURE 21-4: TIMER1 GATE TOGGLE MODE

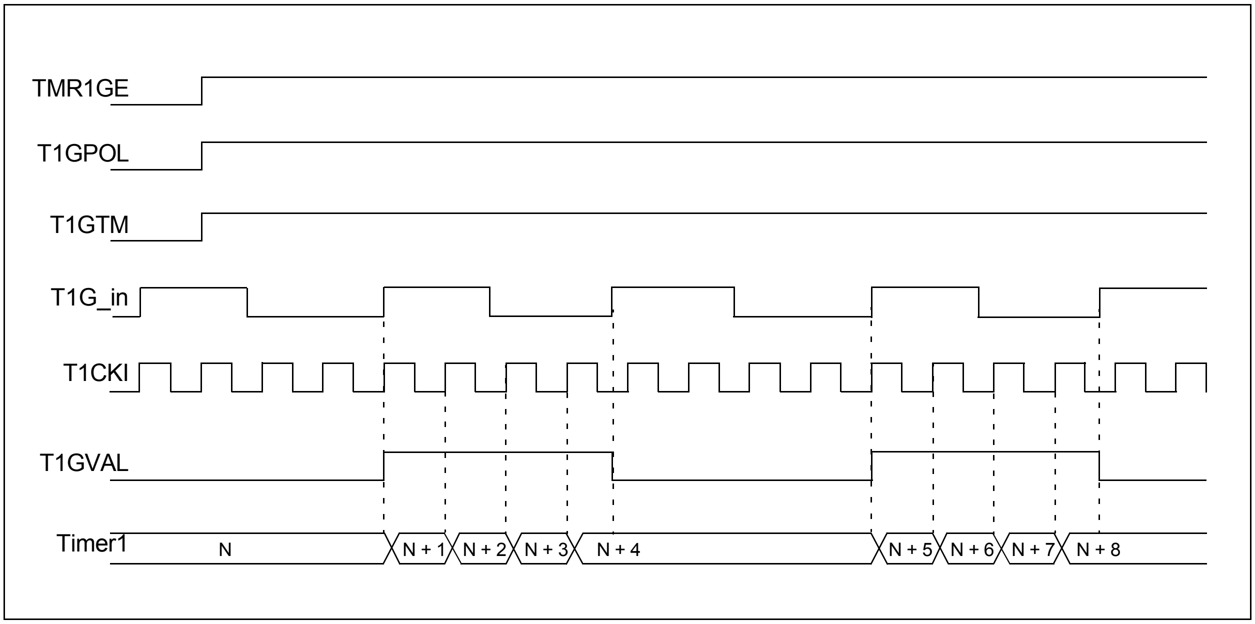


FIGURE 21-5: TIMER1 GATE SINGLE-PULSE MODE

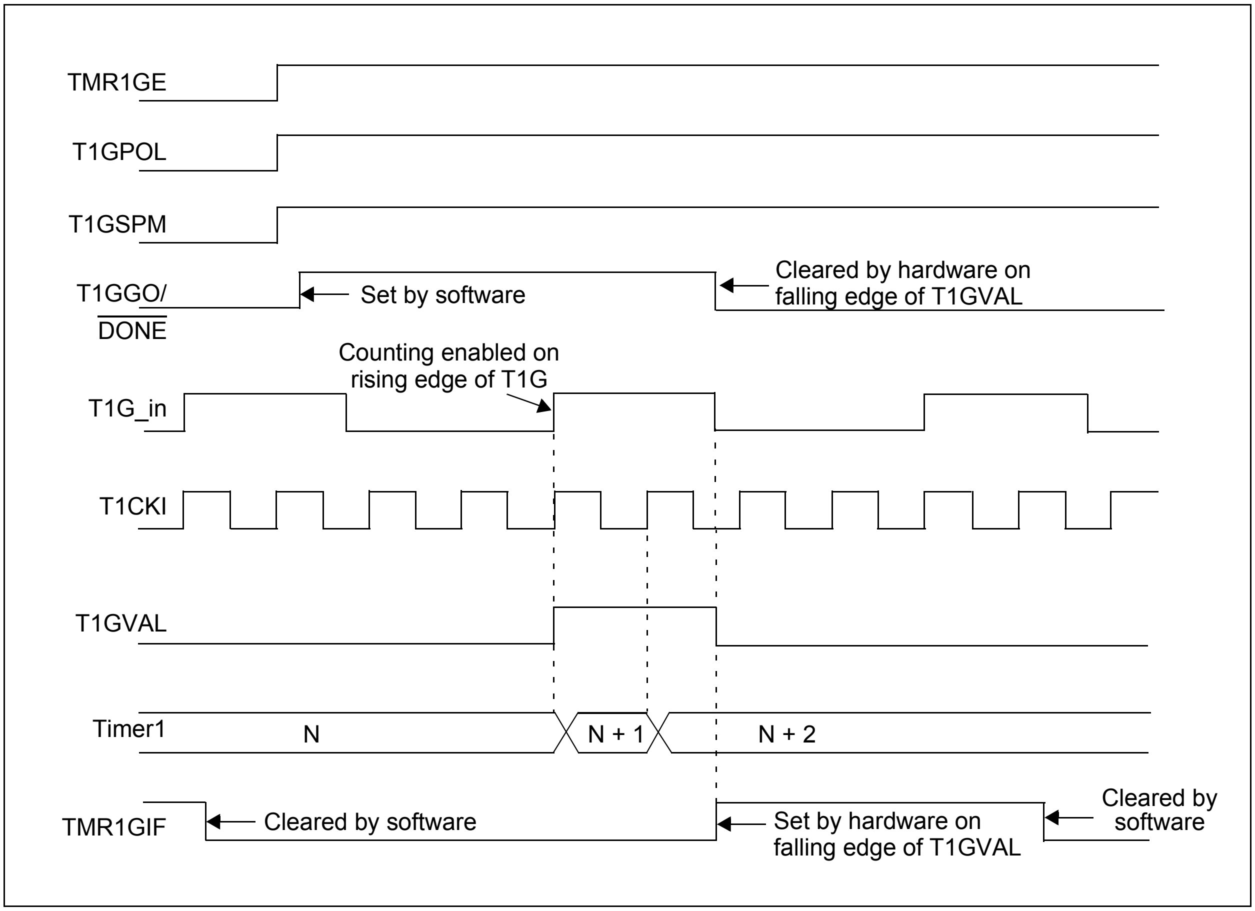
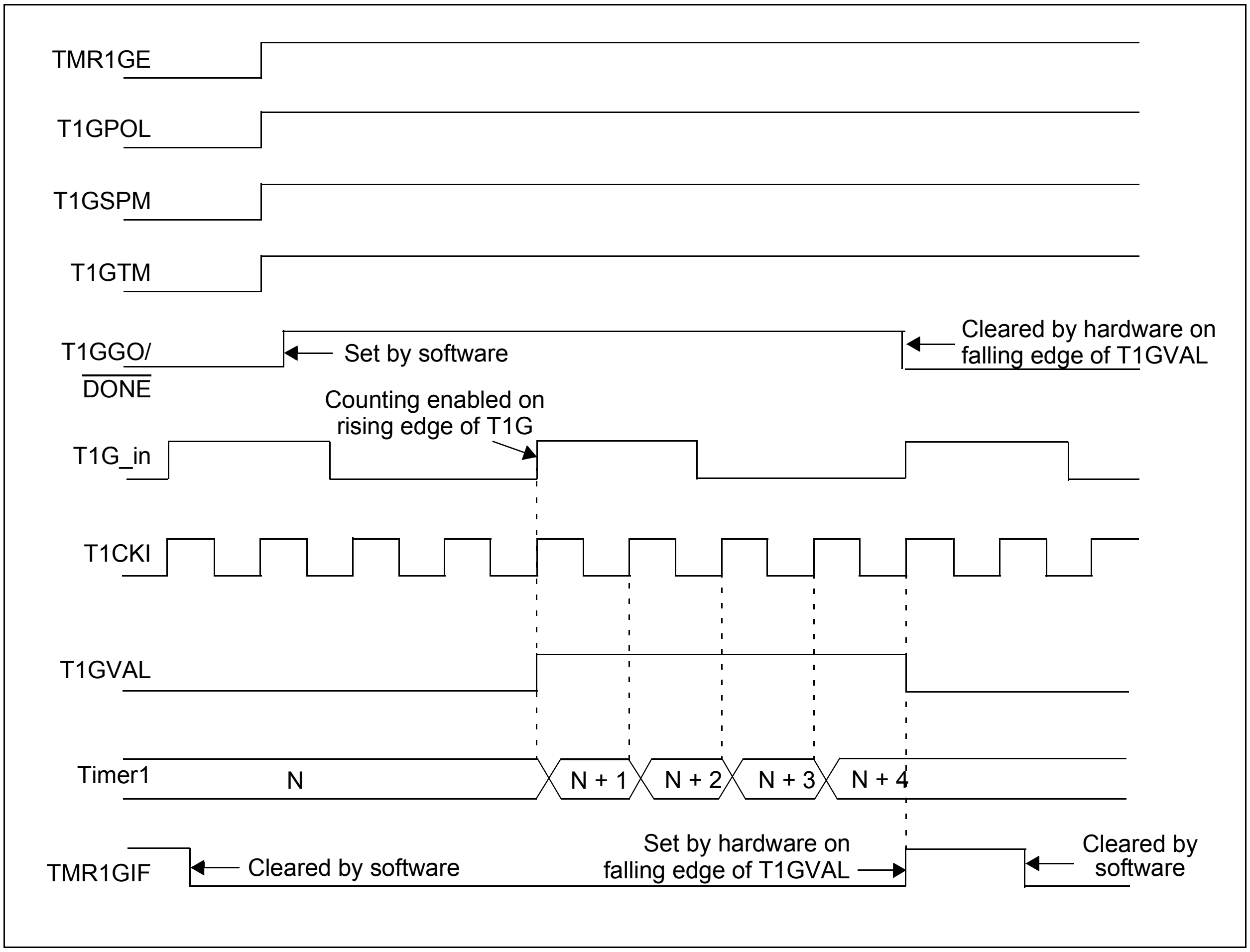


FIGURE 21-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



21.8 Register Definitions: Timer1 Control

REGISTER 21-1: 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` =LFINTOSC
`10` =T1CKI
`01` =FOSC
`00` =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 **$\overline{\textbf{T1SYNC}}$:** 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 21-2: 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/$\overline{\textbf{DONE}}$:** 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 0 **T1GSS<1:0>:** Timer1 Gate Source Select bits
`11` =Comparator 2 optionally synchronized output (C2_OUT_sync)
`10` =Comparator 1 optionally synchronized output (C1_OUT_sync)
`01` =Timer0 overflow output (T0_overflow)
`00` =Timer1 gate pin (T1G)

TABLE 21-5: 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	136
APFCON	—	CWGASEL(2)	CWGBSEL(2)	—	T1GSEL	—	CCP2SEL(3)	CCP1SEL(2)	132
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	83
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	87
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								196*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count								196*
TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Count								196*
TMR3L	Holding Register for the Least Significant Byte of the 16-bit TMR3 Count								196*
TMR5H	Holding Register for the Most Significant Byte of the 16-bit TMR5 Count								196*
TMR5L	Holding Register for the Least Significant Byte of the 16-bit TMR5 Count								196*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	135
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON	200
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		201
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		—	T3SYNC	—	TMR3ON	200
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{DONE}}$	T3GVAL	T3GSS<1:0>		201
T5CON	TMR5CS<1:0>		T5CKPS<1:0>		—	T5SYNC	—	TMR5ON	200
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		201

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: PIC12(L)F1612 only.
3: PIC16(L)F1613 only.

22.0 TIMER2/4/6 MODULE

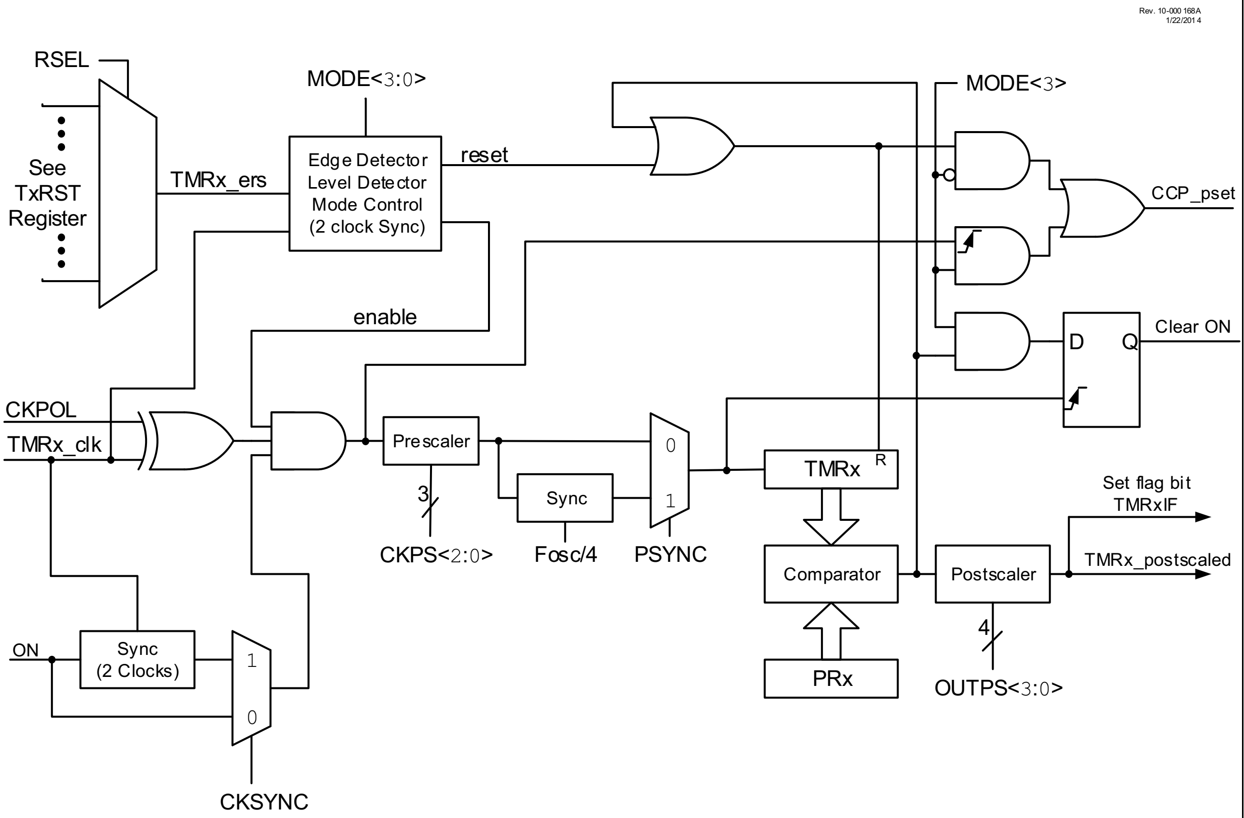
The Timer2/4/6 modules are 8-bit timers that can operate as free-running period counters or in conjunction with external signals that control start, run, freeze, and reset operations in a One-Shot mode of operation. Sophisticated waveform control such as pulse density modulation are possible by combining the operation of these timers with other internal peripherals such as the comparators and CCP modules. Features of the timer include:

- 8-bit Timer register
- 8-bit Period register
- Selectable external hardware timer Resets
- Programmable prescaler (1:1 to 1:128)
- Programmable postscaler (1:1 to 1:16)
- Selectable synchronous/asynchronous operation
- Alternate clock sources
- Interrupt-on-period
- Two modes of operation
  - Free Running Period
  - One-Shot

See Figure 22-2 for Timer2 clock sources. See Figure 22-1 for a block diagram of Timer2 with HLT.

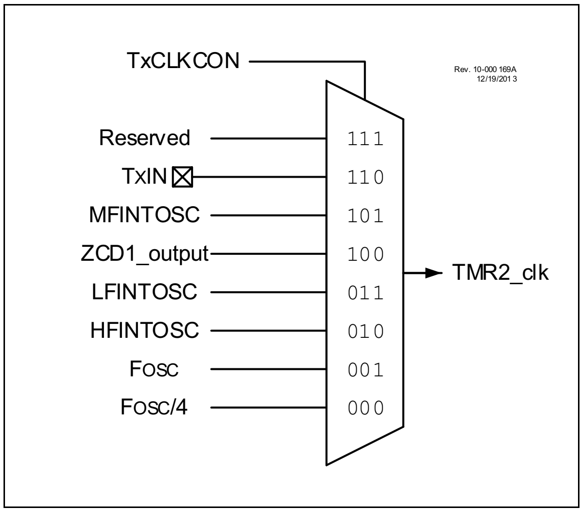
Note:	Three identical Timer2 modules are implemented on this device. The timers are named Timer2, Timer4, and Timer6. All references to Timer2 apply as well to Timer4 and Timer6. All references to PR2 apply as well to PR4 and PR6.

FIGURE 22-1: TIMER2 WITH HARDWARE LIMIT TIMER (HLT) BLOCK DIAGRAM



Note 1: Signal to the CCP to trigger the PWM pulse
2: See Section 22.5 for description of CCP interaction in the different TMR modes

FIGURE 22-2: TIMER2 CLOCK SOURCE BLOCK DIAGRAM



22.1 Timer2 Operation

Timer2 operates in two major modes:

- Free Running Period mode
- One-Shot mode

Within each mode there are several options for starting, stopping, and reset. Table 22-1 lists the options.

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
- Any device Reset
- External Reset Source events, which resets the timer.

Note: TMR2 is not cleared when T2CON is written.

22.1.1 FREE RUNNING PERIOD MODE

The value of TMR2 is compared to that of the Period register, T2PR, on each clock cycle. When the two values match, the comparator resets the value of TMR2 to 00h on the next cycle and increments the output postscaler counter. When the postscaler count equals the value in the OUTPS<3:0> bits of the TMRxCON1 register, a one clock period wide pulse occurs on the TMR2_postscaled output and the postscaler count is cleared.

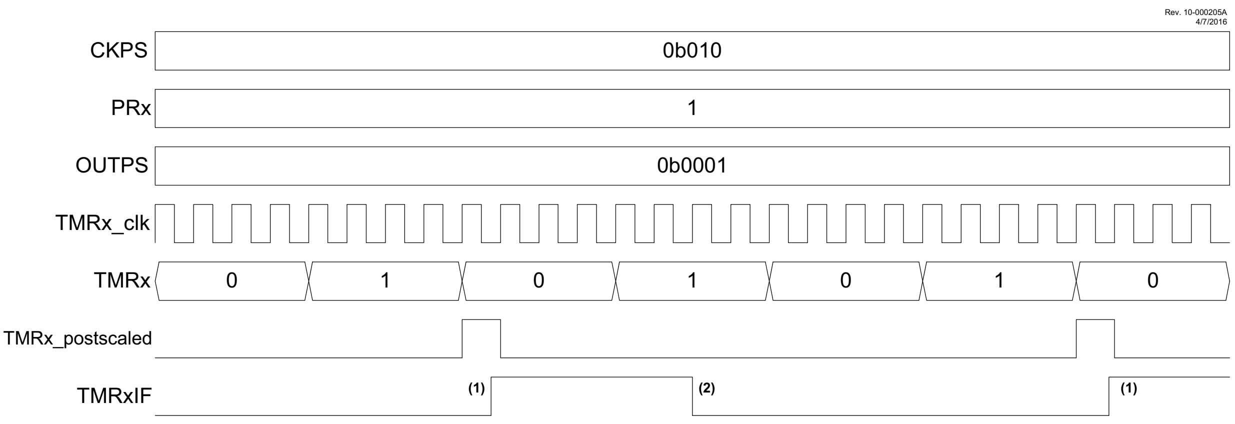
22.1.2 ONE-SHOT MODE

The One-Shot mode is identical to the Free Running Period mode except that the ON bit is cleared and the timer is stopped when TMR2 matches T2PR and will not restart until the T2ON bit is cycled off and on. Postscaler OUTPS<3:0> values other than 0 are meaningless in this mode because the timer is stopped at the first period event and the postscaler is reset when the timer is restarted.

22.2 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches one of 16 postscale options (from 1:1 through 1:16), which is selected with the postscaler control bits, OUTPS<3:0> of the T2CON register. The interrupt is enabled by setting the TMR2 Interrupt Enable bit, TMR2IE, of the PIE1 register. The interrupt timing is illustrated in Figure 22-3.

FIGURE 22-3: TIMER2 PRESCALER, POSTSCALER, AND INTERRUPT TIMING DIAGRAM



Note 1: Setting the interrupt flag is synchronized with the instruction clock. Synchronization may take as many as 2 instruction cycles

2: Cleared by software.

22.3 Timer2 Output

The Timer2 module's primary output is TMR2_postscaled, which pulses for a single TMR2_clk period upon each match of the postscaler counter and the OUTPS TMR2xCON. The PR2 postscaler is incremented each time the TMR2 value matches the PR2 value. this signal can be selected as an input to several other input modules:

- The CRC memory scanner, as a trigger for Triggered mode
- The ADC module, as an auto-conversion trigger
- Both SMT modules, as both a window and/or a signal input
- CWG, as an auto-shutdown source

In addition, the Timer2 is also used by the CCP module for pulse generation in PWM mode. Both the actual TMR2 value as well as other internal signals are sent to the CCP module to properly clock both the period and pulse width of the PWM signal. See **Section23.4 "CCP/PWM Clock Selection"** for more details on setting up Timer2 for use with the CCP, as well as the timing diagrams in **Section22.5 "Operation Examples"** for examples of how the varying Timer2 modes affect CCP PWM output.

22.4 External Reset Sources

In addition to the clock source, the Timer2 also takes in an external Reset source. This external Reset source is selected for Timer2, Timer4, and Timer6 with the T2RST, T4RST, and T6RST registers, respectively. This source can control starting and stopping of the timer, as well as resetting the timer, depending on which mode the timer is in. The mode of the timer is controlled by the MODE<3:0> bits of the TxHLT register.

Note 1: Because of Synchronization, there needs to be at least six clock pulses between each external Reset signal pulse while in edge-triggered modes. A second pulse fewer than six clock pulses after a first will not be detected by the module. Similarly, in level-triggered modes, the input signal active time must be at least three clock pulses wide to be detected.

2: While the part is in a debug freeze state, external Reset sources will continue to trigger.

22.5 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the TxCON register are cleared).
- The diagrams illustrate any clock except FOSC/4 and show clock-sync delays of at least two full cycles for both ON and TMRx_ers. When using FOSC/4, the clock-sync delay is at least one instruction period for TMRx_ers; ON applies in the next instruction period.
- ON and TMRx_ers are somewhat generalized, and clock-sync delays may produce results that are slightly different than illustrated.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in **Section23.4 "CCP/PWM Clock Selection"**. The signals are not a part of the Timer2 module.

Note: The CKSYNC bit should be set while running Timer2/4/6 in order to ensure proper operation of the timer and its interactions with other modules. Clearing the CKSYNC bit should be done only in specific cases where a very specific number of clock cycles is desired, and should only be done with extreme caution.

TABLE 22-1: TIMER2 OPERATING MODES

Mode	MODE<3:0>		Output Operation	Operation	Timer Control		
	<3>	<2:0>			Start	Reset	Stop
Free Running Period	0	000	Period Pulse	Software gate (Figure 22-4)	ON = 1	—	ON = 0
		001		Hardware gate, active-high (Figure 22-5)	ON = 1 and TMRx_ers = 1	—	ON = 0 or TMRx_ers = 0
		010		Hardware gate, active-low	ON = 1 and TMRx_ers = 0	—	ON = 0 or TMRx_ers = 1
		011	Period Pulse with Hardware Reset	Rising or falling edge Reset	ON = 1	TMRx_ers ↕	ON = 0
		100		Rising edge Reset (Figure 22-6)		TMRx_ers ↑	
		101		Falling edge Reset		TMRx_ers ↓	
		110		Low level Reset		TMRx_ers = 0	ON = 0 or TMRx_ers = 0
		111		High level Reset (Figure 22-7)		TMRx_ers = 1	ON = 0 or TMRx_ers = 1
One-Shot	1	000	One-Shot	Software start (Figure 22-8)	ON = 1	—	ON = 0 or Next clock after TMRx = PRx (**Note 2**)
		001	Edge triggered start (**Note 1**)	Rising edge start (Figure 22-9)	ON = 1 and TMRx_ers ↑	—	
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
		100	Edge triggered start and hardware Reset (**Note 1**)	Rising edge start and Rising edge Reset (Figure 22-10)	ON = 1 and TMRx_ers ↑	TMRx_ers ↑	
		101		Falling edge start and Falling edge Reset	ON = 1 and TMRx_ers ↓	TMRx_ers ↓	
		110		Rising edge start and Low level Reset (Figure 22-11)	ON = 1 and TMRx_ers ↑	TMRx_ers = 0	
		111		Falling edge start and High level Reset	ON = 1 and TMRx_ers ↓	TMRx_ers = 1	

Note 1: If ON = 0 then an edge is required to restart the timer after ON = 1.

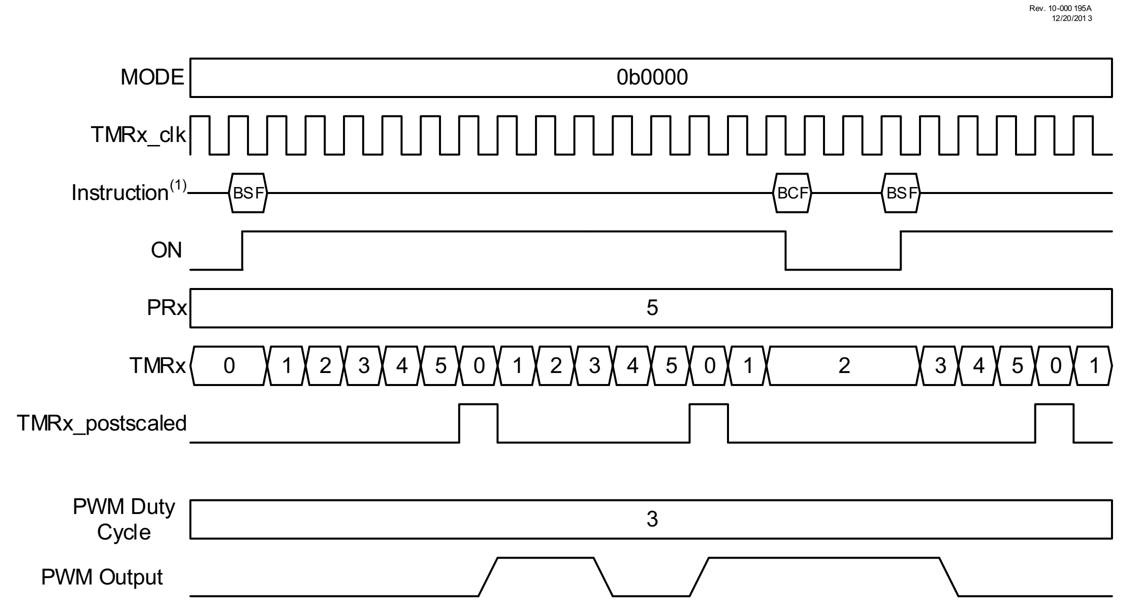
2: When TMRx = PRx then the next clock clears ON and stops TMRx at 00h.

3: When TMRx = PRx then the next clock stops TMRx at 00h but does not clear ON.

22.5.1 SOFTWARE GATE MODE

This mode corresponds to legacy Timer2 operation. The timer increments with each clock input when ON = `1` and does not increment when ON = `0`. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 22-4. With PRx = 5, the counter advances until TMRx = 5, and goes to zero with the next clock.

FIGURE 22-4: SOFTWARE GATE MODE TIMING DIAGRAM



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

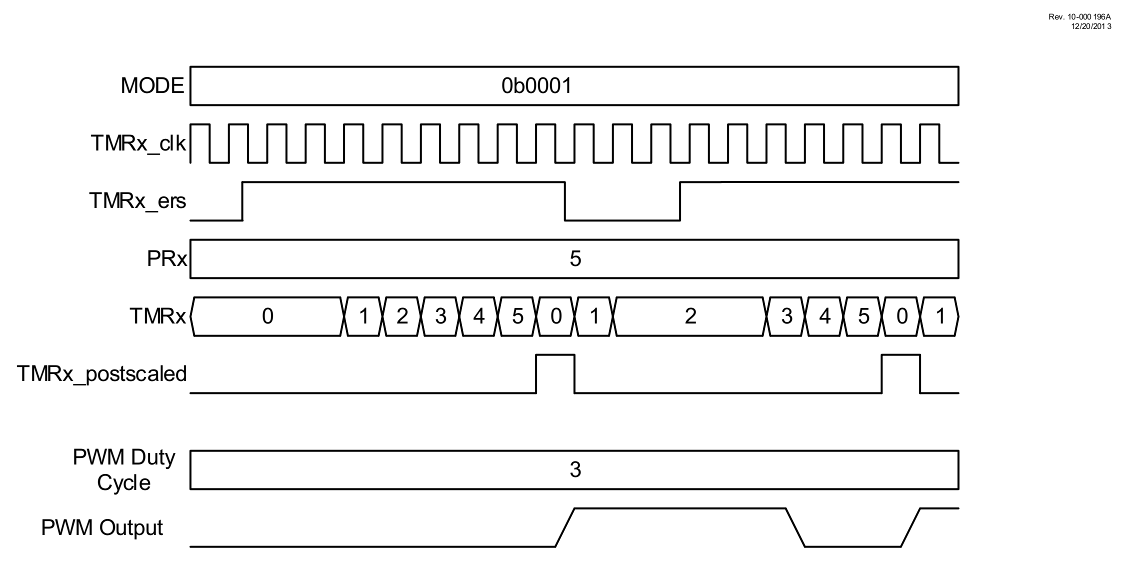
22.5.2 HARDWARE GATE MODE

The Hardware Gate modes operate the same as the software gate mode except the TMRx_ers external signal can also gate the timer. When used with the CCP the gating extends the PWM period. If the timer is stopped when the PWM output is high then the duty cycle is also extended.

When MODE<3:0> = `0001` then the timer is stopped when the external signal is high. When MODE<3:0> = `0010`, the timer is stopped when the external signal is low.

Figure 22-5 illustrates the hardware gating mode for MODE<3:0> = `0001` in which a high input level starts the counter.

FIGURE 22-5: HARDWARE GATE MODE TIMING DIAGRAM



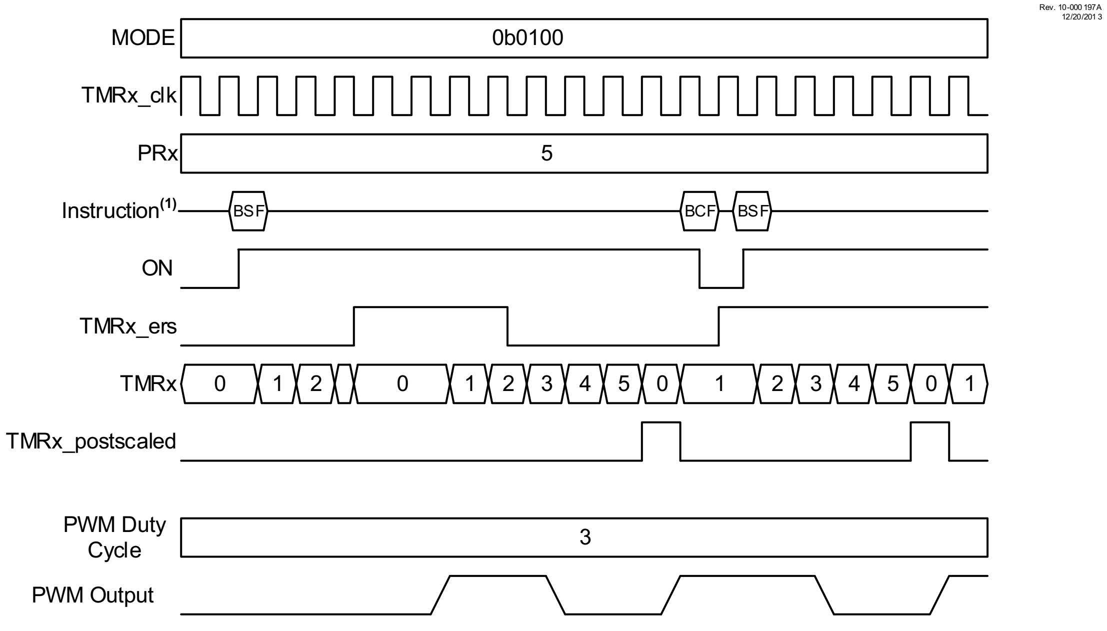
22.5.3 EDGE-TRIGGERED HARDWARE LIMIT MODE

In Edge-Triggered Hardware Limit mode, the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE<3:0> = `0011`)
- Reset on rising edge (MODE<3:0> = `0100`)
- Reset on falling edge (MODE<3:0> = `0101`)

When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two clock delay. Refer to Figure 22-6.

FIGURE 22-6: EDGE-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

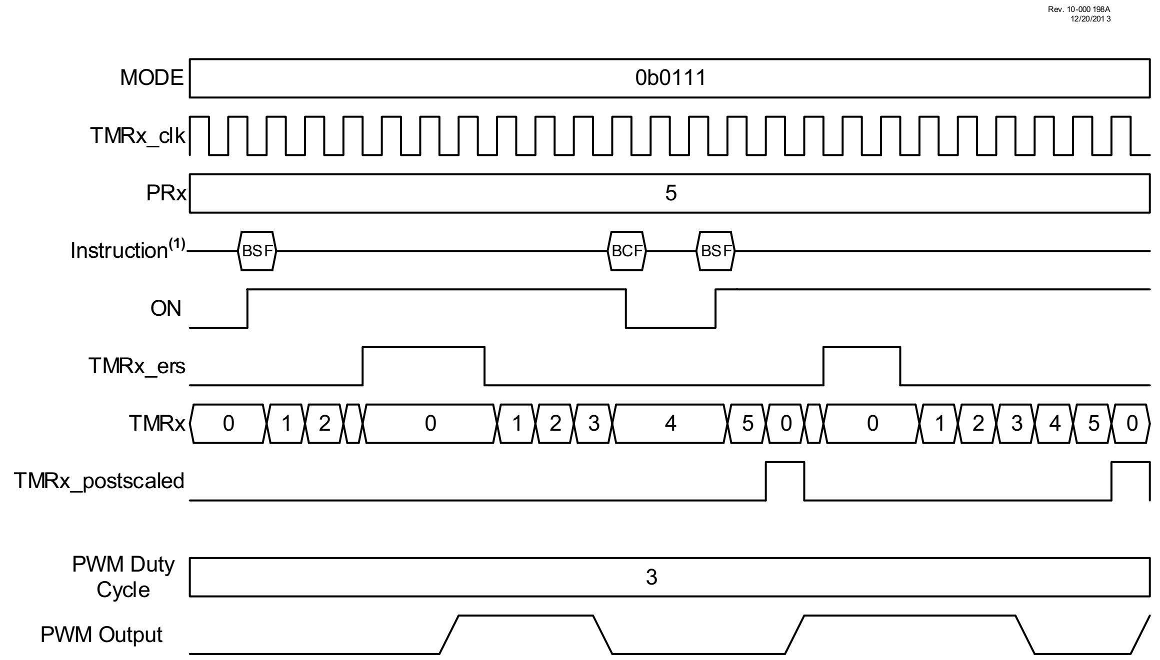
22.5.4 LEVEL-TRIGGERED HARDWARE LIMIT MODE

In the Level-Triggered Hardware Limit Timer modes, the counter is reset by high or low levels of the external signal TMRx_ers, as shown in Figure 22-7. Selecting MODE<3:0> = `0110` will cause the timer to reset on a low level external signal. Selecting MODE<3:0> = `0111` will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMRx_ers = `1`. ON is controlled by BSF and BCF instructions. When ON = `0` the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the PRx value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting and the PWM output is set high, on either the clock following the PRx match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.

FIGURE 22-7: LEVEL-TRIGGERED HARDWARE LIMIT TIMING DIAGRAM



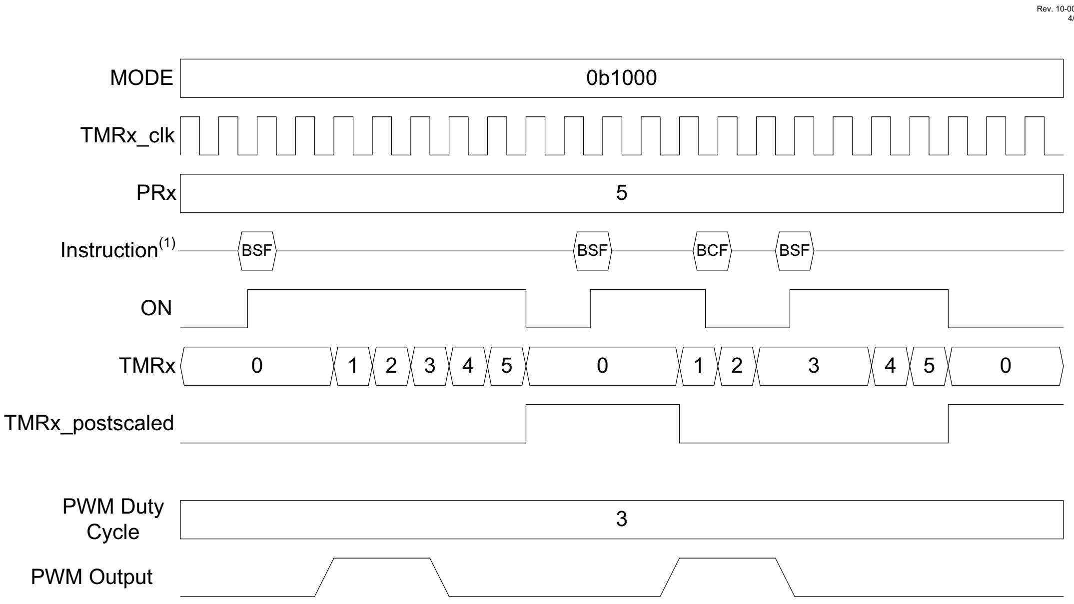
Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.5.5 SOFTWARE START ONE-SHOT MODE

In One-Shot mode, the timer resets and the ON bit is cleared when the timer value matches the PRx period value. The ON bit must be set by software to start another timer cycle. Setting MODE<3:0> = `1000` selects One-Shot mode which is illustrated in Figure 22-8. In the example, ON is controlled by BSF and BCF instructions. In the first case, a BSF instruction sets ON and the counter runs to completion and clears ON. In the second case, a BSF instruction starts the cycle, BCF/BSF instructions turn the counter off and on during the cycle, and then it runs to completion.

When One-Shot mode is used in conjunction with the CCP PWM operation the PWM pulse drive starts concurrent with setting the ON bit. Clearing the ON bit while the PWM drive is active will extend the PWM drive. The PWM drive will terminate when the timer value matches the CCPRx pulse width value. The PWM drive will remain off until software sets the ON bit to start another cycle. If software clears the ON bit after the CCPRx match but before the PRx match then the PWM drive will be extended by the length of time the ON bit remains cleared. Another timing cycle can only be initiated by setting the ON bit after it has been cleared by a PRx period count match.

FIGURE 22-8: SOFTWARE START ONE-SHOT MODE TIMING DIAGRAM



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.5.6 EDGE-TRIGGERED ONE-SHOT MODE

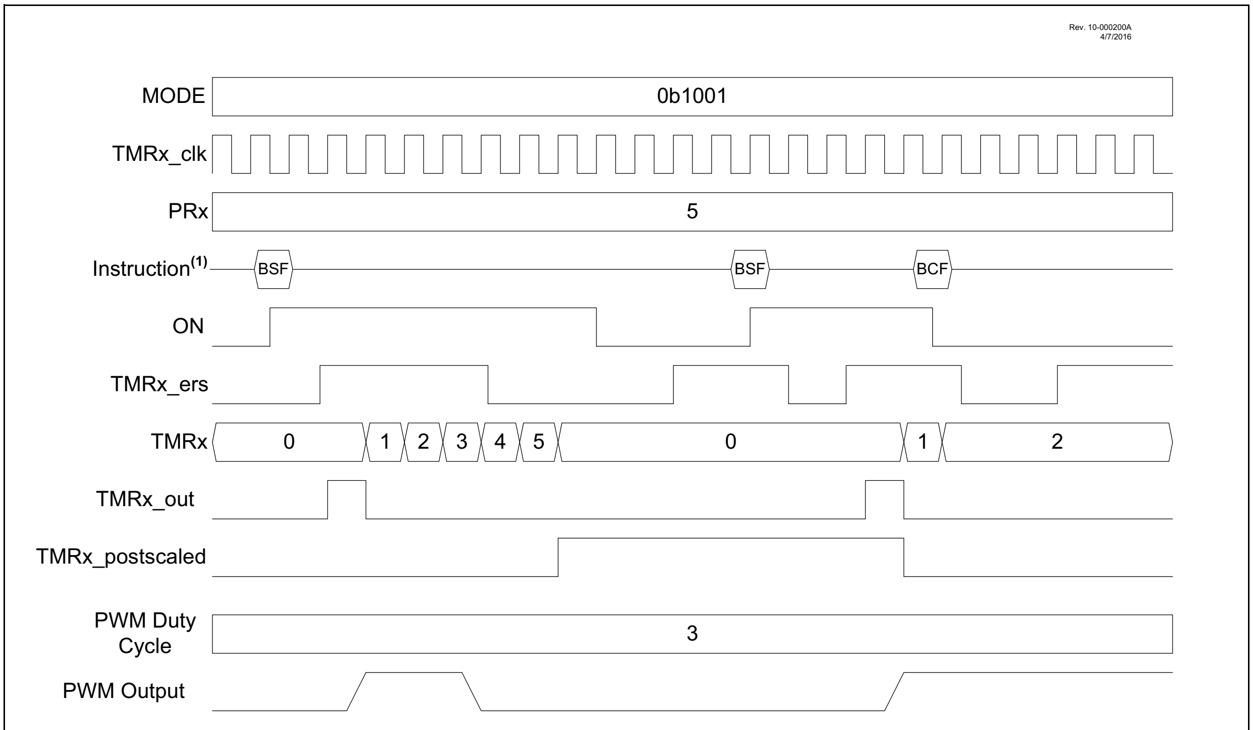
The Edge-Triggered One-Shot modes start the timer on an edge from the external signal input, after the ON bit is set, and clear the ON bit when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<3:0> = `1001`)
- Falling edge (MODE<3:0>= `1010`)
- Rising or Falling edge (MODE<3:0> = `1011`)

If the timer is halted by clearing the ON bit then another TMRx_ers edge is required after the ON bit is set to resume counting. Figure 22-9 illustrates operation in the rising edge One-Shot mode.

When the Edge-Triggered One-Shot mode is used in conjunction with the CCP then the edge-trigger will activate the PWM drive and the PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated when the timer halts at the PRx period count match.

FIGURE 22-9: EDGE-TRIGGERED ONE-SHOT MODE TIMING DIAGRAM



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.5.7 EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE

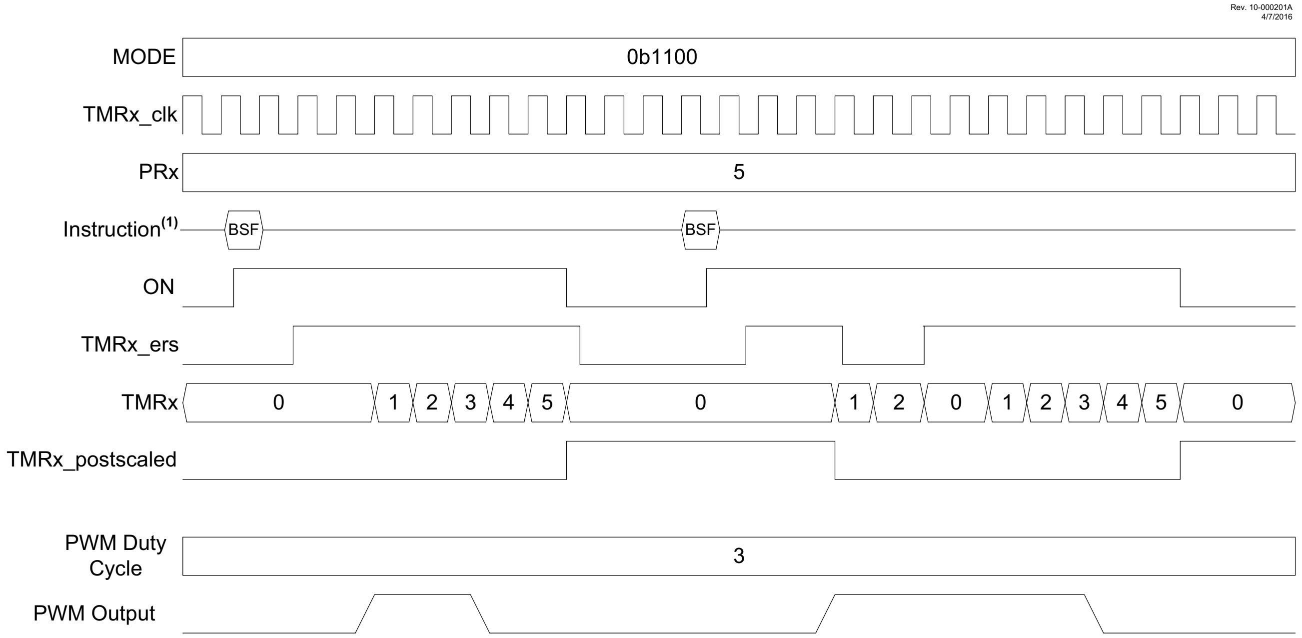
In Edge-Triggered Hardware Limit One-Shot modes the timer starts on the first external signal edge after the ON bit is set and resets on all subsequent edges. Only the first edge after the ON bit is set is needed to start the timer. The counter will resume counting automatically two clocks after all subsequent external Reset edges. Edge triggers are as follows:

- Rising edge start and reset (MODE<3:0> = `1100`)
- Falling edge start and reset (MODE<3:0> = `1101`)

The timer resets and clears the ON bit when the timer value matches the PRx period value. External signal edges will have no effect until after software sets the ON bit. Figure 22-10 illustrates the rising edge hardware limit one-shot operation.

When this mode is used in conjunction with the CCP then the first starting edge trigger, and all subsequent Reset edges, will activate the PWM drive. The PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated until the timer halts at the PRx period match unless an external signal edge resets the timer before the match occurs.

FIGURE 22-10: EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT TIMING DIAGRAM



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.5.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

In Level Reset, Edge-Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level when the ON bit is set. Reset levels are selected as follows:

- High Reset level (MODE<3:0> = `1110`)
- Low Reset level (MODE<3:0> = `1111`)

When the timer count matches the PRx period count then the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control a new external signal edge is required after the ON bit is set to start the counter.

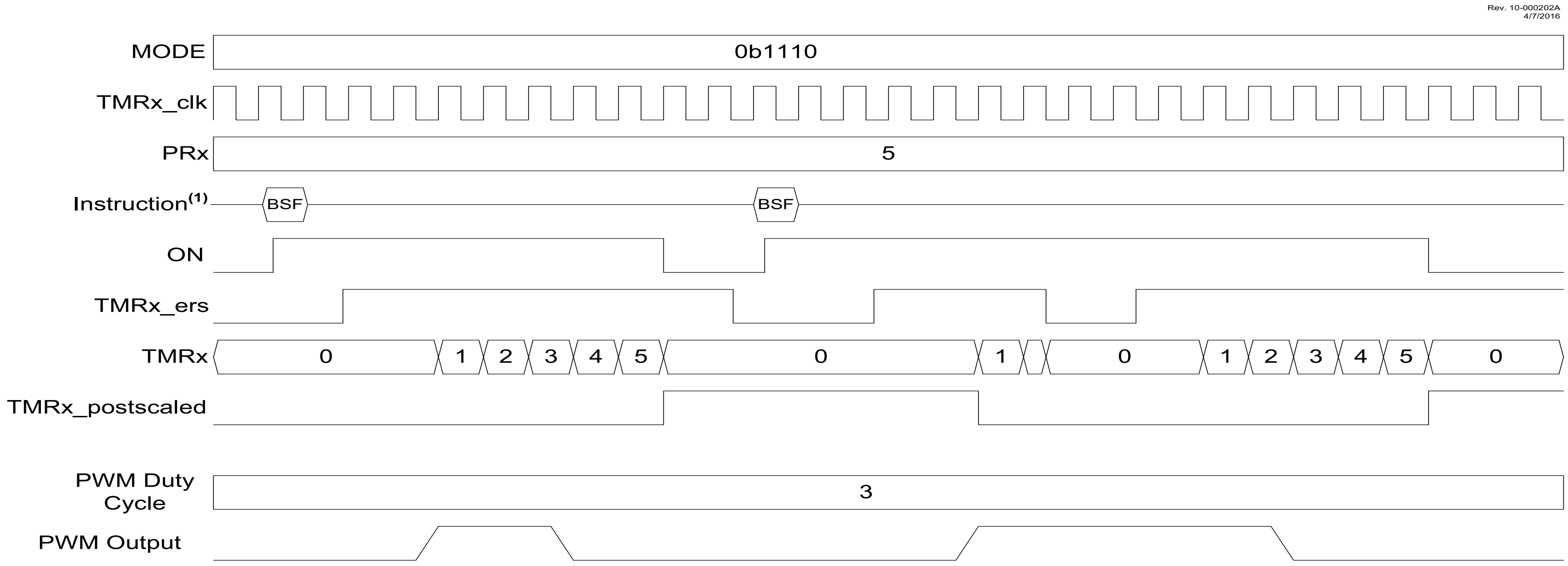
When Level Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.

22.6 Timer2 Operation During Sleep

When PSYNC = `1`, Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while processor is in Sleep mode.

When PSYNC = `0`, Timer2 will operate in Sleep as long as the clock source selected is also still running. Selecting the LFINTOSC, MFINTOSC, or HFINTOSC oscillator as the timer clock source will keep the selected oscillator running during Sleep.

FIGURE 22-11: LEVEL-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.7 Register Definitions: Timer2/4/6 Control

Long bit name prefixes for the Timer2/4/6 peripherals are shown in Table 22-2. Refer to **Section 1.1 "Register and Bit Naming Conventions"** for more information.

TABLE 22-2:

Peripheral	Bit Name Prefix
TMR2	TMR2
TMR4	TMR4
TMR6	TMR6

REGISTER 22-1: TxCLKCON: TIMERx CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	TxCS<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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **TxCS:** Timerx Clock Selection bits
111 = Reserved
110 = TxIN
101 = MFINTOSC 31.25 kHz
100 = ZCD_output
011 = LFINTOSC
010 = HFINTOSC 16 MHz
001 = FOSC
000 = FOSC/4

REGISTER 22-2: TxCON: TIMERx CONTROL REGISTER

R/W/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ON[1]	CKPS<2:0>			OUTPS<3:0>			
bit 7							bit 0

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

bit 7 **ON:** Timerx On bit

`1` = Timerx is on
`0` = Timerx is off: all counters and state machines are reset

bit 6-4 **CKPS<2:0>:** Timer2-type Clock Prescale Select bits

`111` =1:128 Prescaler
`110` =1:64 Prescaler
`101` =1:32 Prescaler
`100` =1:16 Prescaler
`011` =1:8 Prescaler
`010` =1:4 Prescaler
`001` =1:2 Prescaler
`000` =1:1 Prescaler

bit 3-0 **OUTPS<3:0>:** Timerx Output Postscaler Select bits

`1111` =1:16 Postscaler
`1110` =1:15 Postscaler
`1101` =1:14 Postscaler
`1100` =1:13 Postscaler
`1011` =1:12 Postscaler
`1010` =1:11 Postscaler
`1001` =1:10 Postscaler
`1000` =1:9 Postscaler
`0111` =1:8 Postscaler
`0110` =1:7 Postscaler
`0101` =1:6 Postscaler
`0100` =1:5 Postscaler
`0011` =1:4 Postscaler
`0010` =1:3 Postscaler
`0001` =1:2 Postscaler
`0000` =1:1 Postscaler

Note 1: In certain modes, the ON bit will be auto-cleared by hardware. See **Section22.5.5 "Software Start One-Shot Mode"**.

REGISTER 22-3: TxHLT: TIMERx CLOCK SELECTION REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PSYNC[1, 2]	CKPOL[3]	CKSYNC[4, 5]	—	MODE<3:0>[6, 7, 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 **PSYNC:** Timerx Prescaler Synchronization Enable bit[1, 2]
`1` = TMRx Prescaler Output is synchronized to Fosc/4
`0` = TMRx Prescaler Output is not synchronized to Fosc/4

bit 6 **CKPOL:** Timerx Clock Polarity Selection bit[3]
`1` = Falling edge of input clock clocks timer/prescaler
`0` = Rising edge of input clock clocks timer/prescaler

bit 5 **CKSYNC:** Timerx Clock Synchronization Enable bit[4, 5]
`1` = ON register bit is synchronized to TMR2_clk input
`0` = ON register bit is not synchronized to TMR2_clk input

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

bit 3-0 **MODE<3:0>:** Timerx Control Mode Selection bits[6, 7, 8]
See Table 22-1.

Note 1: Setting this bit ensures that reading TMRx will return a valid data value.

2: When this bit is ‘`1`’, Timer2 cannot operate in Sleep mode.

3: CKPOL should not be changed while ON = `1`.

4: Setting this bit ensures glitch-free operation when the ON is enabled or disabled.

5: When this bit is set, the timer operation will be delayed by two TMRx input clocks after the ON bit is set.

6: Unless otherwise indicated, all modes start upon ON = `1` and stop upon ON = `0` (stops occur without affecting the value of TMRx).

7: When TMRx = PRx, the next clock clears TMRx, regardless of the operating mode.

8: In edge-triggered **“**One-Shot**”** modes, the triggered-start mechanism is reset and rearmed when ON = `0`; the counter will not restart until an input edge occurs.

REGISTER 22-4: TxRST: TIMER2 EXTERNAL RESET SIGNAL SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	RSEL<3:0>			
bit 7							bit 0

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

bit 7-4	**Unimplemented:** Read as ‘`0`’
bit 3-0	**RSEL<3:0>:** Timer2 External Reset Signal Source Selection bits See Table 22-3.

TABLE 22-3: EXTERNAL RESET SOURCES

RSEL<4:0>	Timer2	Timer4	Timer6
`1111`	Reserved	Reserved	Reserved
`1110`	PWM4_out	PWM4_out	PWM4_out
`1101`	PWM3_out	PWM3_out	PWM3_out
`1100`	LC4_out	LC4_out	LC4_out
`1011`	LC3_out	LC3_out	LC3_out
`1010`	LC2_out	LC2_out	LC2_out
`1001`	LC1_out	LC1_out	LC1_out
`1000`	ZCD1_out	ZCD1_out	ZCD1_out
`0111`	TMR6_postscaled	TMR6_postscaled	Reserved
`0110`	TMR4_postscaled	Reserved	TMR4_postscaled
`0101`	Reserved	TMR2_postscaled	TMR2_postscaled
`0100`	CCP2_out	CCP2_out	CCP2_out
`0011`	CCP1_out	CCP1_out	CCP1_out
`0010`	C2OUT_sync	C2OUT_sync	C2OUT_sync
`0001`	C1OUT_sync	C1OUT_sync	C1OUT_sync
`0000`	Pin selected by T2INPPS	Pin selected by T2INPPS	Pin selected by T2INPPS

TABLE 22-4: 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
CCP1CON	EN	OE	OUT	FMT	MODE<3:0>				232
CCP2CON	EN	OE	OUT	FMT	MODE<3:0>				232
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	83
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	87
PR2	Timer2 Module Period Register								205*
TMR2	Holding Register for the 8-bit TMR2 Register								205*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				219
T2CLKCON	—	—	—	—	CS<3:0>				218
T2RST	—	—	—	—	RSEL<3:0>				221
T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					220
PR4	Timer4 Module Period Register								205*
TMR4	Holding Register for the 8-bit TMR4 Register								205*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				219
T4CLKCON	—	—	—	—	CS<3:0>				218
T4RST	—	—	—	—	RSEL<3:0>				221
T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					220
PR6	Timer6 Module Period Register								205*
TMR6	Holding Register for the 8-bit TMR6 Register								205*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				219
T6CLKCON	—	—	—	—	—	T6CS<2:0>			218
T6RST	—	—	—	—	RSEL<3:0>				221
T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					220

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

23.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains two standard Capture/Compare/PWM modules (CCP1 and CCP2).

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

23.1 Capture Mode

The Capture mode function described in this section is available and identical for all CCP modules.

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the CCPx input, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the MODE<3:0> bits of the CCPxCON register:

- Every edge (rising or falling)
- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

The CCPx capture input signal is configured by the CTS bits of the CCPxCAP register with the following options:

- CCPx pin
- Comparator 1 output (C1_OUT_sync)
- Comparator 2 output (C2_OUT_sync) (PIC16(L)F1613 only)
- Interrupt-on-change interrupt trigger (IOC_interrupt)

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 23-1 shows a simplified diagram of the capture operation.

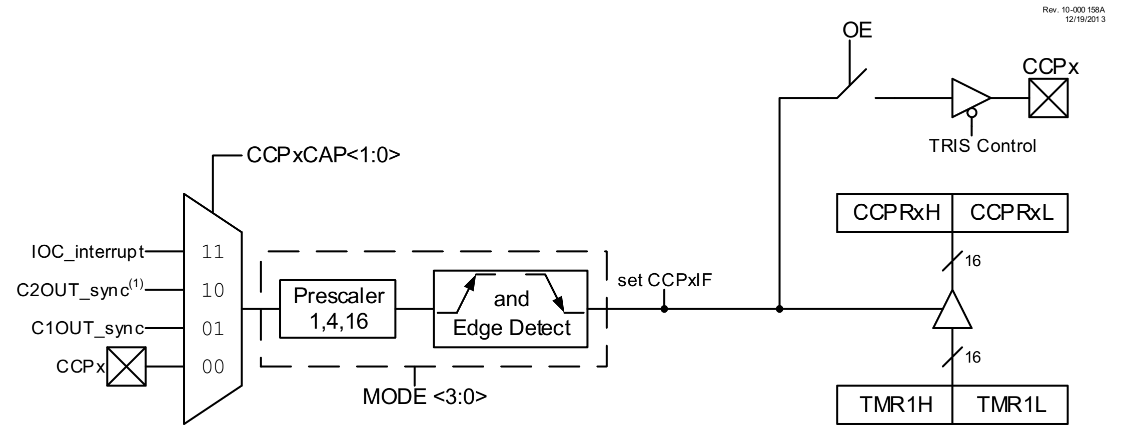
23.1.1 CCP PIN CONFIGURATION

In Capture mode, select the interrupt source using the CTS bits of the CCPxCAP register. If the CCPx pin is chosen, it should be configured as an input by setting the associated TRIS control bit.

Also, the CCP2 pin function can be moved to alternative pins using the APFCON register. Refer to **Section12.1 “Alternate Pin Function”** for more details.

Note: If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

FIGURE 23-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



Note 1: PIC16(L)F1613 Only

23.1.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section21.0 “Timer1/3/5 Module with Gate Control”** for more information on configuring Timer1.

23.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIEx register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIRx register following any change in Operating mode.

Note:	Clocking Timer1 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

23.1.4 CCP PRESCALER

There are four prescaler settings specified by the MODE<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the EN bit of the CCPxCON register before changing the prescaler.

23.1.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1 is clocked by FOSC/4, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep when Timer1 is clocked by an external clock source.

23.1.6 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function register APFCON. To determine which pins can be moved and what their default locations are upon a Reset, see **Section12.1 “Alternate Pin Function”** for more information.

23.1.7 CAPTURE OUTPUT

Whenever a capture occurs, the output of the CCP will go high for a period equal to one system clock period (1/FOSC). This output is available as an input signal to the CWG, as an auto-conversion trigger for the ADC, as an External Reset Signal for the TMR2 modules, as a window input to the SMT, and as an input to the CLC module.

23.2 Compare Mode

The Compare mode function described in this section is available and identical for all CCP modules.

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMR1H:TMR1L register pair. When a match occurs, one of the following events can occur:

- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Pulse the CCPx output
- Generate a Software Interrupt
- Optionally Reset TMR1

The action on the pin is based on the value of the MODE<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set.

All Compare modes can generate an interrupt.

Figure 23-2 shows a simplified diagram of the compare operation.

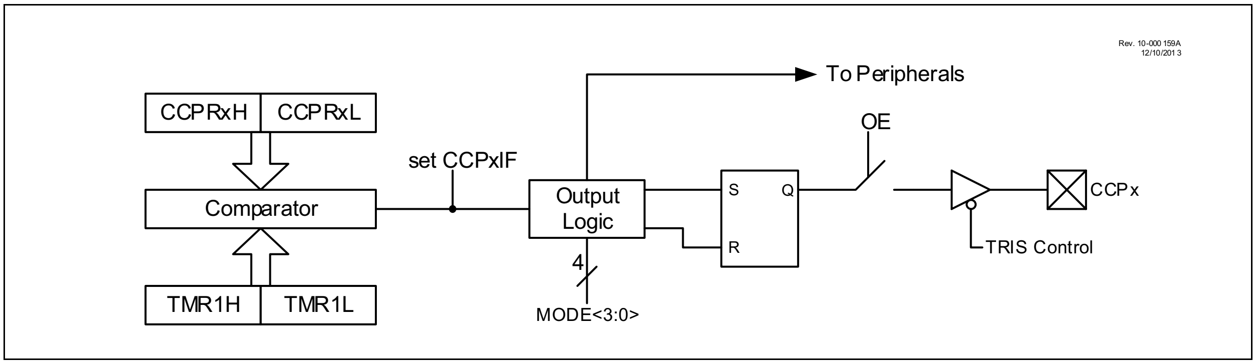
23.2.1 CCPx PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

The CCPx pin function can be moved to alternate pins using the APFCON register (Register 12-1). Refer to **Section12.1 “Alternate Pin Function”** for more details.

Note:	Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

FIGURE 23-2: COMPARE MODE OPERATION BLOCK DIAGRAM



23.2.2 TIMER1 MODE RESOURCE

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section21.0 “Timer1/3/5 Module with Gate Control”** for more information on configuring Timer1.

Note:	Clocking Timer1 from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TImer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

23.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (MODE<3:0> = `1010`), the CCPx module does not assert control of the CCPx pin (see the CCPxCON register).

23.2.4 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (FOSC) for proper operation. Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep.

23.2.5 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function register, APFCON. To determine which pins can be moved and what their default locations are upon a Reset, see **Section12.1 “Alternate Pin Function”** for more information.

23.2.6 CAPTURE OUTPUT

When in Compare mode, the CCP will provide an output upon the 16-bit value of the CCPRxH:CCPRxL register pair matching the TMR1H:TMR1L register pair. The compare output depends on which Compare mode the CCP is configured as. If the MODE bits of CCPxCON register are equal to ‘`1011`’ or ‘`1010`’, the CCP module will output high, while TMR1 is equal to CCPRxH:CCPRxL register pair. This means that the pulse width is determined by the TMR1 prescaler. If the MODE bits of CCPxCON are equal to ‘`0001`’ or ‘`0010`’, the output will toggle upon a match, going from ‘`0`’ to ‘`1`’ or vice-versa. If the MODE bits of CCPxCON are equal to ‘`1001`’, the output is cleared on a match, and if the MODE bits are equal to ‘`1000`’, the output is set on a match. This output is available as an input signal to the CWG, as an auto-conversion trigger for the ADC, as an external Reset signal for the TMR2 modules, as a window input to the SMT, and as an input to the CLC module.

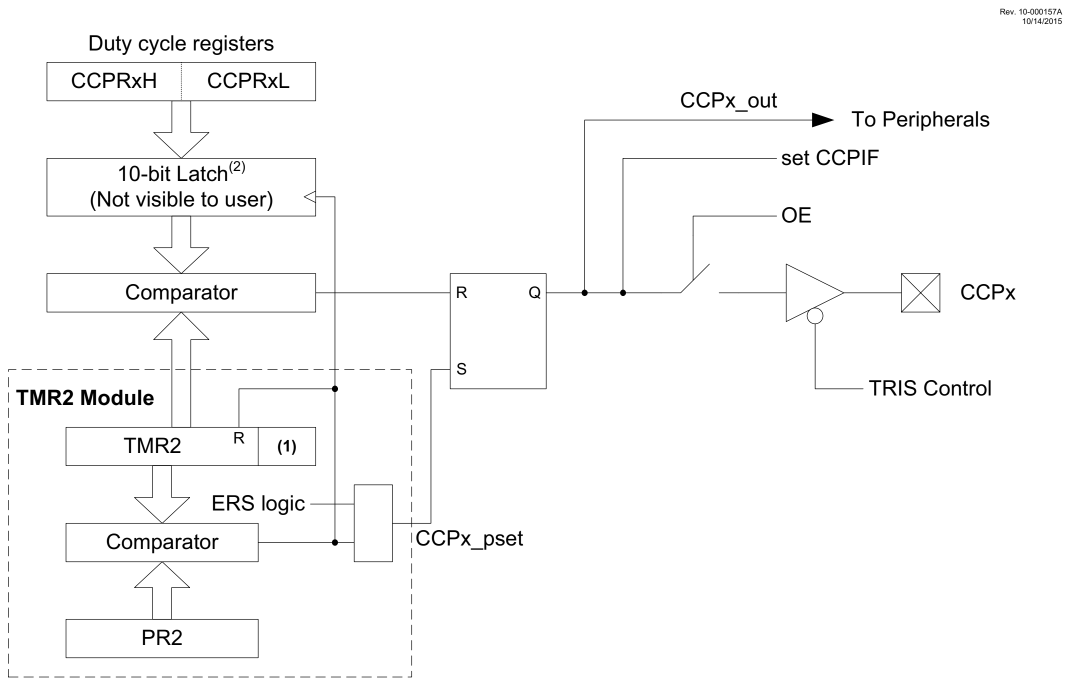
23.3 PWM Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the on state and the low portion of the signal is considered the off state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

FIGURE 23-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with two bits generated by Fosc or two bits of the internal prescaler to create 10-bit time-base.

2: The alignment of the 10 bits from the CCPR register is determined by the FMT bit. Refer to Figure 23-4 for more information.

23.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2/4/6 registers
- T2CON/T4CON/T6CON registers
- CCPRxH:CCPRxL register pair

Figure shows a simplified block diagram of PWM operation.

Note 1: The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.

2: Clearing the CCPxCON register will relinquish control of the CCPx pin.

23.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Disable the CCPx pin output driver by setting the associated TRIS bit.
2. Determine which timer will be used to clock the CCP; Timer2/4/6.
3. Load the associated PR2/4/6 register with the PWM period value.
4. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
5. Load the CCPRxH:CCPRxL register pair with the PWM duty cycle value.
6. Configure and start Timer2/4/6:
   - Clear the TMR2IF/TMR4IF/TMR6IF interrupt flag bit of the PIRx register. See Note below.
   - Configure the CKPS bits of the TxCON register with the Timer prescale value.
   - Enable the Timer by setting the ON bit of the TxCON register.
7. Enable PWM output pin:
   - Wait until the Timer overflows and the TMR2IF/TMR4IF/TMR6IF bit of the PIRx register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRIS bit.

Note: In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

23.4 CCP/PWM Clock Selection

The PIC12(L)F1612/16(L)F1613 allows each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

As there are up to three 8-bit timers with auto-reload (Timer2/4/6), PWM mode on the CCP and PWM modules can use any of these timers.

The CCPTMRS register is used to select which timer is used.

23.4.1 USING THE TMR2/4/6 WITH THE CCP MODULE

This device has a new version of the TMR2 module that has many new modes, which allow for greater customization and control of the PWM signals than older parts. Refer to **Section23.5 “Operation Examples”** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the FOSC/4 clock source selected.

23.4.2 PWM PERIOD

The PWM period is specified by the PR2/4/6 register of Timer2/4/6. The PWM period can be calculated using the formula of Equation 23-1.

EQUATION 23-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

Note 1: TOSC = 1/FOSC

When TMR2/4/6 is equal to its respective PR2/4/6 register, the following three events occur on the next increment cycle:

- TMR2/4/6 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from the CCPRxH:CCPRxL pair into the internal 10-bit latch.

Note:	The Timer postscaler (see Figure) is not used in the determination of the PWM frequency.

23.4.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to two registers: the CCPRxH:CCPRxL register pair. Where the particular bits go is determined by the FMT bit of the CCPxCON register. If FMT = `0`, the two Most Significant bits of the duty cycle value should be written to bits <1:0> of CCPRxH register and the remaining eight bits to the CCPRxL register. If FMT = `1`, the Least Significant two bits of the duty cycle should be written to bits <7:6> of the CCPRxL register and the Most Significant eight bits to the CCPRxH register. This is illustrated in Figure 23-4. These bits can be written at any time. The duty cycle value is not latched into the internal latch until after the period completes (i.e., a match between PR2/4/6 and TMR2/4/6 registers occurs).

Equation 23-2 is used to calculate the PWM pulse width. Equation 23-3 is used to calculate the PWM duty cycle ratio.

EQUATION 23-2: PULSE WIDTH

$$Pulse\ Width = CCPRxH{:}CCPRxL \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

EQUATION 23-3: DUTY CYCLE RATIO

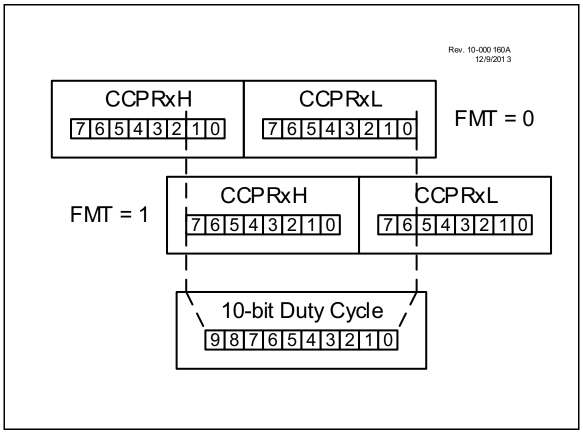
$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL)}{4(PRx + 1)}$$

The PWM duty cycle registers are double buffered for glitchless PWM operation.

The 8-bit timer TMR2/4/6 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2/4/6 prescaler is set to 1:1.

When the 10-bit time base matches the internal buffer register, then the CCPx pin is cleared (see Figure).

FIGURE 23-4: CCPx DUTY-CYCLE ALIGNMENT



23.4.4 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2/4/6 is 255. The resolution is a function of the PR2/4/6 register value as shown by Equation 23-4.

EQUATION 23-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)} \text{ bits}$$

Note: If the pulse width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

TABLE 23-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6

TABLE 23-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

23.4.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section5.0 “Oscillator Module”** for additional details.

23.4.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

23.4.7 PWM OUTPUT

The output of the CCP in PWM mode is the PWM signal generated by the module and described above. This output is available as an input signal to the CWG, as an auto-conversion trigger for the ADC, as an external Reset signal for the TMR2 modules, as a window input to the SMT, and as an input to the CLC module.

23.5 Register Definitions: CCP Control

REGISTER 23-1: CCPxCON: CCPx CONTROL REGISTER

R/W-0/0	R/W-0/0	R-x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	OE	OUT	FMT	MODE<3:0>			
bit 7							bit 0

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

bit 7 **EN:** CCPx Module Enable bit

1 = CCPx is enabled
0 = CCPx is disabled

bit 6 **OE:** CCPx Output Enable bit

1 = CxOUT is present on the CxOUT pin. Requires that the associated TRIS bit be cleared to drive the pin. Not affected by CxON.
0 = CxOUT is internal only

bit 5 **OUT:** CCPx Output Data bit (read-only)

bit 4 **FMT:** CCPW (Pulse-Width) Alignment bit

If MODE = PWM Mode:

1 = Left-aligned format, CCPRxH <7> is the MSb of the PWM duty cycle
0 = Right-aligned format, CCPRxL<0> is the LSb of the PWM duty cycle

bit 3-0 **MODE<3:0>:** CCPx Mode Selection bit

11xx = PWM mode

1011 = Compare mode: Pulse output, clear TMR1
1010 = Compare mode: Pulse output (0 - 1 - 0)
1001 = Compare mode: clear output on compare match
1000 = Compare mode: set output on compare match

0111 = Capture mode: every 16th rising edge
0110 = Capture mode: every 4th rising edge
0101 = Capture mode: every rising edge
0100 = Capture mode: every falling edge

0011 = Capture mode: every rising or falling edge
0010 = Compare mode: toggle output on match
0001 = Compare mode: Toggle output and clear TMR1 on match
0000 = Capture/Compare/PWM off (resets CCPx module) (reserved for backwards compatibility)

REGISTER 23-2: CCPTMRS: PWM TIMER SELECTION CONTROL REGISTER 0

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	C2TSEL<1:0>		C1TSEL<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 7-4 **Unimplemented:** Read as ‘0’

bit 3-2 **C2TSEL<1:0>:** CCP2 (PWM2) Timer Selection bits
11 = Reserved
10 = CCP2 is based off Timer6 in PWM mode
01 = CCP2 is based off Timer4 in PWM mode
00 = CCP2 is based off Timer2 in PWM mode

bit 1-0 **C1TSEL<1:0>:** CCP1 (PWM1) Timer Selection bits
11 = Reserved
10 = CCP1 is based off Timer6 in PWM mode
01 = CCP1 is based off Timer4 in PWM mode
00 = CCP1 is based off Timer2 in PWM mode

REGISTER 23-3: CCPRxL: CCPx 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
CCPR<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 MODE = Capture Mode

CCPRxL<7:0>: LSB of captured TMR1 value

MODE = Compare Mode

CCPRxL<7:0>: LSB compared to TMR1 value

MODE = PWM Mode && FMT = `0`

CCPRxL<7:0>: CCPW<7:0> — Pulse width Least Significant eight bits

MODE = PWM Mode && FMT = `1`

CCPRxL<7:6>: CCPW<1:0> — Pulse width Least Significant two bits

CCPRxL<5:0>: Not used

REGISTER 23-4: CCPRxH: CCPx HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPR<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 MODE = Capture Mode:
CCPRxH<7:0>: MSB of captured TMR1 value
MODE = Compare Mode:
CCPRxH<7:0>: MSB compared to TMR1 value
MODE = PWM Mode && FMT = `0`:
CCPRxH<7:2>: Not used
CCPRxH<1:0>: CCPW<9:8> — Pulse width Most Significant two bits
MODE = PWM Mode && FMT = `1`:
CCPRxH<7:0>: CCPW<9:2> — Pulse width Most Significant eight bits

REGISTER 23-5: CCPxCAP: CCPx CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	CTS<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 Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-2 **Unimplemented:** Read as ‘`0`’

bit 1-0 **CTS<1:0>:** Capture Trigger Input Selection bits
`11` = IOC_interrupt
`10` = C2_OUT_sync[1]
`01` = C1_OUT_sync
`00` = CCPx pin

Note 1: PIC16(L)F1613 only. Reserved on PIC12(L)F1612.

TABLE 23-3: SUMMARY OF REGISTERS ASSOCIATED WITH STANDARD PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	—	CWGASEL(2)	CWGBSEL(2)	—	T1GSEL	—	CCP2SEL(3)	CCP1SEL(2)	132
CCP1CON	EN	OE	OUT	FMT	MODE<3:0>				232
CCP2CON	EN	OE	OUT	FMT	MODE<3:0>				232
CCPRxL	Capture/Compare/PWM Register x (LSB)								234
CCPRxH	Capture/Compare/PWM Register x (MSB)								235
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		233
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	82
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	83
PIE2	—	C2IE(1)	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	84
PR2	Timer2 Period Register								235*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				254
TMR2	Timer2 Module Register								235*
PR4	Timer4 Period Register								235*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				254
TMR4	Timer4 Module Register								235*
PR6	Timer6 Period Register								235*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				254
TMR6	Timer6 Module Register								235*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	135

Legend: — = Unimplemented location, read as ‘0’. Shaded cells are not used by the PWM.
* Page provides register information.

Note 1: Unimplemented, read as ‘1’.

24.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces half-bridge, full-bridge, and steering of PWM waveforms. It is backwards compatible with previous ECCP functions.

The CWG has the following features:

- Six operating modes:
  - Synchronous Steering mode
  - Asynchronous Steering mode
  - Full-Bridge mode, Forward (PIC16(L)F1613 only)
  - Full-Bridge mode, Reverse (PIC16(L)F1613 only)
  - Half-Bridge mode
  - Push-Pull mode
- Output polarity control
- Output steering
  - Synchronized to rising event
  - Immediate effect
- Independent 6-bit rising and falling event dead-band timers
  - Clocked dead band
  - Independent rising and falling dead-band enables
- Auto-shutdown control with:
  - Selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control

24.1 Fundamental Operation

The CWG module can operate in six different modes, as specified by MODE of the CWGxCON0 register:

- Half-Bridge mode (Figure 24-9)
- Push-Pull mode (Figure 24-2)
  - Full-Bridge mode, Forward (Figure 24-3) (PIC16(L)F1613 only)
  - Full-Bridge mode, Reverse (Figure 24-3) (PIC16(L)F1613 only)
- Steering mode (Figure 24-10)
- Synchronous Steering mode (Figure 24-11)

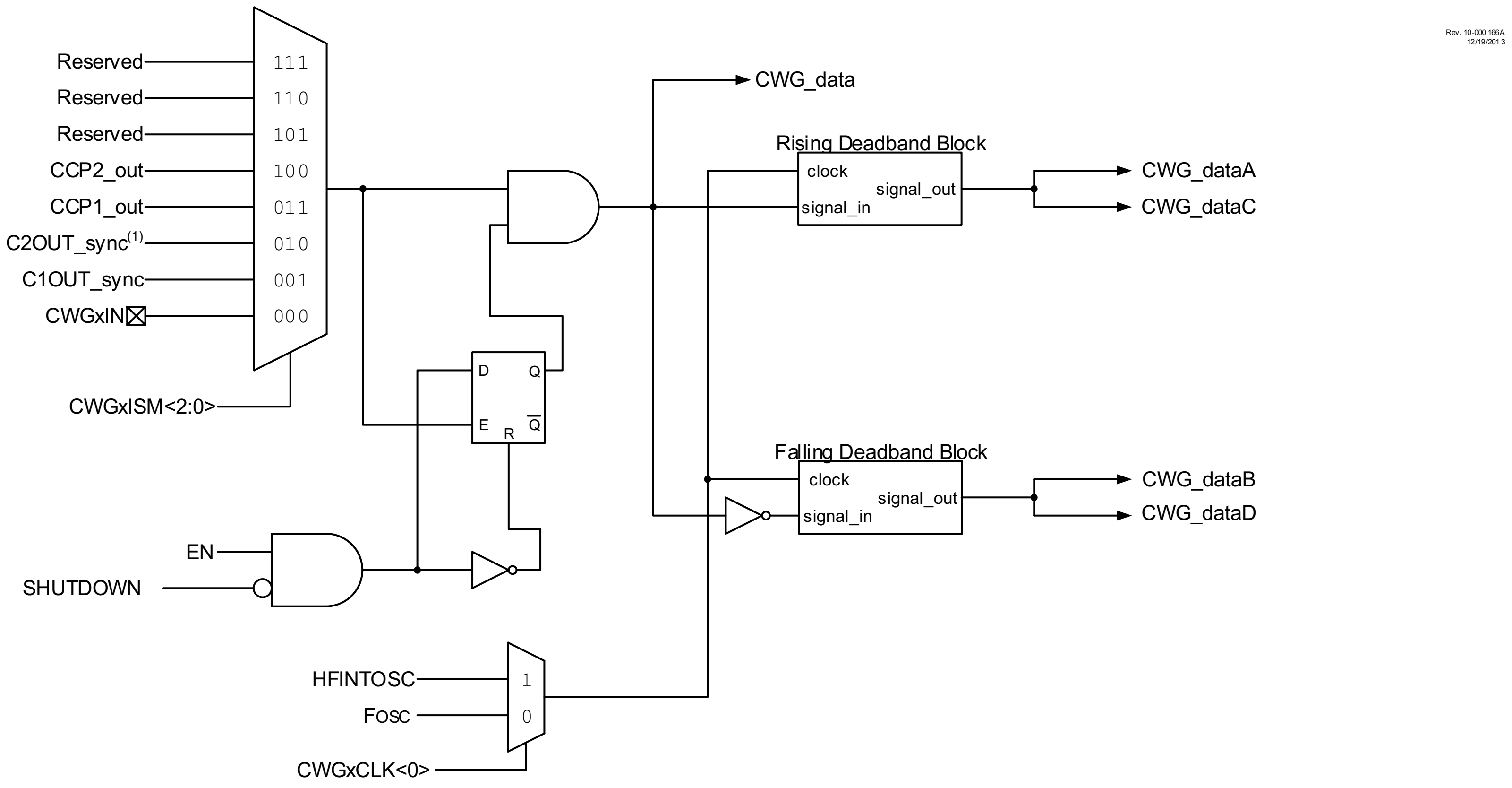
It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. Thus, all output modes support auto-shutdown, which is covered in **24.10 “Auto-Shutdown”**.

24.1.1 HALF-BRIDGE MODE

In Half-Bridge mode, two output signals are generated as true and inverted versions of the input as illustrated in Figure 24-9. A non-overlap (dead-band) time is inserted between the two outputs to prevent shoot through current in various power supply applications. Dead-band control is described in **Section 24.5 “Dead-Band Control”**.

The unused outputs CWGxC and CWGxD drive similar signals, with polarity independently controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

FIGURE 24-1: SIMPLIFIED CWG BLOCK DIAGRAM (HALF BRIDGE MODE)



Note 1: PIC16(L)F1613 Only

24.1.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 24-2. This alternation creates the push-pull effect required for driving some transformer-based power supply designs.

The push-pull sequencer is reset whenever EN = `0` or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWGxA.

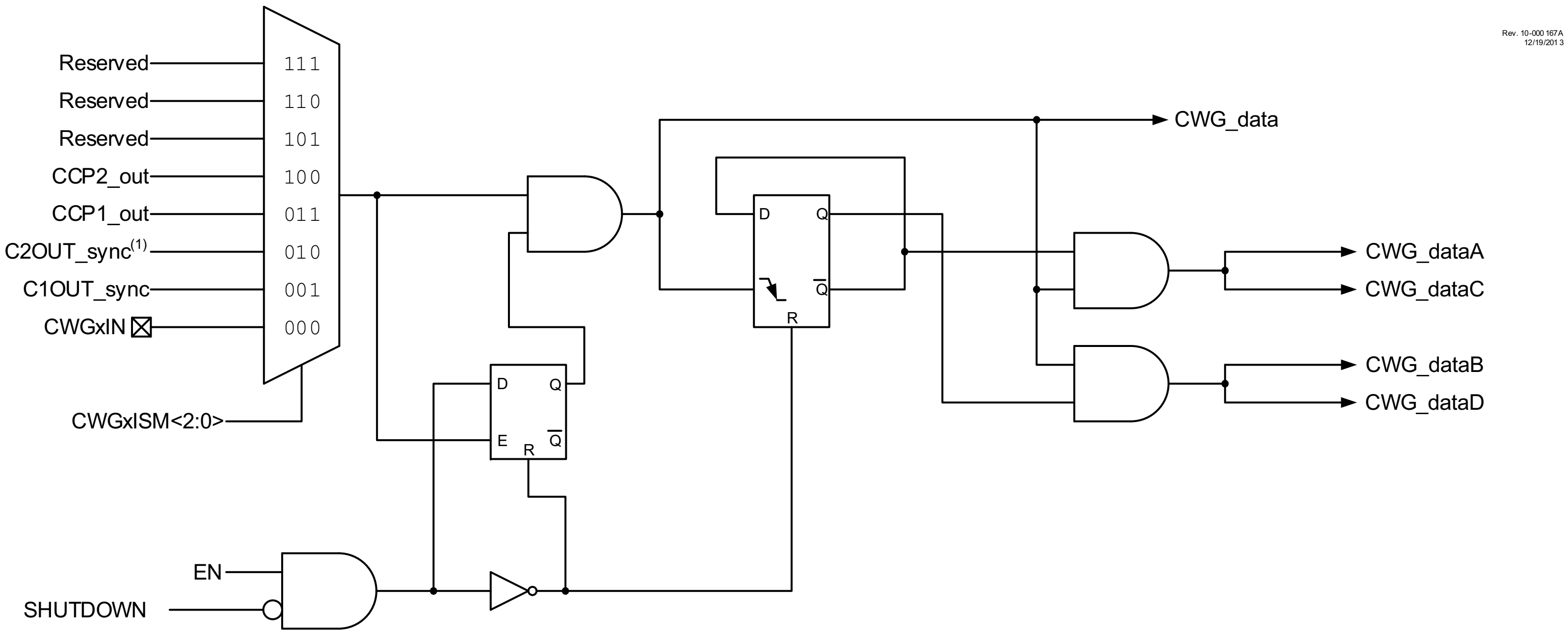
The unused outputs CWGxC and CWGxD drive copies of CWGxA and CWGxB, respectively, but with polarity controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

24.1.3 FULL-BRIDGE MODES

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the input data signal. In Forward Full-Bridge mode, CWGxA is driven to its active state, CWGxB and CWGxC are driven to their inactive state, and CWGxD is modulated by the input signal. In Reverse Full-Bridge mode, CWGxC is driven to its active state, CWGxA and CWGxD are driven to their inactive states, and CWGxB is modulated by the input signal. In Full-Bridge mode, the dead-band period is used when there is a switch from forward to reverse or vice-versa. This dead-band control is described in **Section 24.5 “Dead-Band Control”**, with additional details in **Section 24.6 “Rising Edge and Reverse Dead Band”** and **Section 24.7 “Falling Edge and Forward Dead Band”**.

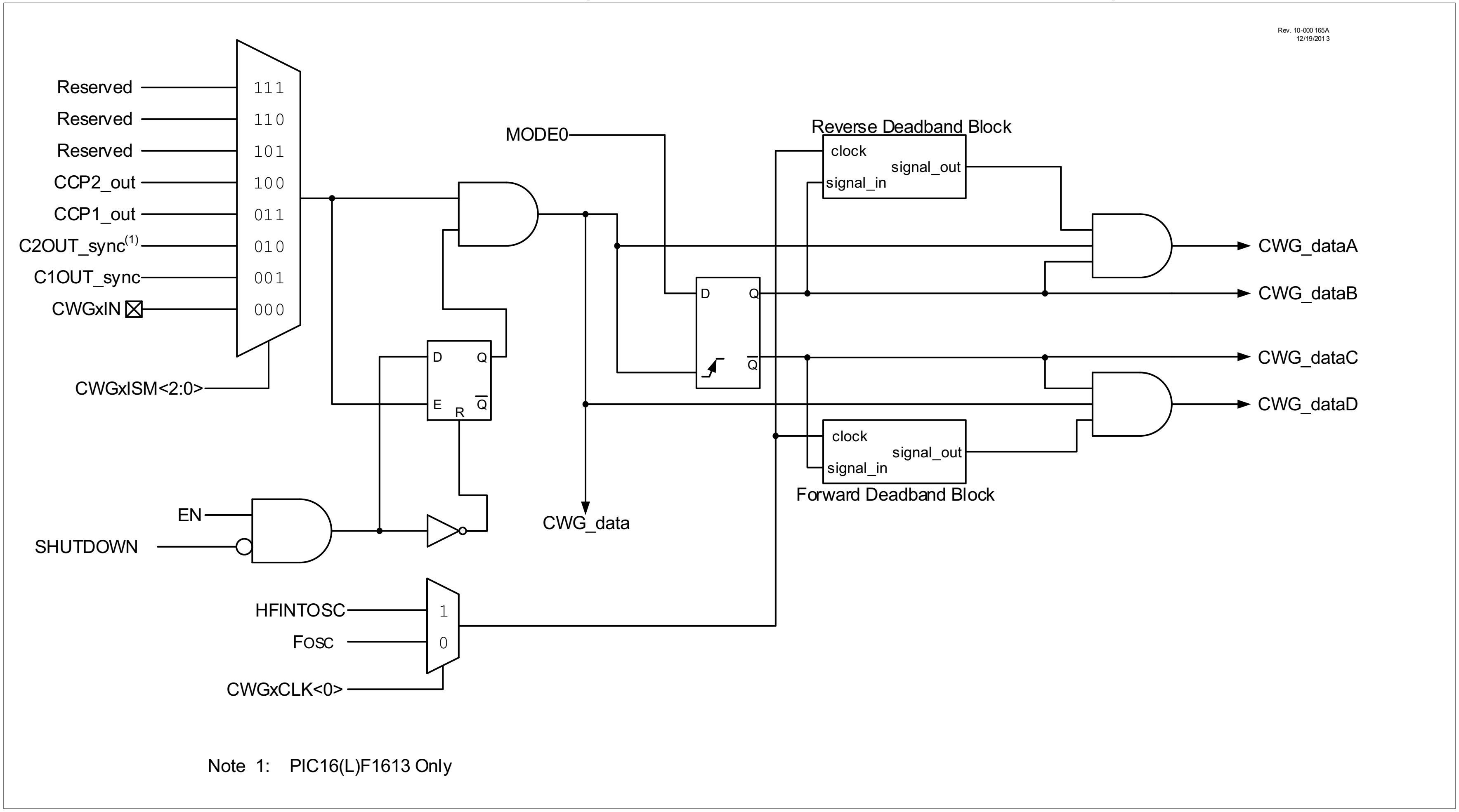
The mode selection may be toggled between forward and reverse by toggling the MODE<0> bit of the CWGxCON0 while keeping MODE<2:1> static, without disabling the CWG module.

FIGURE 24-2: SIMPLIFIED CWG BLOCK DIAGRAM (PUSH-PULL MODE)



Note 1: PIC16(L)F1613 Only

FIGURE 24-3: SIMPLIFIED CWG BLOCK DIAGRAM (FORWARD AND REVERSE FULL-BRIDGE MODES)



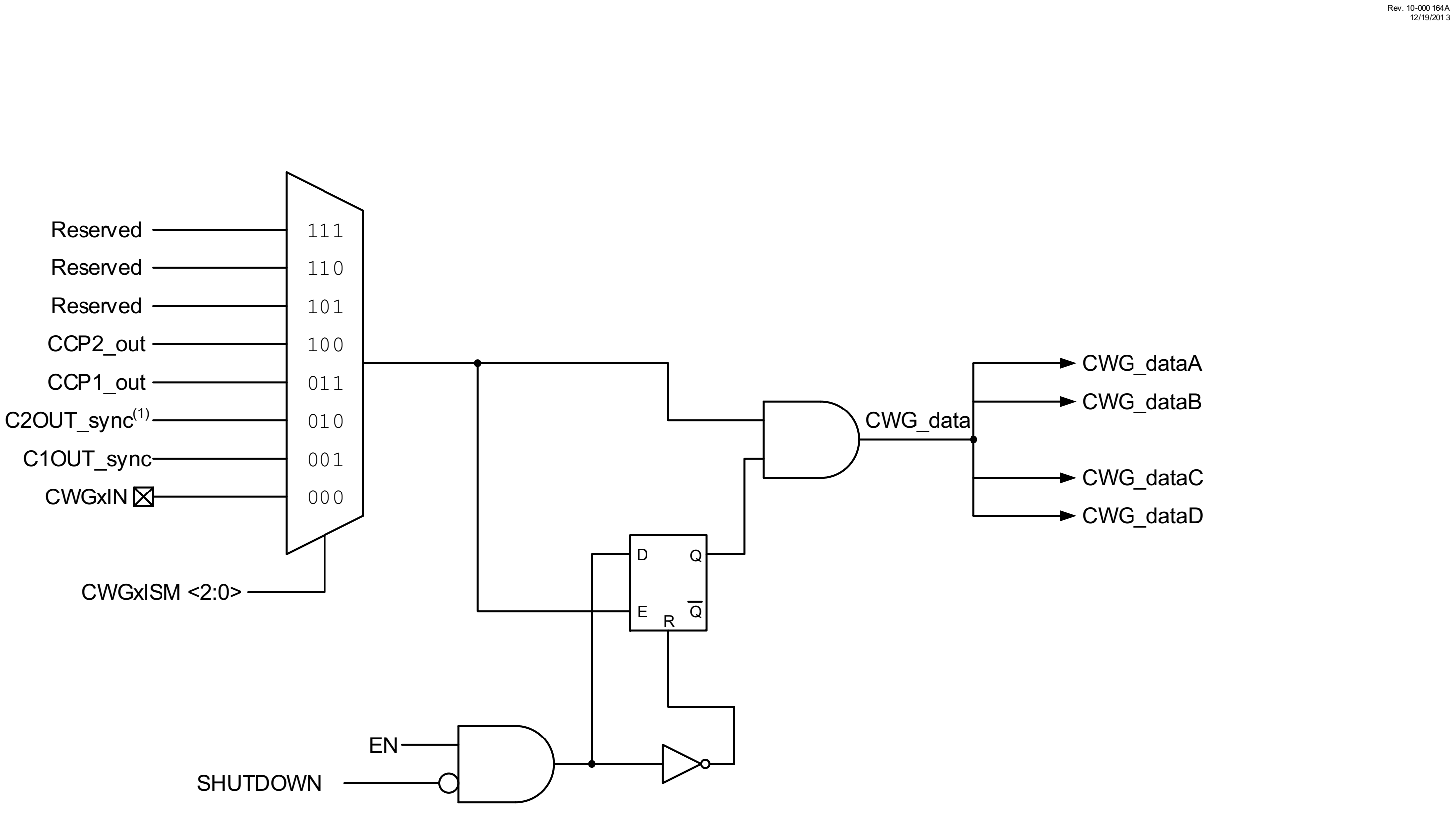
Note 1: PIC16(L)F1613 Only

24.1.4 STEERING MODES

In Steering modes, the data input can be steered to any or all of the four CWG output pins. In Synchronous Steering mode, changes to steering selection registers take effect on the next rising input.

In Non-Synchronous mode, steering takes effect on the next instruction cycle. Additional details are provided in **Section 24.9 "CWG Steering Mode"**.

FIGURE 24-4: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



Note 1: PIC16(L)F1613 Only

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 CS bit of the CWGxCLKCON register.

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 pin	CWGxIN pin
Comparator C1	C1_OUT_sync
Comparator C2[(1)]	C2_OUT_sync
CCP1	CCP1_out
CCP2	CCP2_out

Note 1: PIC16(L)F1613 only.

The input sources are selected using the CWGxISM register.

24.4 Output Control

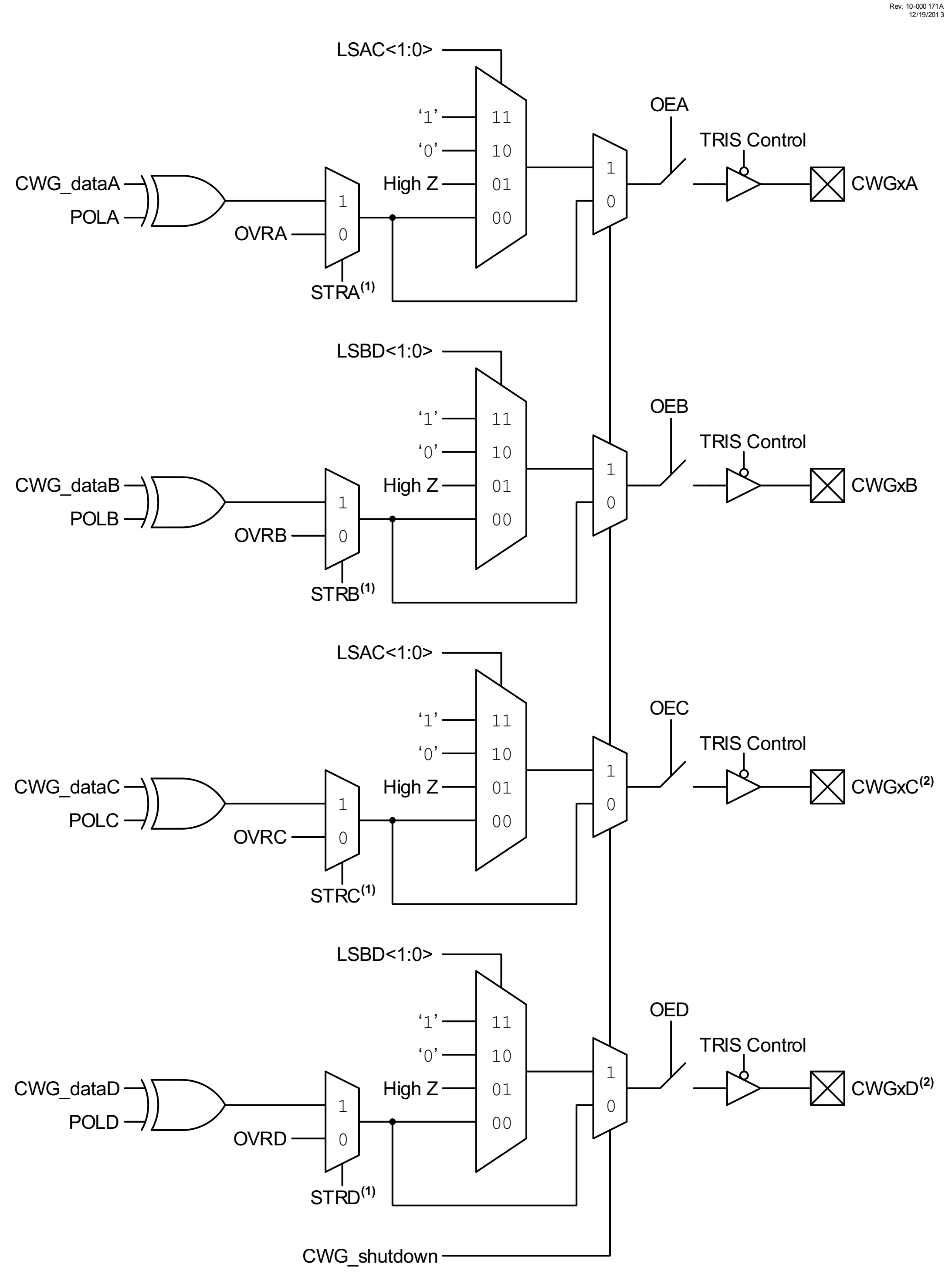
24.4.1 OUTPUT ENABLES

Each CWG output pin has individual output enable control. Output enables are selected with the Gx1OEx <3:0> bits. When an output enable control is cleared, the module asserts no control over the pin. When an output enable is set, the override value or active PWM waveform is applied to the pin per the port priority selection. The output pin enables are dependent on the module enable bit, EN of the CWGxCON0 register. When EN is cleared, CWG output enables and CWG drive levels have no effect.

24.4.2 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the POLx bits of the CWGxCON1. Auto-shutdown and steering options are unaffected by polarity.

FIGURE 24-5: CWG OUTPUT BLOCK DIAGRAM



Note 1: STRx is held to 1 in all modes other than Output Steering Mode.
2: PIC16(L)F1613 ONLY

24.5 Dead-Band Control

The dead-band control provides non-overlapping PWM signals to prevent shoot-through current in PWM switches. Dead-band operation is employed for Half-Bridge and Full-Bridge modes. The CWG contains two 6-bit dead-band counters. One is used for the rising edge of the input source control in Half-Bridge mode or for reverse dead-band Full-Bridge mode. The other is used for the falling edge of the input source control in Half-Bridge mode or for forward dead band in Full-Bridge mode.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling dead-band counter registers. See CWGxDBR and CWGxDBF registers, respectively.

24.5.1 DEAD-BAND FUNCTIONALITY IN HALF-BRIDGE MODE

In Half-Bridge mode, the dead-band counters dictate the delay between the falling edge of the normal output and the rising edge of the inverted output. This can be seen in Figure 24-9.

24.5.2 DEAD-BAND FUNCTIONALITY IN FULL-BRIDGE MODE

In Full-Bridge mode, the dead-band counters are used when undergoing a direction change. The MODE<0> bit of the CWGxCON0 register can be set or cleared while the CWG is running, allowing for changes from Forward to Reverse mode. The CWGxA and CWGxC signals will change immediately upon the first rising input edge following a direction change, but the modulated signals (CWGxB or CWGxD, depending on the direction of the change) will experience a delay dictated by the dead-band counters. This is demonstrated in Figure 24-3.

24.6 Rising Edge and Reverse Dead Band

CWGxDBR controls the rising edge dead-band time at the leading edge of CWGxA (Half-Bridge mode) or the leading edge of CWGxB (Full-Bridge mode). The CWGxDBR value is double-buffered. When EN = `0`, the CWGxDBR register is loaded immediately when CWGxDBR is written. When EN = `1`, then software must set the LD bit of the CWGxCON0 register, and the buffer will be loaded at the next falling edge of the CWG input signal. If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

24.7 Falling Edge and Forward Dead Band

CWGxDBF controls the dead-band time at the leading edge of CWGxB (Half-Bridge mode) or the leading edge of CWGxD (Full-Bridge mode). The CWGxDBF value is double-buffered. When EN = `0`, the CWGxDBF register is loaded immediately when CWGxDBF is written. When EN = `1` then software must set the LD bit of the CWGxCON0 register, and the buffer will be loaded at the next falling edge of the CWG input signal. If the input source signal is not present for enough time for the count to be completed, no output will be seen on the respective output.

Refer to Figure 24.6 and Figure 24-7 for examples.

FIGURE 24-6: DEAD-BAND OPERATION CWGXDBR = 0X01, CWGXDBF = 0X02

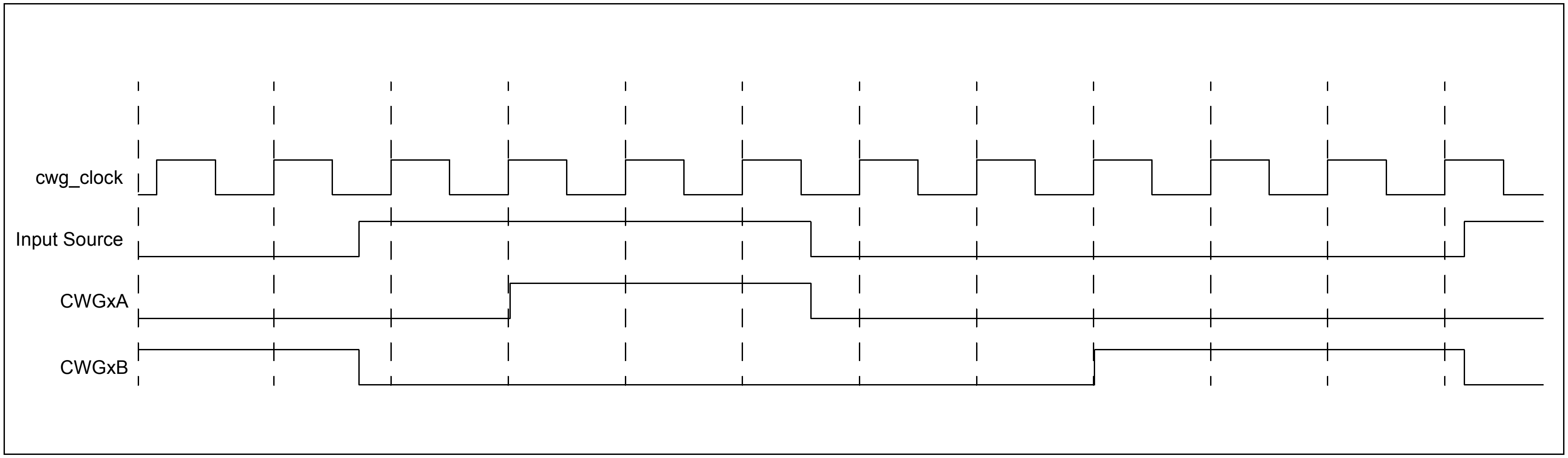
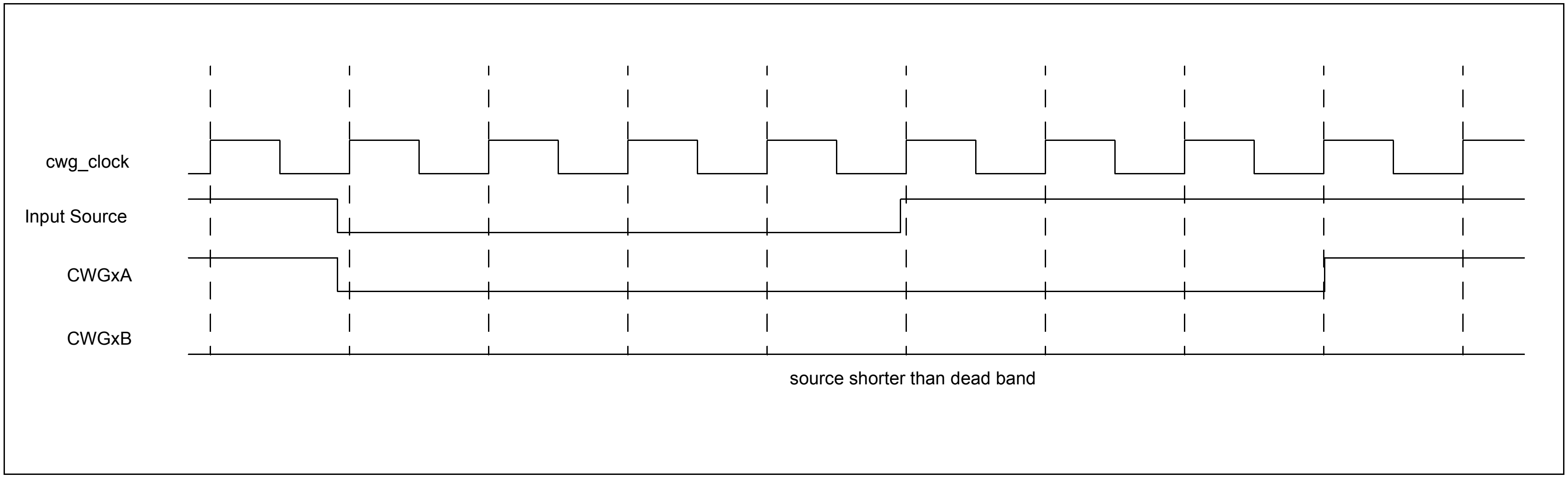


FIGURE 24-7: DEAD-BAND OPERATION, CWGXDBR = 0X03, CWGXDBF = 0X04, SOURCE SHORTER THAN DEAD BAND



24.8 Dead-Band Uncertainty

When the rising and falling edges of the input source are asynchronous to the CWG clock, it creates uncertainty in the dead-band time delay. The maximum uncertainty is equal to one CWG clock period. Refer to Equation 24-1 for more details.

EQUATION 24-1: DEAD-BAND UNCERTAINTY

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

Example:

$$F_{CWG\_CLOCK} = 16\ MHz$$

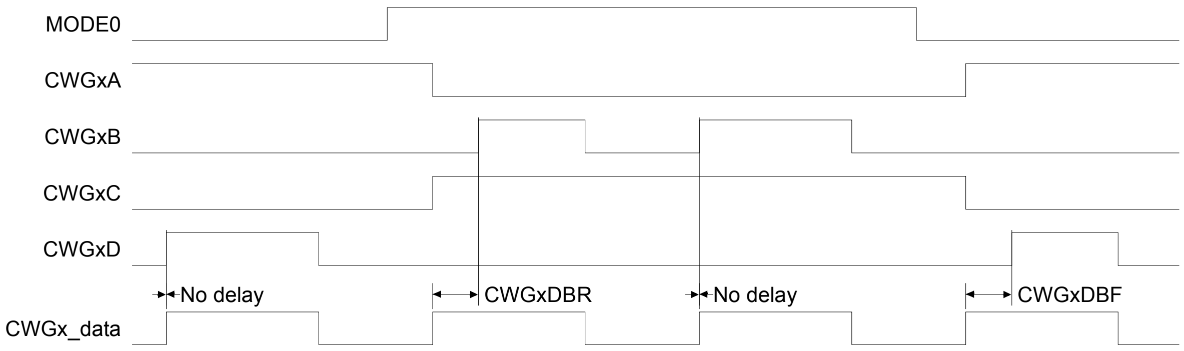
Therefore:

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

$$= \frac{1}{16MHz}$$

$$= 62.5ns$$

FIGURE 24-8: EXAMPLE OF PWM DIRECTION CHANGE

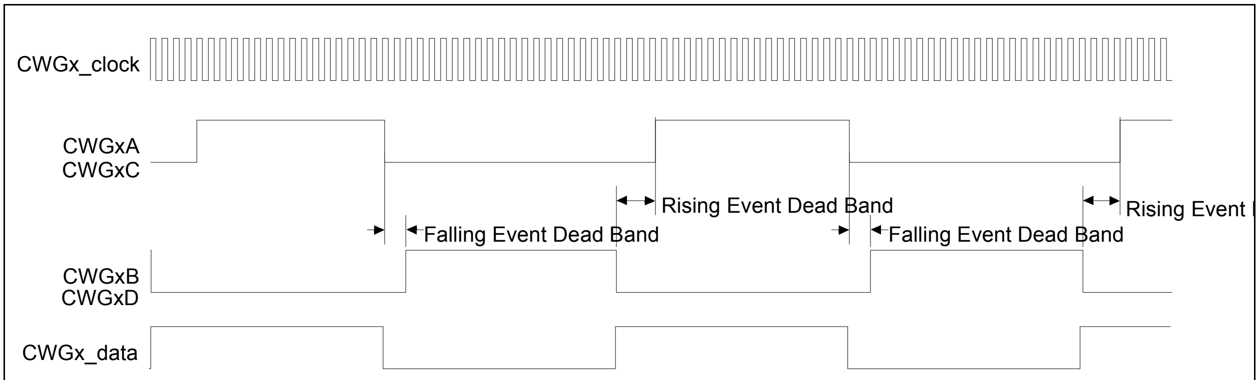


Note 1: WGPOL{ABCD} = `0`

2: The direction bit MODE<0> (Register 24-1) can be written any time during the PWM cycle, and takes effect at the next rising CWGx_data.

3: When changing directions, CWGxA and CWGxC switch at rising CWGx_data; modulated CWGxB and CWGxD are held inactive for the dead band duration shown; dead band affects only the first pulse after the direction change.

FIGURE 24-9: CWG HALF-BRIDGE MODE OPERATION



Note: CWGx_rising_src = CCP1_out, CWGx_falling_src = ~CCP1_out

24.9 CWG Steering Mode

In Steering mode (MODE = `00x`), the CWG allows any combination of the CWGxx pins to be the modulated signal. The same signal can be simultaneously available on multiple pins, or a fixed-value output can be presented.

When the respective STRx bit of CWGxOCON0 is ‘`0`’, the corresponding pin is held at the level defined. When the respective STRx bit of CWGxOCON0 is ‘`1`’, the pin is driven by the input data signal. The user can assign the input data signal to one, two, three, or all four output pins.

The POLx bits of the CWGxCON1 register control the signal polarity only when STRx = `1`.

The CWG auto-shutdown operation also applies in Steering modes as described in **Section 24.10 “Auto-Shutdown”**. An auto-shutdown event will only affect pins that have STRx = `1`.

24.9.1 STEERING SYNCHRONIZATION

Changing the MODE bits allows for two modes of steering, synchronous and asynchronous.

When MODE = `000`, the steering event is asynchronous and will happen at the end of the instruction that writes to STRx (that is, immediately). In this case, the output signal at the output pin may be an incomplete waveform. This can be useful for immediately removing a signal from the pin.

When MODE = `001`, the steering update is synchronous and occurs at the beginning of the next rising edge of the input data signal. In this case, steering the output on/off will always produce a complete waveform.

Figure 24-10 and Figure 24-11 illustrate the timing of asynchronous and synchronous steering, respectively.

FIGURE 24-10: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (MODE<2:0> = `000`)

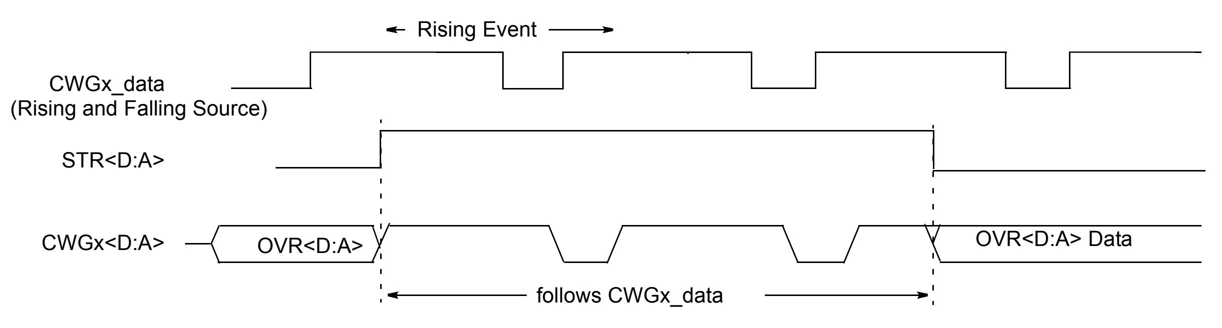
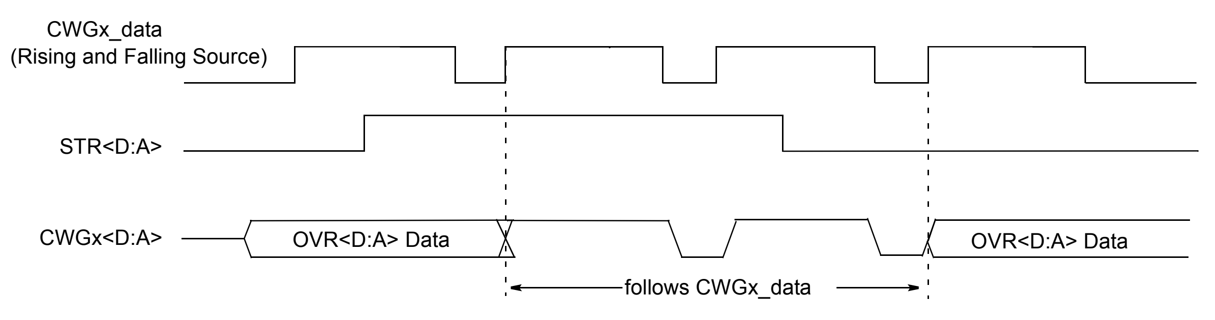


FIGURE 24-11: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (MODE<2:0> = `001`)



24.10 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in Figure 24-12.

24.10.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

24.10.1.1 Software Generated Shutdown

Setting the SHUTDOWN bit of the CWGxAS0 register will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event.

24.10.2 EXTERNAL INPUT SOURCE

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the selected override levels without software delay. Several input sources can be selected to cause a shutdown condition. All input sources are active-low. The sources are:

- Comparator C1_OUT_sync
- Comparator C2_OUT_sync (PIC16(L)F1613 only)
- Timer2 – TMR2_postscaled
- Timer4 – TMR4_postscaled
- Timer6 – TMR6_postscaled
- CWGxIN input pin

Shutdown inputs are selected using the CWGxAS1 register (Register 24-6).

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.

24.11 Operation During Sleep

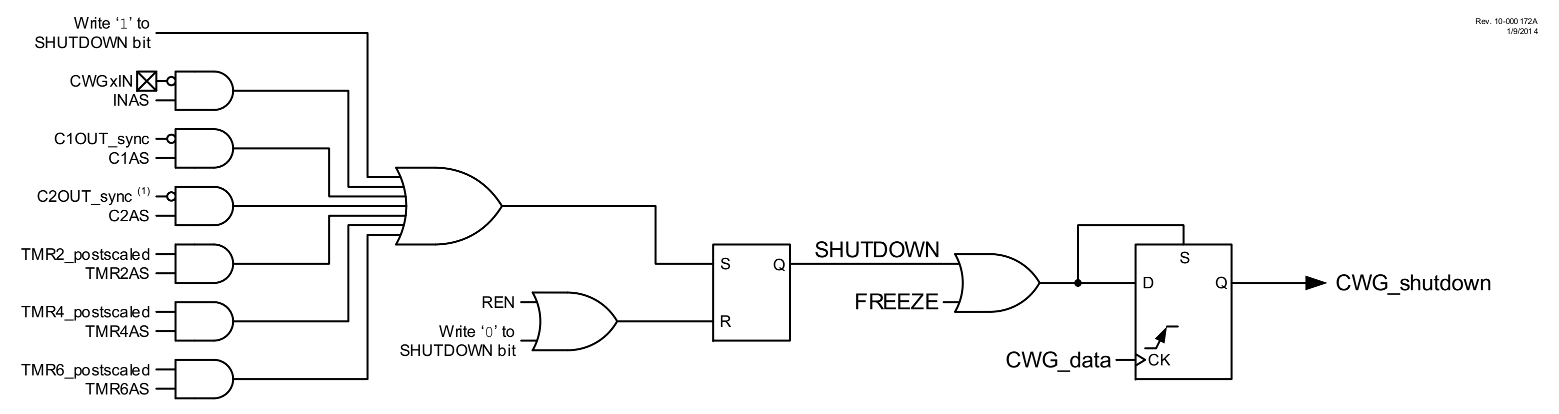
The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep when all the following conditions are met:

- CWG module is enabled
- Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.

FIGURE 24-12: CWG SHUTDOWN BLOCK DIAGRAM



Note 1: PIC16(L)F1613 only

24.12 Configuring the CWG

The following steps illustrate how to properly configure the CWG.

1. Ensure that the TRIS control bits corresponding to the desired CWG pins for your application are set so that the pins are configured as inputs.
2. Clear the EN bit, if not already cleared.
3. Set desired mode of operation with the MODE bits.
4. Set desired dead-band times, if applicable to mode, with the CWGxDBR and CWGxDBF registers.
5. Setup the following controls in the CWGxAS0 and CWGxAS1 registers.
   a. Select the desired shutdown source.
   b. Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
   c. Set which pins will be affected by auto-shutdown with the CWGxAS1 register.
   d. Set the SHUTDOWN bit and clear the REN bit.
6. Select the desired input source using the CWGxISM register.
7. Configure the following controls.
   a. Select desired clock source using the CWGxCLKCON register.
   b. Select the desired output polarities using the CWGxCON1 register.
   c. Set the output enables for the desired outputs.
8. Set the EN bit.
9. Clear TRIS control bits corresponding to the desired output pins to configure these pins as outputs.
10. If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit to start the CWG.

24.12.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the LSBD and LSAC bits of the CWGxAS0 register. LSBD<1:0> controls the CWGxB and D override levels and LSAC<1:0> controls the CWGxA and C override levels. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not affect the override level.

24.12.2 AUTO-SHUTDOWN RESTART

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

- Software controlled
- Auto-restart

The restart method is selected with the REN bit of the CWGxAS0 register. Waveforms of software controlled and automatic restarts are shown in Figure 24-13 and Figure 24-14.

24.12.2.1 Software Controlled Restart

When the REN bit of the CWGxAS0 register is cleared, the CWG must be restarted after an auto-shutdown event by software. Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the SHUTDOWN bit will remain set. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

24.12.2.2 Auto-Restart

When the REN bit of the CWGxAS0 register is set, the CWG will restart from the auto-shutdown state automatically. The SHUTDOWN bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

24.12.3 ALTERNATE OUTPUT PINS

This module incorporates outputs that can be moved to alternate pins with the use of the alternate pin function register APFCON. To determine which outputs can be moved and what their default pins are upon a Reset, see **Section 12.1 “Alternate Pin Function”** for more information.

FIGURE 24-13: SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (REN = `0`, LSAC = `01`, LSBD = `01`)

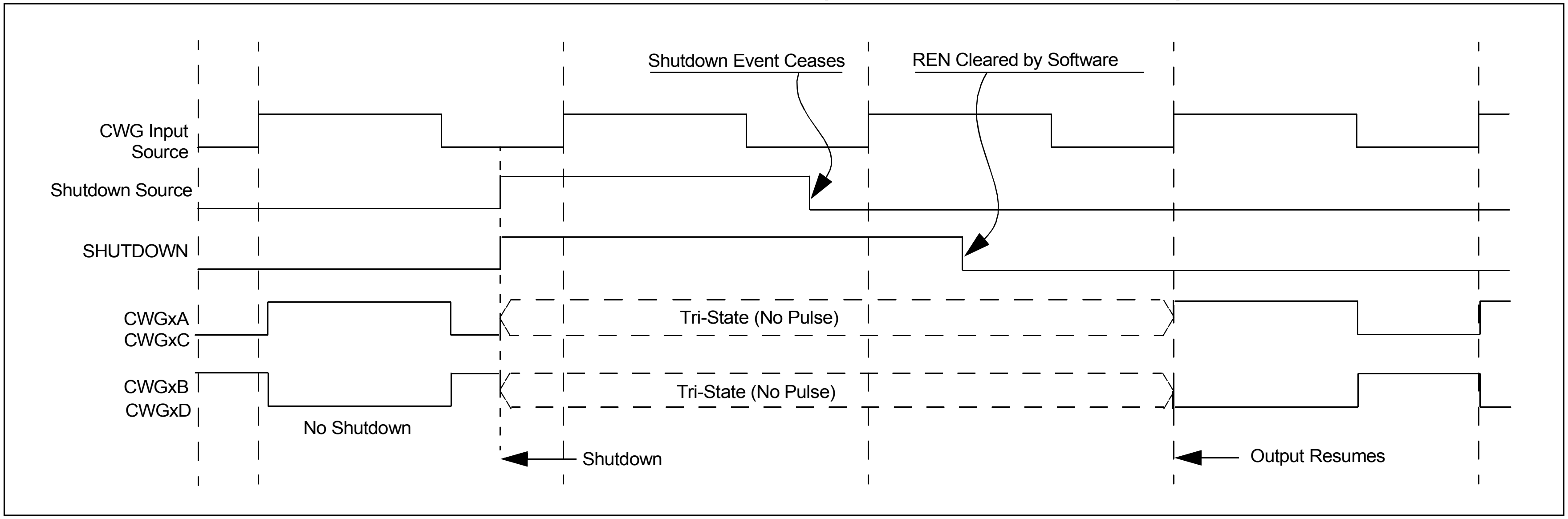
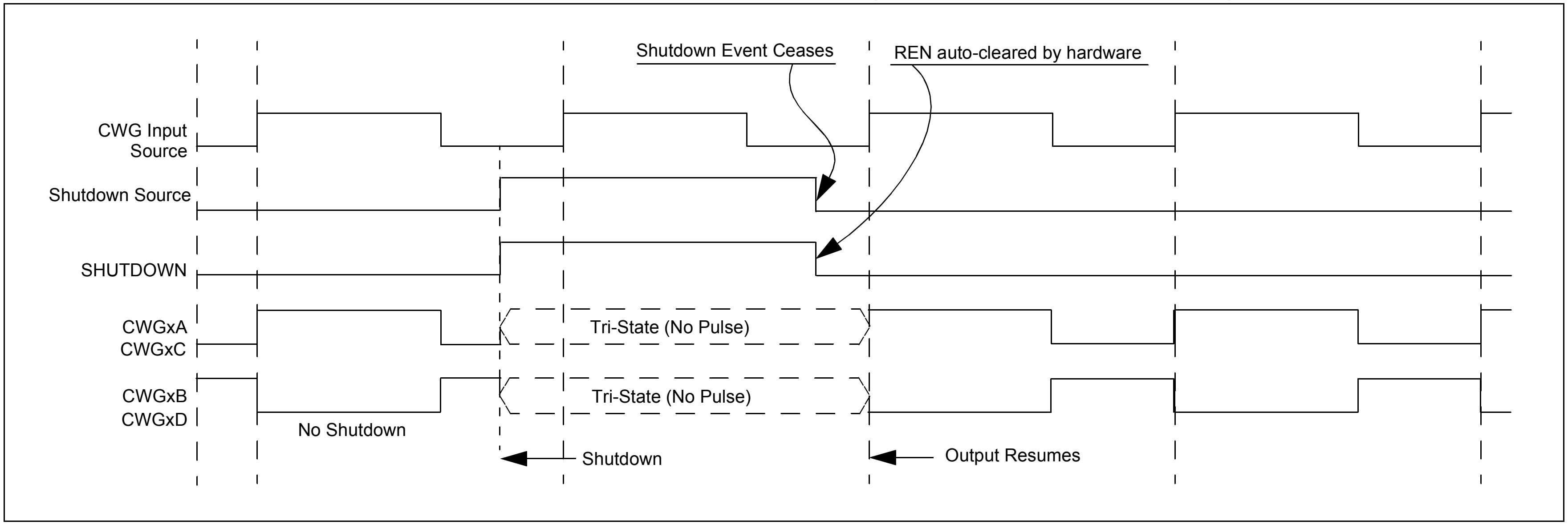


FIGURE 24-14: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (REN = `1`, LSAC = `01`, LSBD = `01`)



24.13 Register Definitions: CWG Control

REGISTER 24-1: CWGxCON0: CWGx CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD[(1)]	—	—	—	MODE<2:0>		
bit 7							bit 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 **EN:** CWGx Enable bit
`1` = Module is enabled
`0` = Module is disabled

bit 6 **LD:** CWGx Load Buffer bits(1)
`1` = Buffers to be loaded on the next rising/falling event
`0` = Buffers not loaded

bit 5-3 **Unimplemented**: Read as ‘`0`’

bit 2-0 **MODE<2:0>:** CWGx Mode bits
`111` = Reserved
`110` = Reserved
`101` = CWG outputs operate in Push-Pull mode
`100` = CWG outputs operate in Half-Bridge mode
`011` = CWG outputs operate in Reverse Full-Bridge mode
`010` = CWG outputs operate in Forward Full-Bridge mode
`001` = CWG outputs operate in Synchronous Steering mode
`000` = CWG outputs operate in Steering mode

Note 1: This bit can only be set after EN = `1` and cannot be set in the same instruction that EN is set.

REGISTER 24-2: CWGxCON1: CWGx CONTROL REGISTER 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable 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	**IN:** CWG Input Value
bit 4	**Unimplemented:** Read as ‘`0`’
bit 3	**POLD:** CWGxD Output Polarity bit `1` = Signal output is inverted polarity `0` = Signal output is normal polarity
bit 2	**POLC:** CWGxC Output Polarity bit `1` = Signal output is inverted polarity `0` = Signal output is normal polarity
bit 1	**POLB:** CWGxB Output Polarity bit `1` = Signal output is inverted polarity `0` = Signal output is normal polarity
bit 0	**POLA:** CWGxA Output Polarity bit `1` = Signal output is inverted polarity `0` = Signal output is normal polarity

REGISTER 24-3: CWGxDBR: CWGx RISING DEAD-BAND COUNTER REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	DBR<5:0>					
bit 7							bit 0

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

bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**DBR<5:0>:** Rising Event Dead-Band Value for Counter bits

REGISTER 24-4: CWGxDBF: CWGx FALLING DEAD-BAND COUNTER REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	DBF<5:0>					
bit 7							bit 0

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

bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**DBF<5:0>:** Falling Event Dead-Band Value for Counter bits

REGISTER 24-5: CWGxAS0: CWGx AUTO-SHUTDOWN CONTROL REGISTER 0

R/W/HS-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0
SHUTDOWN[1, 2]	REN	LSBD<1:0>		LSAC<1:0>		—	—
bit 7							bit 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 **SHUTDOWN:** Auto-Shutdown Event Status bit[1, 2]

`1` = An Auto-Shutdown state is in effect
`0` = No Auto-shutdown event has occurred

bit 6 **REN:** Auto-Restart Enable bit

`1` = Auto-restart enabled
`0` = Auto-restart disabled

bit 5-4 **LSBD<1:0>:** CWGxB and CWGxD Auto-Shutdown State Control bits

`11` = A logic ‘`1`’ is placed on CWGxB/D when an auto-shutdown event is present
`10` = A logic ‘`0`’ is placed on CWGxB/D when an auto-shutdown event is present
`01` = Pin is tri-stated on CWGxB/D when an auto-shutdown event is present
`00` = The inactive state of the pin, including polarity, is placed on CWGxB/D after the required dead-band interval

bit 3-2 **LSAC<1:0>:** CWGxA and CWGxC Auto-Shutdown State Control bits

`11` = A logic ‘`1`’ is placed on CWGxA/C when an auto-shutdown event is present
`10` = A logic ‘`0`’ is placed on CWGxA/C when an auto-shutdown event is present
`01` = Pin is tri-stated on CWGxA/C when an auto-shutdown event is present
`00` = The inactive state of the pin, including polarity, is placed on CWGxA/C after the required dead-band interval

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

Note 1: This bit may be written while EN = `0` (CWGxCON0 register) to place the outputs into the shutdown configuration.

2: The outputs will remain in auto-shutdown state until the next rising edge of the input signal after this bit is cleared.

REGISTER 24-6: CWGxAS1: CWGx AUTO-SHUTDOWN CONTROL REGISTER 1

U-1	R/W-0/0	R/W-0/0	R/W-0/0	U-1	R/W-0/0	R/W-0/0	R/W-0/0
—	TMR6AS	TMR4AS	TMR2AS	—	C2AS(1)	C1AS	INAS
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘1’

bit 6 **TMR6AS:** TMR6 Postscale Output bit
1 = TMR6 postscale shut-down is enabled
0 = TMR6 postscale shut-down is disabled

bit 5 **TMR4AS:** TMR4 Postscale Output bit
1 = TMR4 postscale shut-down is enabled
0 = TMR4 postscale shut-down is disabled

bit 4 **TMR2AS:** TMR2 Postscale Output bit
1 = TMR2 postscale shut-down is enabled
0 = TMR2 postscale shut-down is disabled

bit 3 **Unimplemented:** Read as ‘1’

bit 2 **C2AS:** Comparator C2 Output bit(1)
1 = C2 output shut-down is enabled
0 = C2 output shut-down is disabled

bit 1 **C1AS:** Comparator C1 Output bit
1 = C1 output shut-down is enabled
0 = C1 output shut-down is disabled

bit 0 **INAS:** CWGx Input Pin bit
1 = CWGxIN input pin shut-down is enabled
0 = CWGxIN input pin shut-down is disabled

Note 1: PIC16(L)F1613 only.

REGISTER 24-7: CWGxOCON0: CWGx STEERING CONTROL REGISTER 0[(1)]

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OVRD	OVRC	OVRB	OVRA	STRD(2)	STRC(2)	STRB(2)	STRA(2)
bit 7							bit 0

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

bit 7 **OVRD:** Steering Data D bit

bit 6 **OVRC:** Steering Data C bit

bit 5 **OVRB:** Steering Data B bit

bit 4 **OVRA:** Steering Data A bit

bit 3 **STRD:** Steering Enable D bit(2)

`1` = CWGxD output has the CWGx_data waveform with polarity control from POLD bit
`0` = CWGxD output is assigned the value of OVRD bit

bit 2 **STRC:** Steering Enable C bit(2)

`1` = CWGxC output has the CWGx_data waveform with polarity control from POLC bit
`0` = CWGxC output is assigned the value of OVRC bit

bit 1 **STRB:** Steering Enable B bit(2)

`1` = CWGxB output has the CWGx_data waveform with polarity control from POLB bit
`0` = CWGxB output is assigned the value of OVRB bit

bit 0 **STRA:** Steering Enable A bit(2)

`1` = CWGxA output has the CWGx_data waveform with polarity control from POLA bit
`0` = CWGxA output is assigned the value of OVRA bit

Note 1: The bits in this register apply only when MODE<2:0> = `00x`.

2: This bit is effectively double-buffered when MODE<2:0> = `001`.

REGISTER 24-8: CWGxOCON1: CWGx OUTPUT ENABLE REGISTER 1

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OED	OEC	OEB	OEA
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as ‘`0`’

bit 3 **OED:** CWGx D Output Pin Enable bit
`1` = CWGx D output pin is enabled
`0` = CWGx D output pin is disabled

bit 2 **OEC:** CWGx C Output Pin Enable bit
`1` = CWGx C output pin is enabled
`0` = CWGx C output pin is disabled

bit 1 **OEB:** CWGx B Output Pin Enable bit
`1` = CWGx B output pin is enabled
`0` = CWGx B output pin is disabled

bit 0 **OEA:** CWGx A Output Pin Enable bit
`1` = CWGx A output pin is enabled
`0` = CWGx A output pin is disabled

REGISTER 24-9: CWGxCLKCON: CWGx CLOCK SELECTION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	CS
bit 7							bit 0

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

bit 7-1 **Unimplemented:** Read as ‘0’

bit 0 **CS:** CWGx Clock Selection bit
1 = HFINTOSC 16 MHz is selected
0 = FOSC is selected

REGISTER 24-10: CWGxISM: CWGx INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	IS<2:0>		
bit 7							bit 0

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

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **GxIS<2:0>:** CWGx Input Selection bits
111 = Reserved, do not use
110 = Reserved, do not use
101 = Reserved, do not use
100 = CCP2_out
011 = CCP1_out
010 = C2_OUT_sync[1]
001 = C1_OUT_sync
000 = CWGxIN pin

Note 1: PIC16(L)F1613 only.

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
APFCON	—	CWGASEL[(2)]	CWGBSEL[(2)]	—	T1GSEL	—	CCP2SEL[(2)]	CCP1SEL[(1)]	132
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	257
CWG1AS1	—	TMR6AS	TMR4AS	TMR2AS	—	C2AS	C1AS	INAS	258
CWG1CLKCON	—	—	—	—	—	—	—	CS	261
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			259
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	255
CWG1DBF	—	—	DBF<5:0>						256
CWG1DBR	—	—	DBR<5:0>						256
CWG1ISM	—	—	—	—	—	IS<2:0>			261
CWG1OCON0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	259
CWG1OCON1	—	—	—	—	OED	OEC	OEB	OEA	260

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

Note 1: PIC12(L)F1612 only.

2: PIC16(L)F1613 only.

25.0 SIGNAL MEASUREMENT TIMER (SMT)

The SMT is a 24-bit counter with advanced clock and gating logic, which can be configured for measuring a variety of digital signal parameters such as pulse width, frequency and duty cycle, and the time difference between edges on two signals.

Features of the SMT include:

- 24-bit timer/counter
  - Four 8-bit registers (SMTxTMRL/H/U)
  - Readable and writable
  - Optional 16-bit operating mode
- Two 24-bit measurement capture registers
- One 24-bit period match register
- Multi-mode operation, including relative timing measurement
- Interrupt on period match
- Multiple clock, gate and signal sources
- Interrupt on acquisition complete
- Ability to read current input values

Note: These devices implement two SMT modules. All references to SMTx apply to SMT1 and SMT2.

FIGURE 25-1: SMTx BLOCK DIAGRAM

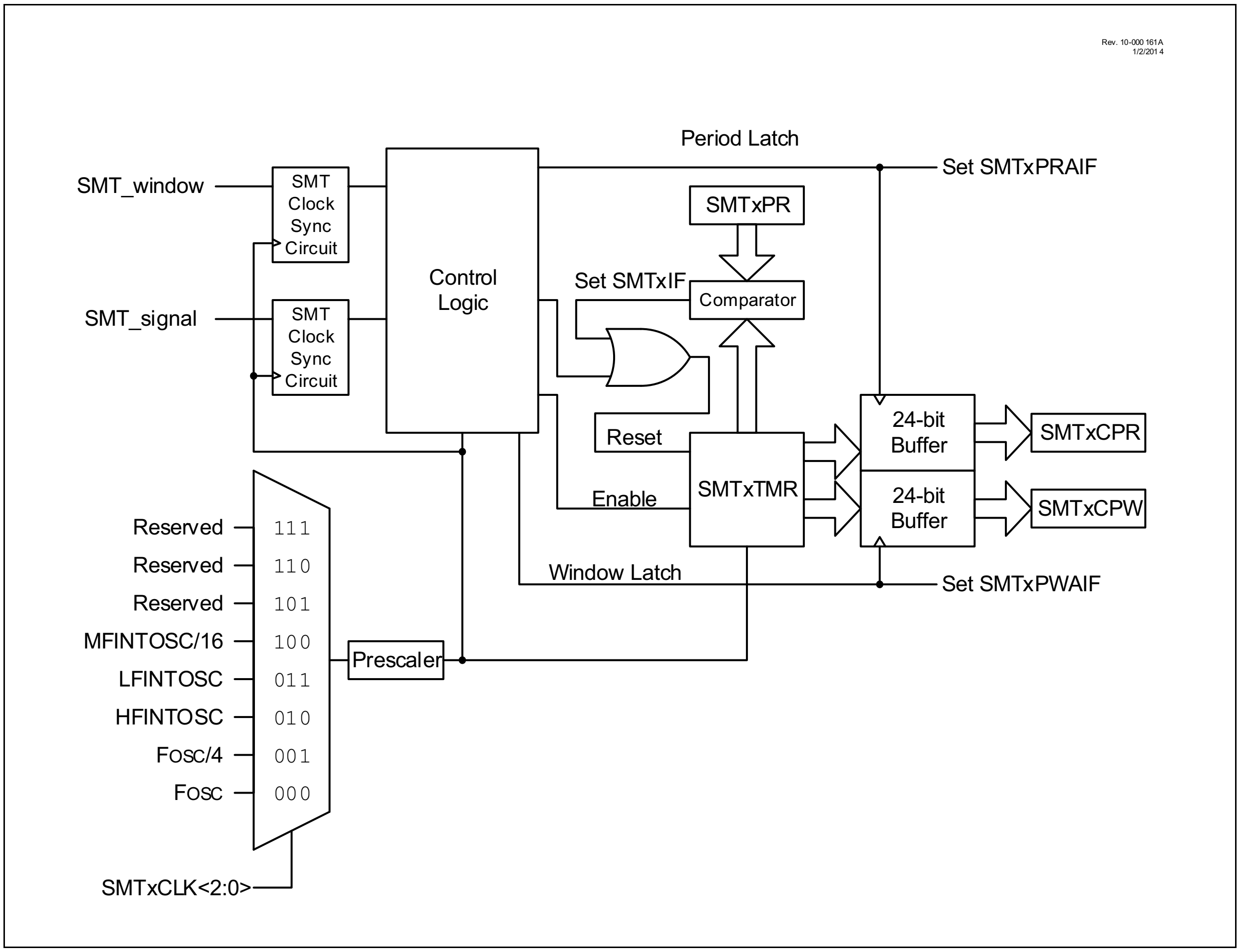
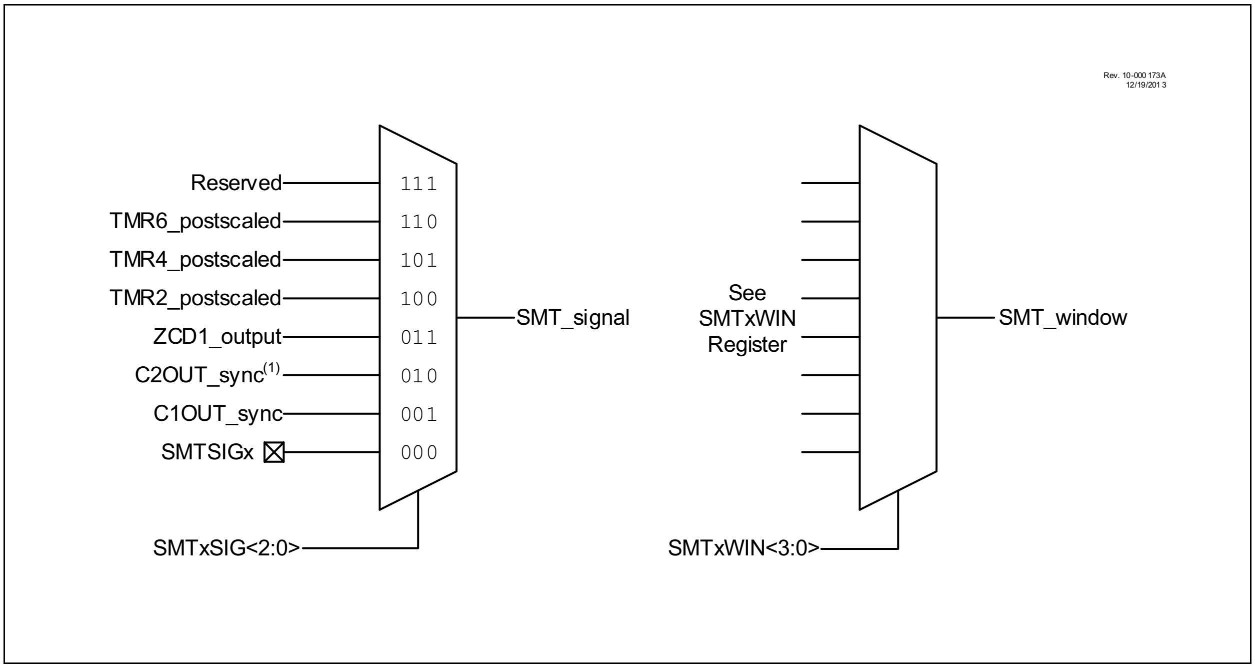


FIGURE 25-2: SMTx SIGNAL AND WINDOW BLOCK DIAGRAM



25.1 SMT Operation

The core of the module is the 24-bit counter, SMTxTMR combined with a complex data acquisition front-end. Depending on the mode of operation selected, the SMT can perform a variety of measurements summarized in Table 25-1.

25.1.1 CLOCK SOURCES

Clock sources available to the SMT include:

- FOSC
- FOSC/4
- HFINTOSC 16 MHz
- LFINTOSC
- MFINTOSC 31.25 kHz

The SMT clock source is selected by configuring the CSEL<2:0> bits in the SMTxCLK register. The clock source can also be prescaled using the PS<1:0> bits of the SMTxCON0 register. The prescaled clock source is used to clock both the counter and any synchronization logic used by the module.

25.1.2 PERIOD MATCH INTERRUPT

Similar to other timers, the SMT triggers an interrupt when SMTxTMR rolls over to ‘`0`’. This happens when SMTxTMR = SMTxPR, regardless of mode. Hence, in any mode that relies on an external signal or a window to reset the timer, proper operation requires that SMTxPR be set to a period larger than that of the expected signal or window.

25.2 Basic Timer Function Registers

The SMTxTMR time base and the SMTxCPW/SMTxPR/SMTxCPR buffer registers serve several functions and can be manually updated using software.

25.2.1 TIME BASE

The SMTxTMR is the 24-bit counter that is the center of the SMT. It is used as the basic counter/timer for measurement in each of the modes of the SMT. It can be reset to a value of 24'h00_0000 by setting the RST bit of the SMTxSTAT register. It can be written to and read from software, but it is not guarded for atomic access, therefore reads and writes to the SMTxTMR should only be made when the GO = `0`, or the software should have other measures to ensure integrity of SMTxTMR reads/writes.

25.2.2 PULSE WIDTH LATCH REGISTERS

The SMTxCPW registers are the 24-bit SMT pulse width latch. They are used to latch in the value of the SMTxTMR when triggered by various signals, which are determined by the mode the SMT is currently in. The SMTxCPW registers can also be updated with the current value of the SMTxTMR value by setting the CPWUP bit of the SMTxSTAT register.

25.2.3 PERIOD LATCH REGISTERS

The SMTxCPR registers are the 24-bit SMT period latch. They are used to latch in other values of the SMTxTMR when triggered by various other signals, which are determined by the mode the SMT is currently in.

The SMTxCPR registers can also be updated with the current value of the SMTxTMR value by setting the CPRUP bit in the SMTxSTAT register.

25.3 Halt Operation

The counter can be prevented from rolling-over using the STP bit in the SMTxCON0 register. When halting is enabled, the period match interrupt persists until the SMTxTMR is reset (either by a manual reset, **Section25.2.1 “Time Base”**) or by clearing the SMTxGO bit of the SMTxCON1 register and writing the SMTxTMR values in software.

25.4 Polarity Control

The three input signals for the SMT have polarity control to determine whether or not they are active high/positive edge or active low/negative edge signals.

The following bits apply to Polarity Control:

- WSEL bit (Window Polarity)
- SSEL bit (Signal Polarity)
- CSEL bit (Clock Polarity)

These bits are located in the SMTxCON0 register.

25.5 Status Information

The SMT provides input status information for the user without requiring the need to deal with the polarity of the incoming signals.

25.5.1 WINDOW STATUS

Window status is determined by the WS bit of the SMTxSTAT register. This bit is only used in Windowed Measure, Gated Counter and Gated Window Measure modes, and is only valid when TS = `1`, and will be delayed in time by synchronizer delays in non-Counter modes.

25.5.2 SIGNAL STATUS

Signal status is determined by the AS bit of the SMTxSTAT register. This bit is used in all modes except Window Measure, Time of Flight and Capture modes, and is only valid when TS = `1`, and will be delayed in time by synchronizer delays in non-Counter modes.

25.5.3 GO STATUS

Timer run status is determined by the TS bit of the SMTxSTAT register, and will be delayed in time by synchronizer delays in non-Counter modes.

25.6 Modes of Operation

The modes of operation are summarized in Table 25-1. The following sections provide detailed descriptions, examples of how the modes can be used. Note that all waveforms assume WPOL/SPOL/CPOL = `0`. When WPOL/SPOL/CPOL = `1`, all SMTSIGx, SMTWINx and SMT clock signals will have a polarity opposite to that indicated. For all modes, the REPEAT bit controls whether the acquisition is repeated or single. When REPEAT = `0` (Single Acquisition mode), the timer will stop incrementing and the SMTxGO bit will be reset upon the completion of an acquisition. Otherwise, the timer will continue and allow for continued acquisitions to overwrite the previous ones until the timer is stopped in software.

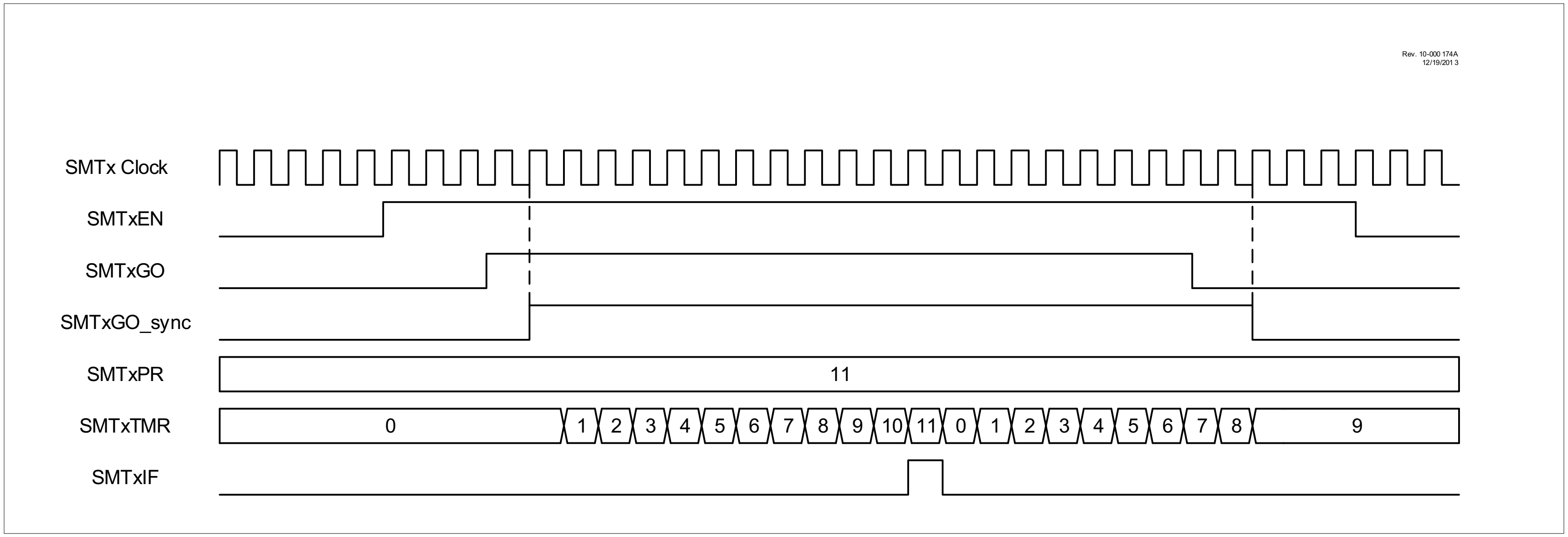
25.6.1 TIMER MODE

Timer mode is the simplest mode of operation where the SMTxTMR is used as a 16/24-bit timer. No data acquisition takes place in this mode. The timer increments as long as the SMTxGO bit has been set by software. No SMT window or SMT signal events affect the SMTxGO bit. Everything is synchronized to the SMT clock source. When the timer experiences a period match (SMTxTMR = SMTxPR), SMTxTMR is reset and the period match interrupt trips. See Figure 25-3.

TABLE 25-1: MODES OF OPERATION

MODE	Mode of Operation	Synchronous Operation	Reference
`0000`	Timer	Yes	**Section25.6.1 “Timer Mode”**
`0001`	Gated Timer	Yes	**Section25.6.2 “Gated Timer Mode”**
`0010`	Period and Duty Cycle Acquisition	Yes	**Section25.6.3 “Period and Duty-Cycle Mode”**
`0011`	High and Low Time Measurement	Yes	**Section25.6.4 “High and Low Measure Mode”**
`0100`	Windowed Measurement	Yes	**Section25.6.5 “Windowed Measure Mode”**
`0101`	Gated Windowed Measurement	Yes	**Section25.6.6 “Gated Window Measure Mode”**
`0110`	Time of Flight	Yes	**Section25.6.7 “Time of Flight Measure Mode”**
`0111`	Capture	Yes	**Section25.6.8 “Capture Mode”**
`1000`	Counter	No	**Section25.6.9 “Counter Mode”**
`1001`	Gated Counter	No	**Section25.6.10 “Gated Counter Mode”**
`1010`	Windowed Counter	No	**Section25.6.11 “Windowed Counter Mode”**
`1011 - 1111`	Reserved	—	—

FIGURE 25-3: TIMER MODE TIMING DIAGRAM



25.6.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMTxTMR will increment. Upon a falling edge of the external signal, the SMTxCPW register will update to the current value of the SMTxTMR. Example waveforms for both repeated and single acquisitions are provided in Figure 25-4 and Figure 25-5.

FIGURE 25-4: GATED TIMER MODE REPEAT ACQUISITION TIMING DIAGRAM

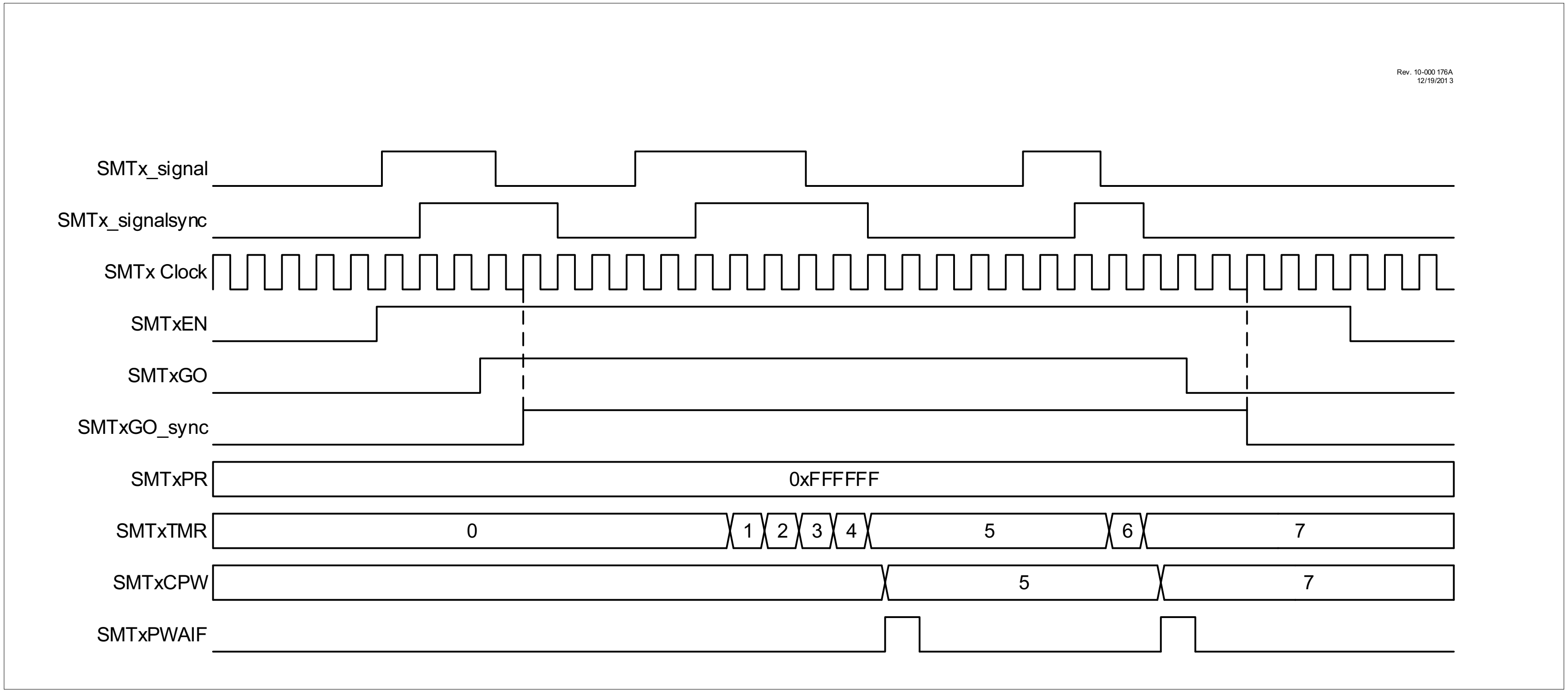
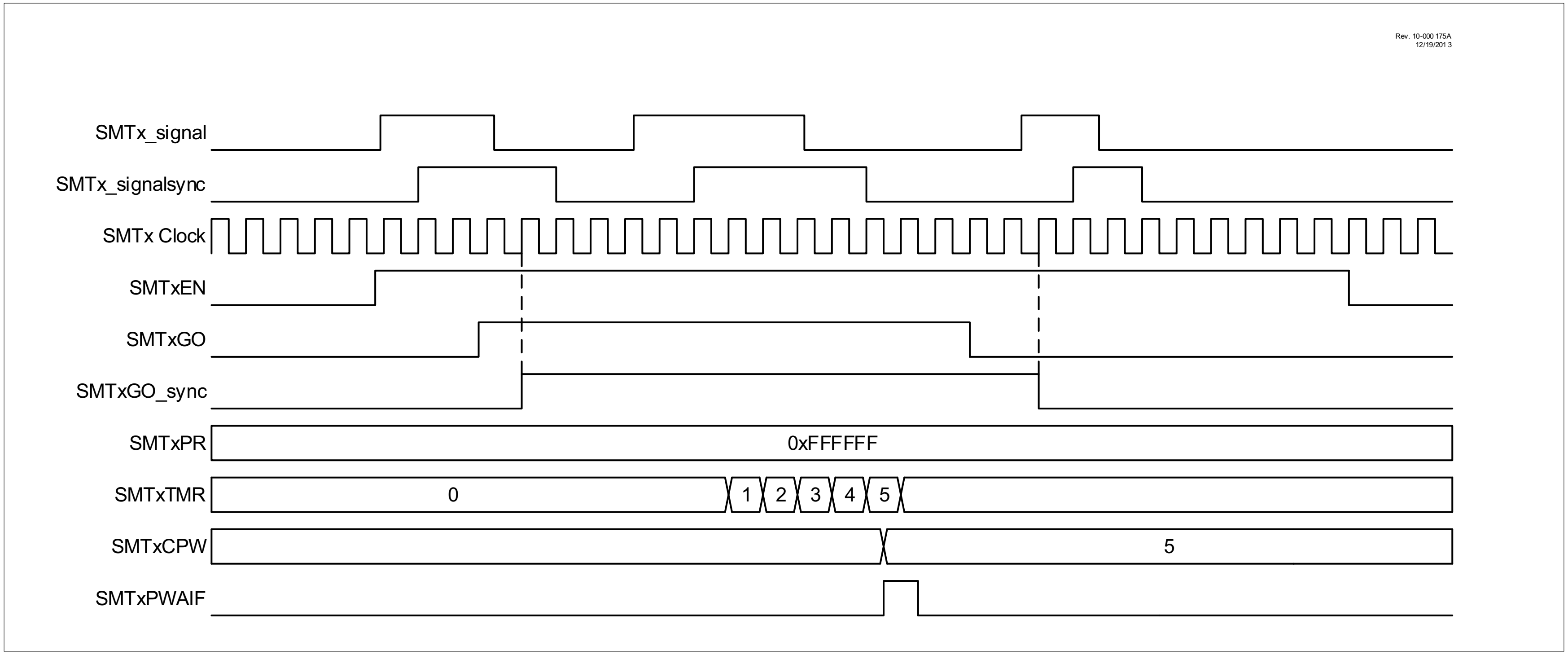


FIGURE 25-5: GATED TIMER MODE SINGLE ACQUISITION TIMING DIAGRAM



25.6.3 PERIOD AND DUTY-CYCLE MODE

In Duty-Cycle mode, either the duty cycle or period (depending on polarity) of the SMTx_signal can be acquired relative to the SMT clock. The CPW register is updated on a falling edge of the signal, and the CPR register is updated on a rising edge of the signal, along with the SMTxTMR resetting to 0x0001. In addition, the SMTxGO bit is reset on a rising edge when the SMT is in Single Acquisition mode. See Figure 25-6 and Figure 25-7.

FIGURE 25-6: PERIOD AND DUTY-CYCLE REPEAT ACQUISITION MODE TIMING DIAGRAM

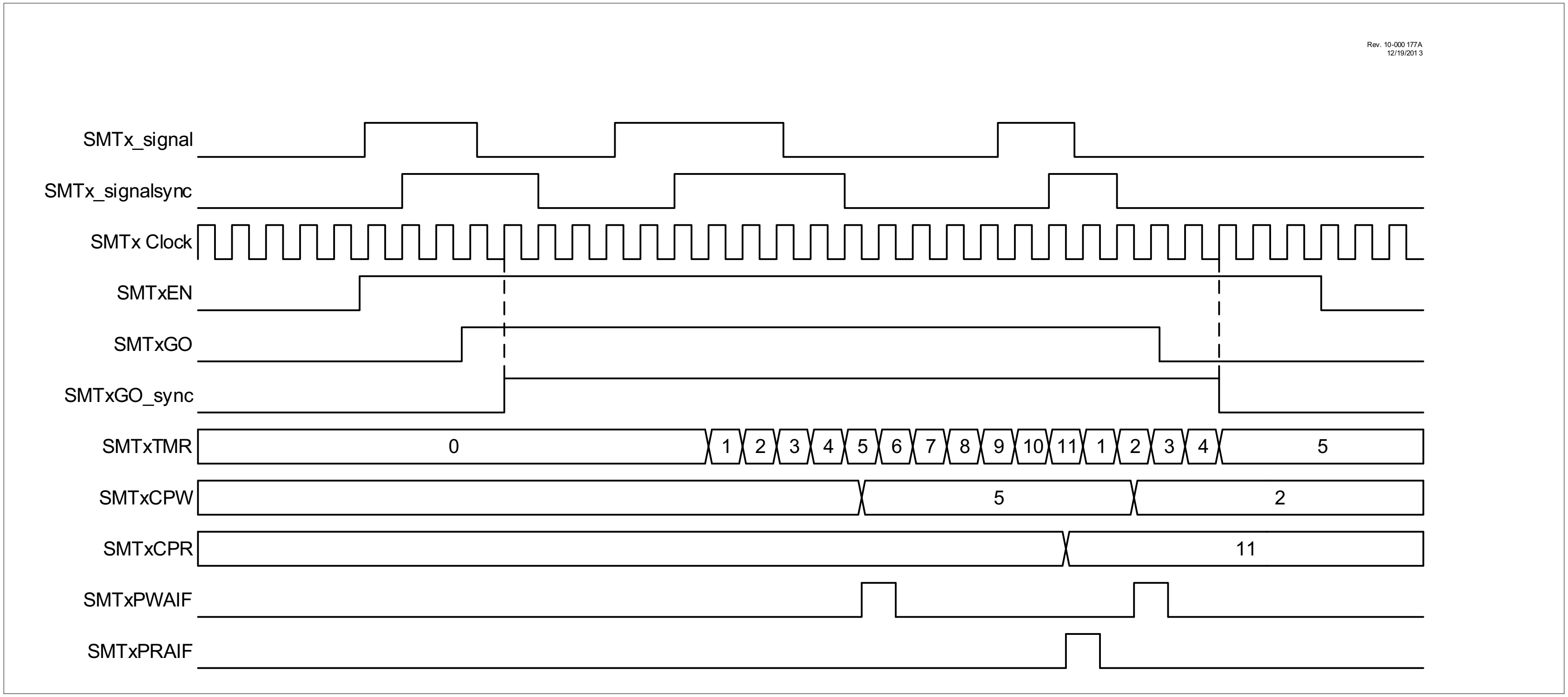
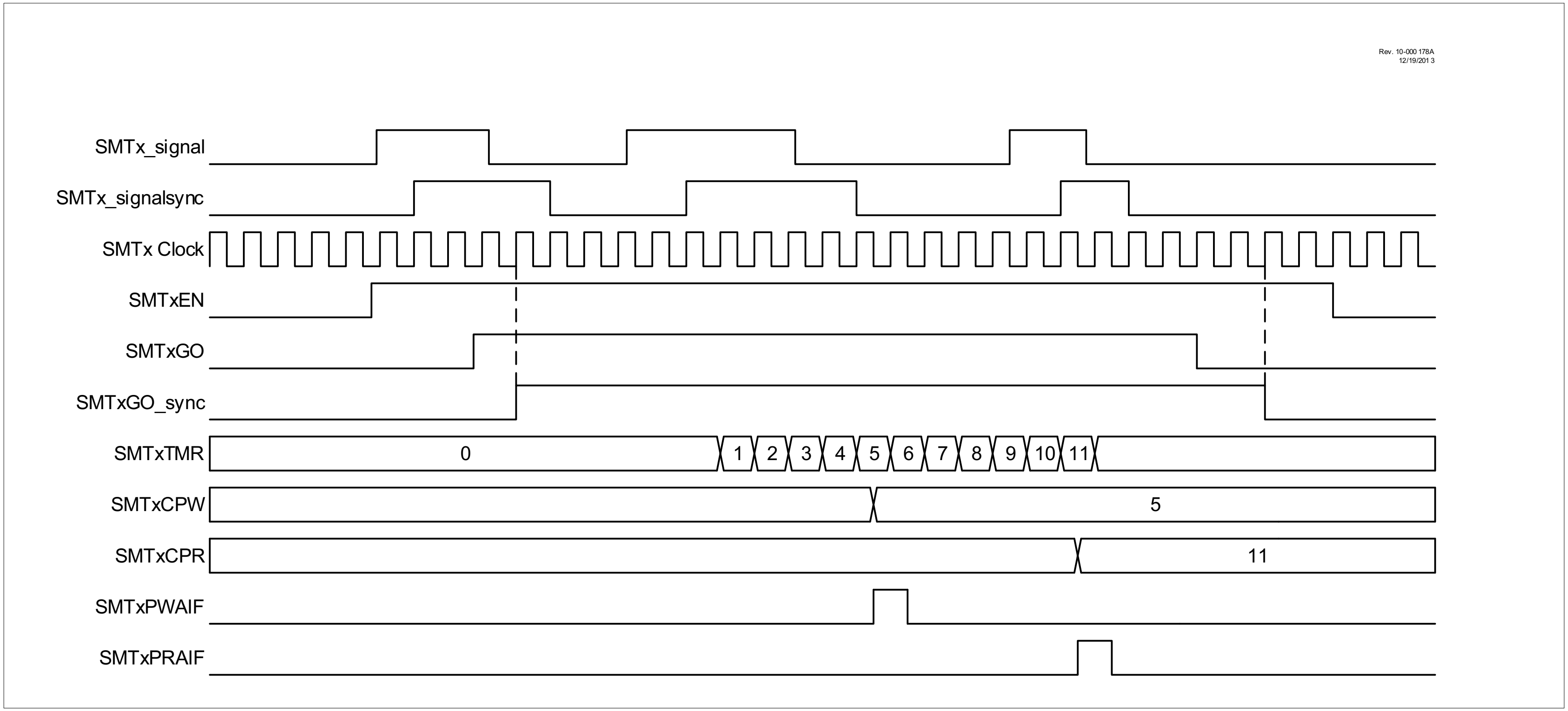


FIGURE 25-7: PERIOD AND DUTY-CYCLE SINGLE ACQUISITION TIMING DIAGRAM



25.6.4 HIGH AND LOW MEASURE MODE

This mode measures the high and low pulse time of the SMTSIGx relative to the SMT clock. It begins incrementing the SMTxTMR on a rising edge on the SMTSIGx input, then updates the SMTxCPW register with the value and resets the SMTxTMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMTxCPR register with its current value and once again resets the SMTxTMR value and begins incrementing again. See Figure 25-8 and Figure 25-9.

FIGURE 25-8: HIGH AND LOW MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

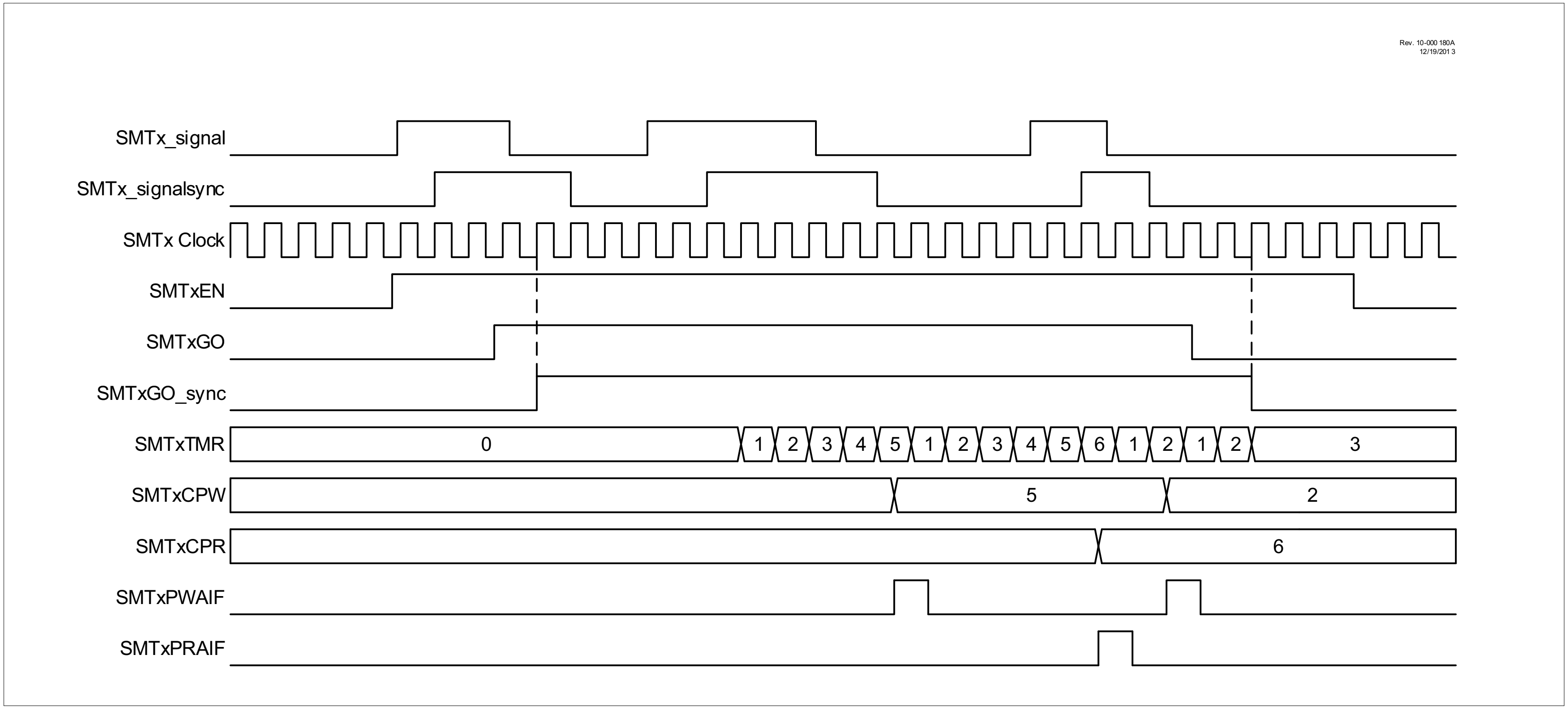
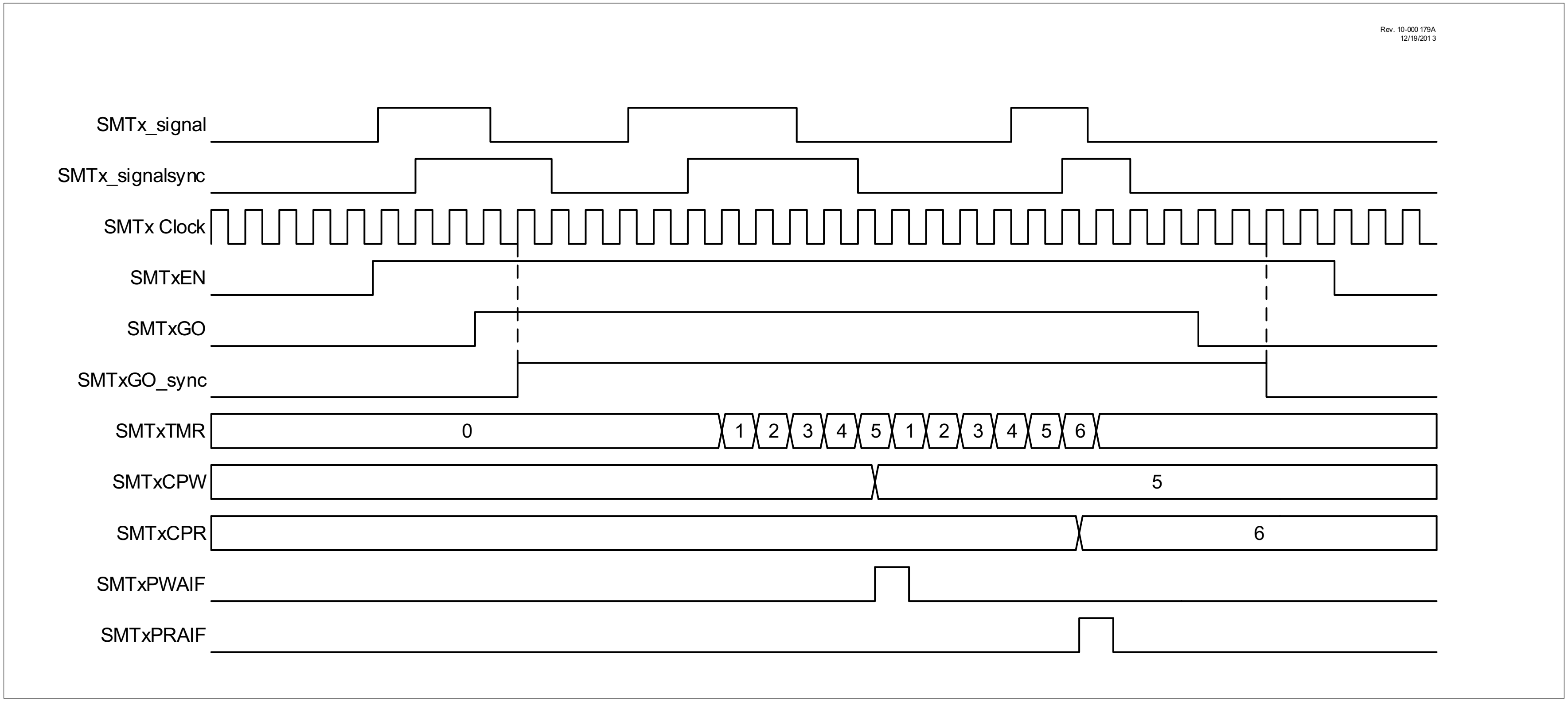


FIGURE 25-9: HIGH AND LOW MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM



25.6.5 WINDOWED MEASURE MODE

This mode measures the window duration of the SMTWINx input of the SMT. It begins incrementing the timer on a rising edge of the SMTWINx input and updates the SMTxCPR register with the value of the timer and resets the timer on a second rising edge. See Figure 25-10 and Figure 25-11.

FIGURE 25-10: WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

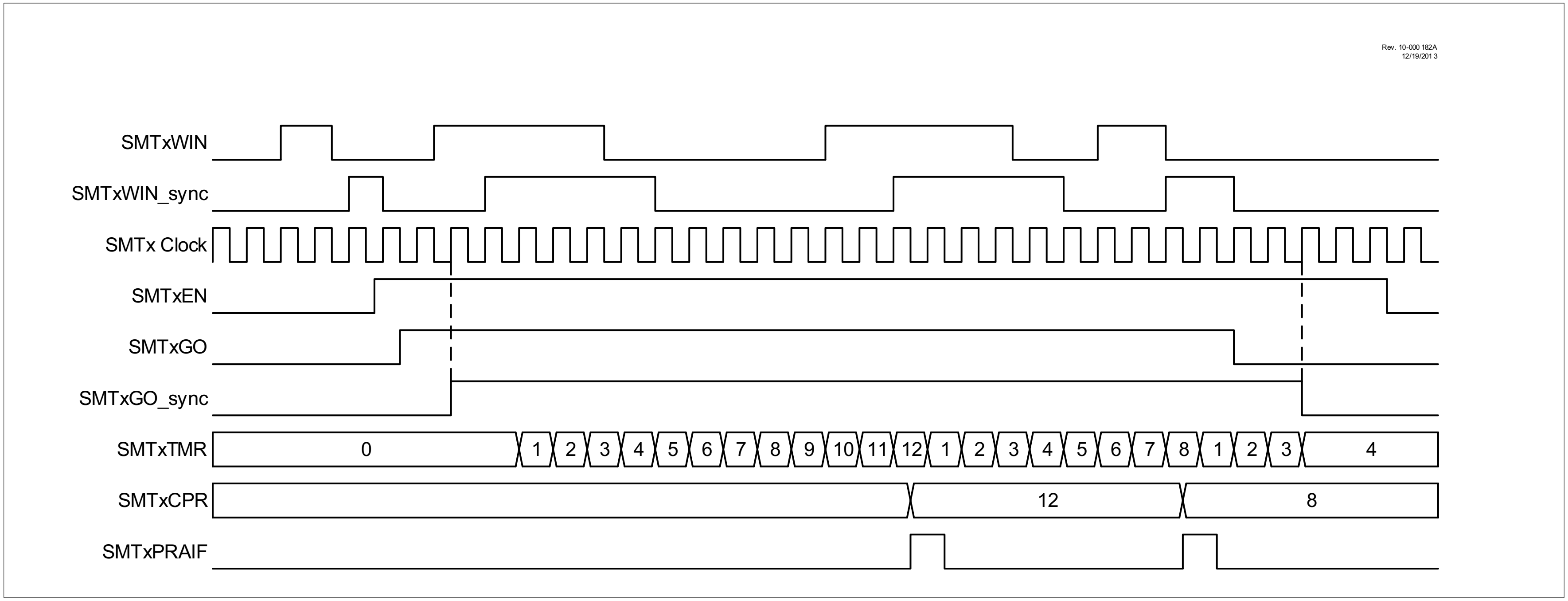
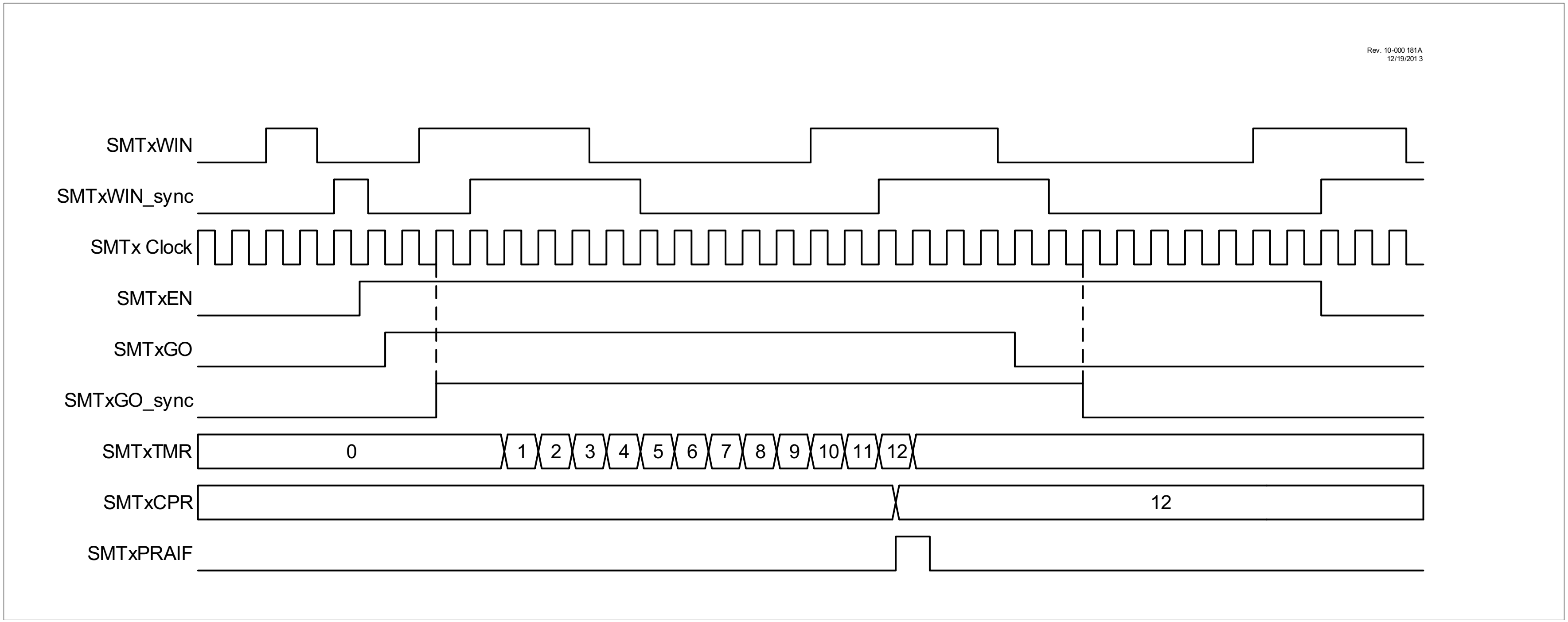


FIGURE 25-11: WINDOWED MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM



25.6.6 GATED WINDOW MEASURE MODE

This mode measures the duty cycle of the SMTx_signal input over a known input window. It does so by incrementing the timer on each pulse of the clock signal while the SMTx_signal input is high, updating the SMTxCPR register and resetting the timer on every rising edge of the SMTWINx input after the first. See Figure 25-12 and Figure 25-13.

FIGURE 25-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

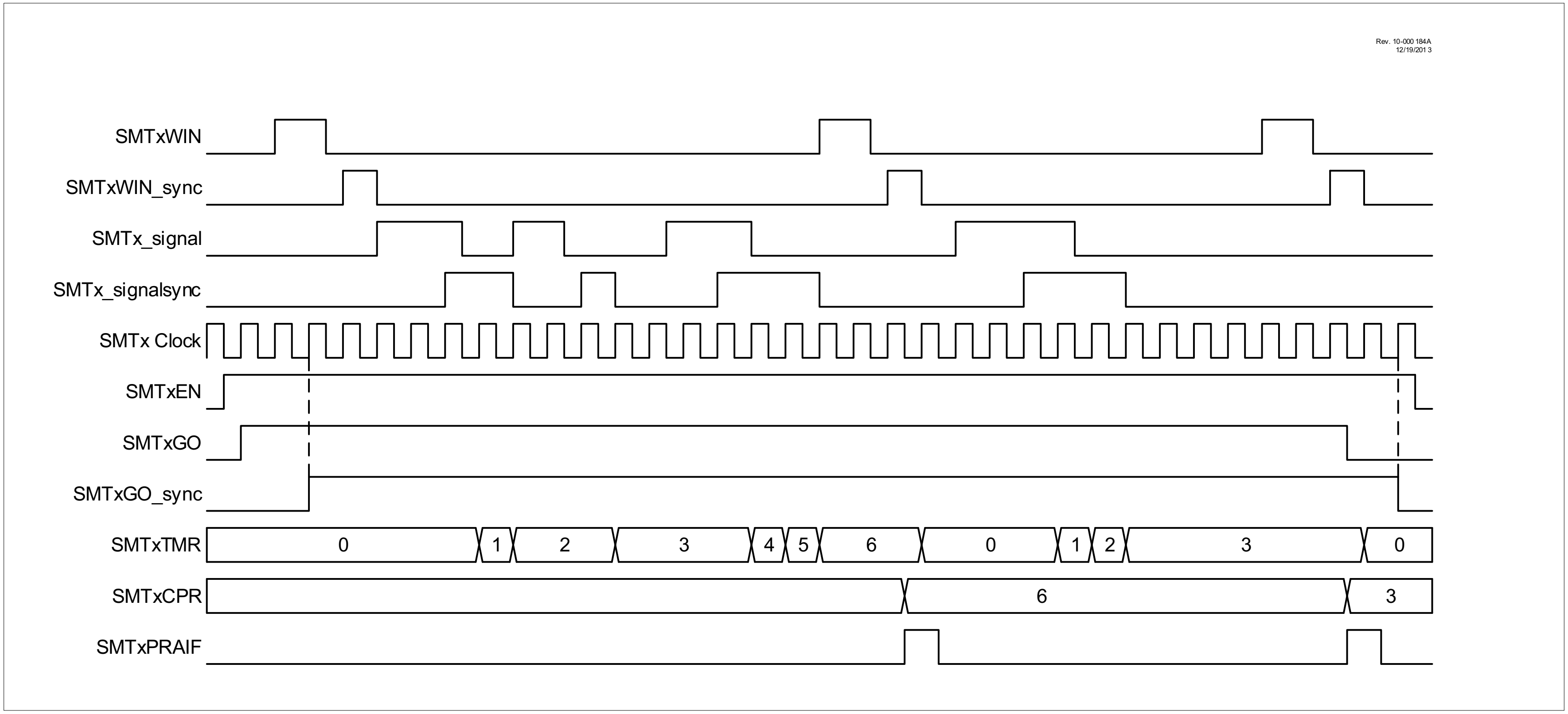
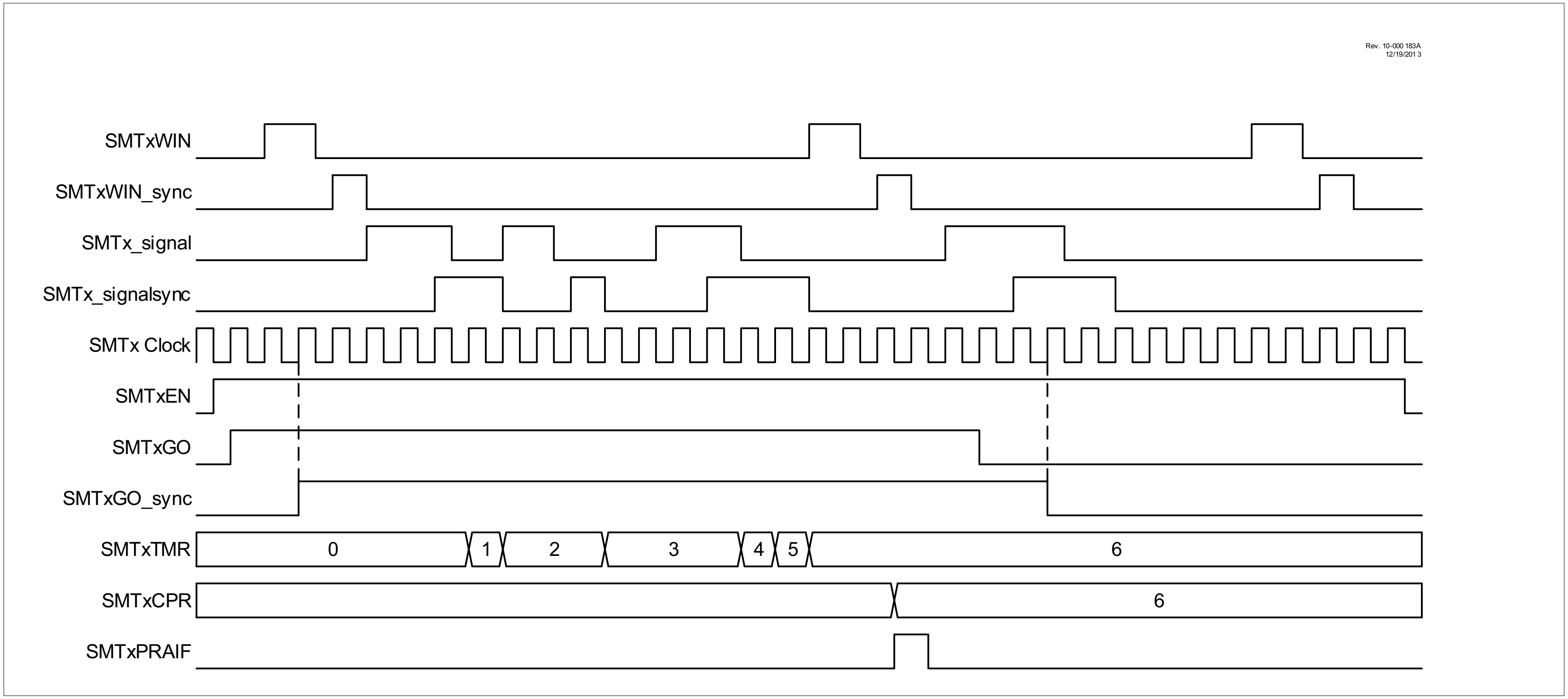


FIGURE 25-13: GATED WINDOWED MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAMS



25.6.7 TIME OF FLIGHT MEASURE MODE

This mode measures the time interval between a rising edge on the SMTWINx input and a rising edge on the SMTx_signal input, beginning to increment the timer upon observing a rising edge on the SMTWINx input, while updating the SMTxCPR register and resetting the timer upon observing a rising edge on the SMTx_signal input. In the event of two SMTWINx rising edges without an SMTx_signal rising edge, it will update the SMTxCPW register with the current value of the timer and reset the timer value. See Figure 25-14 and Figure 25-15.

FIGURE 25-14: TIME OF FLIGHT MODE REPEAT ACQUISITION TIMING DIAGRAM

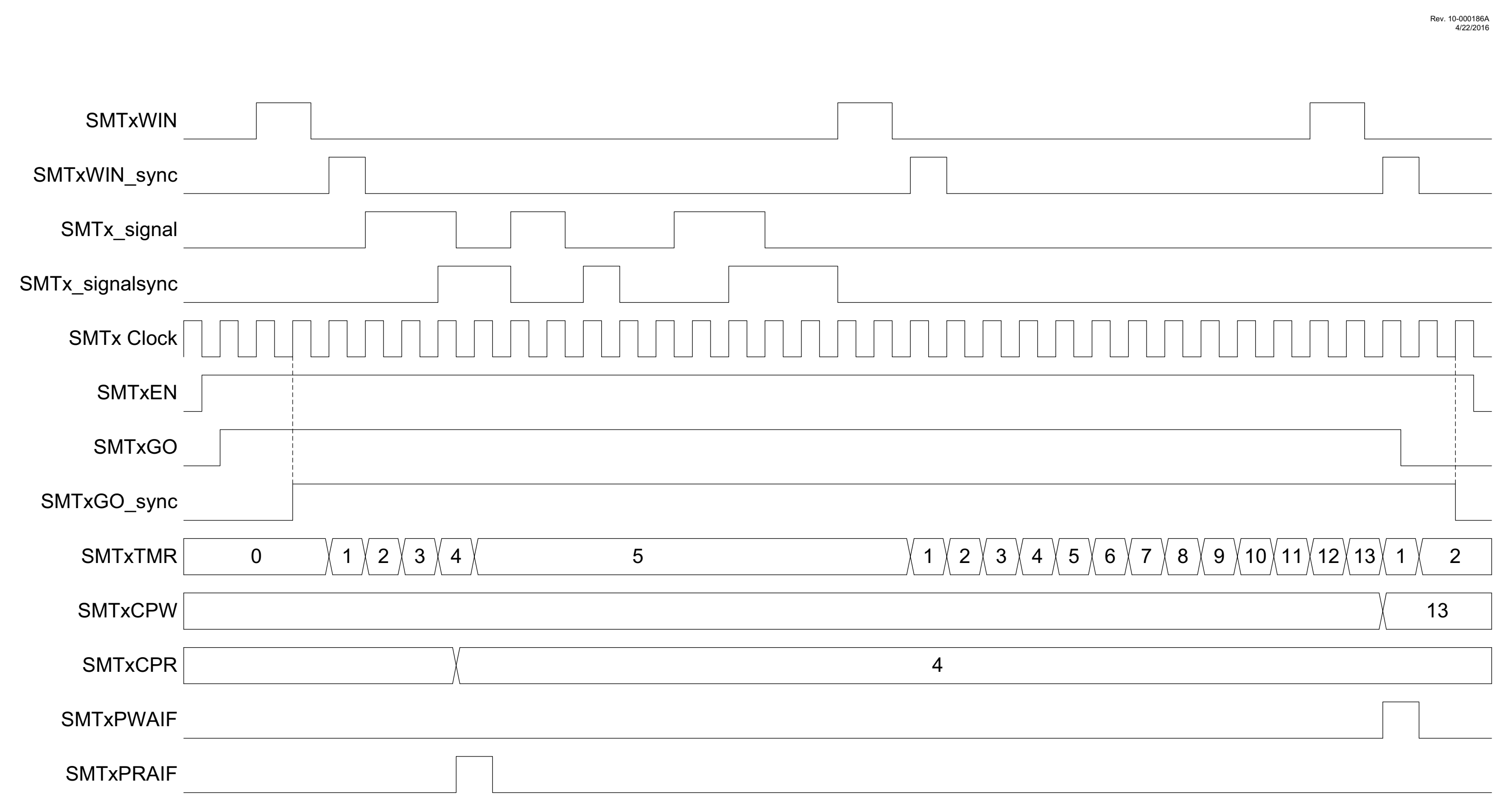
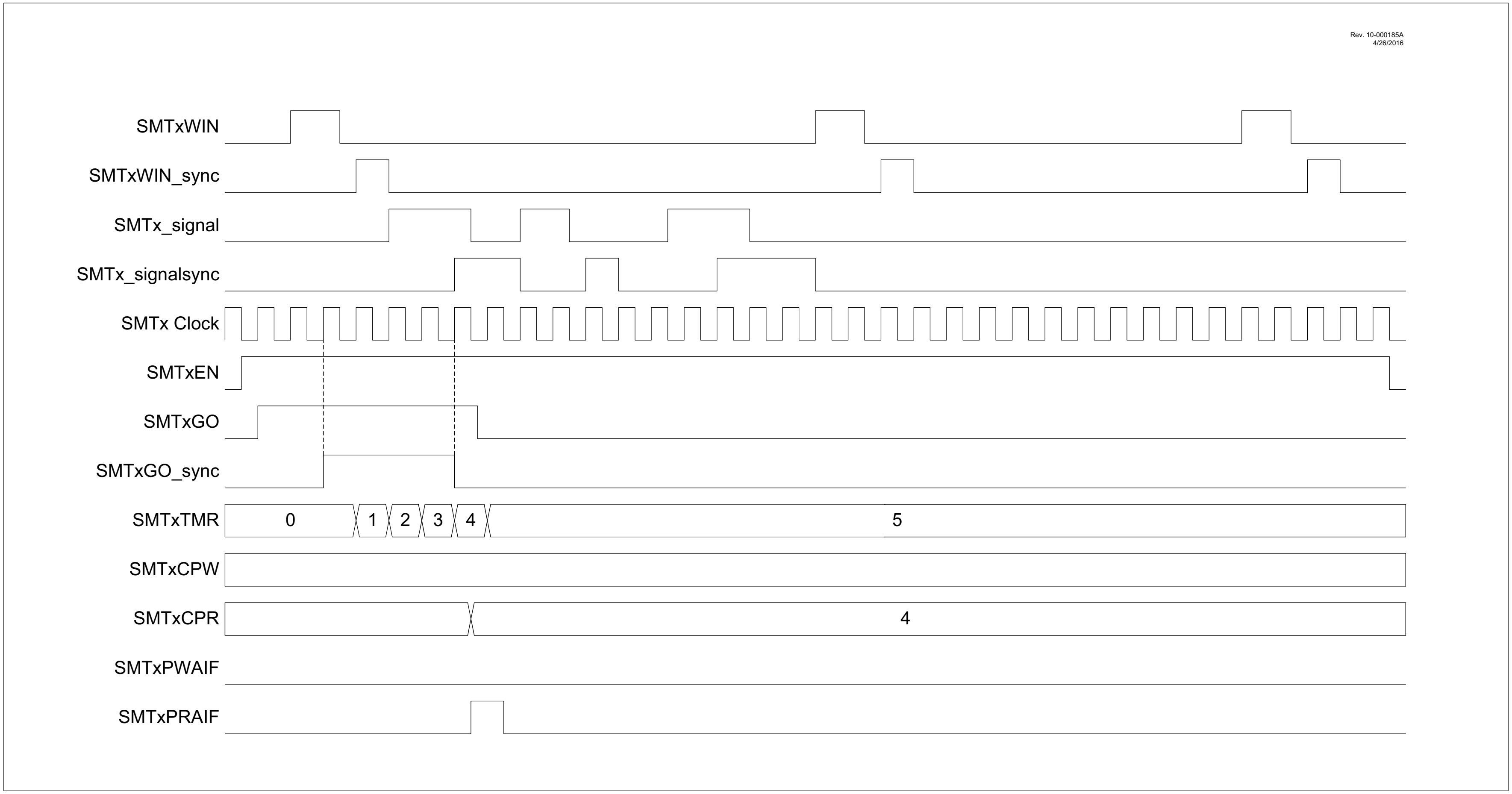


FIGURE 25-15: TIME OF FLIGHT MODE SINGLE ACQUISITION TIMING DIAGRAM



25.6.8 CAPTURE MODE

This mode captures the Timer value based on a rising or falling edge on the SMTWINx input and triggers an interrupt. This mimics the capture feature of a CCP module. The timer begins incrementing upon the SMTxGO bit being set, and updates the value of the SMTxCPR register on each rising edge of SMTWINx, and updates the value of the CPW register on each falling edge of the SMTWINx. The timer is not reset by any hardware conditions in this mode and must be reset by software, if desired. See Figure 25-16 and Figure 25-17.

FIGURE 25-16: CAPTURE MODE REPEAT ACQUISITION TIMING DIAGRAM

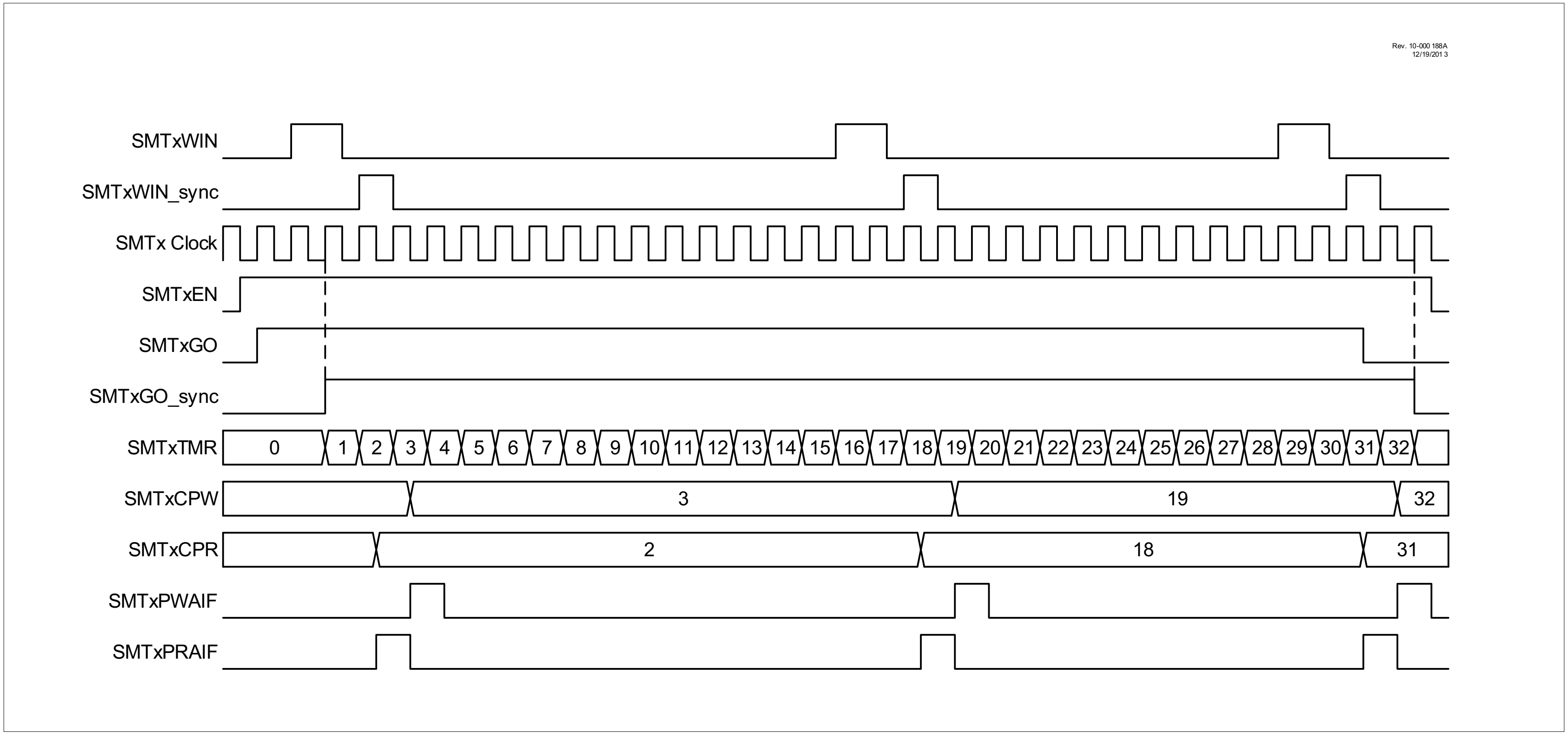
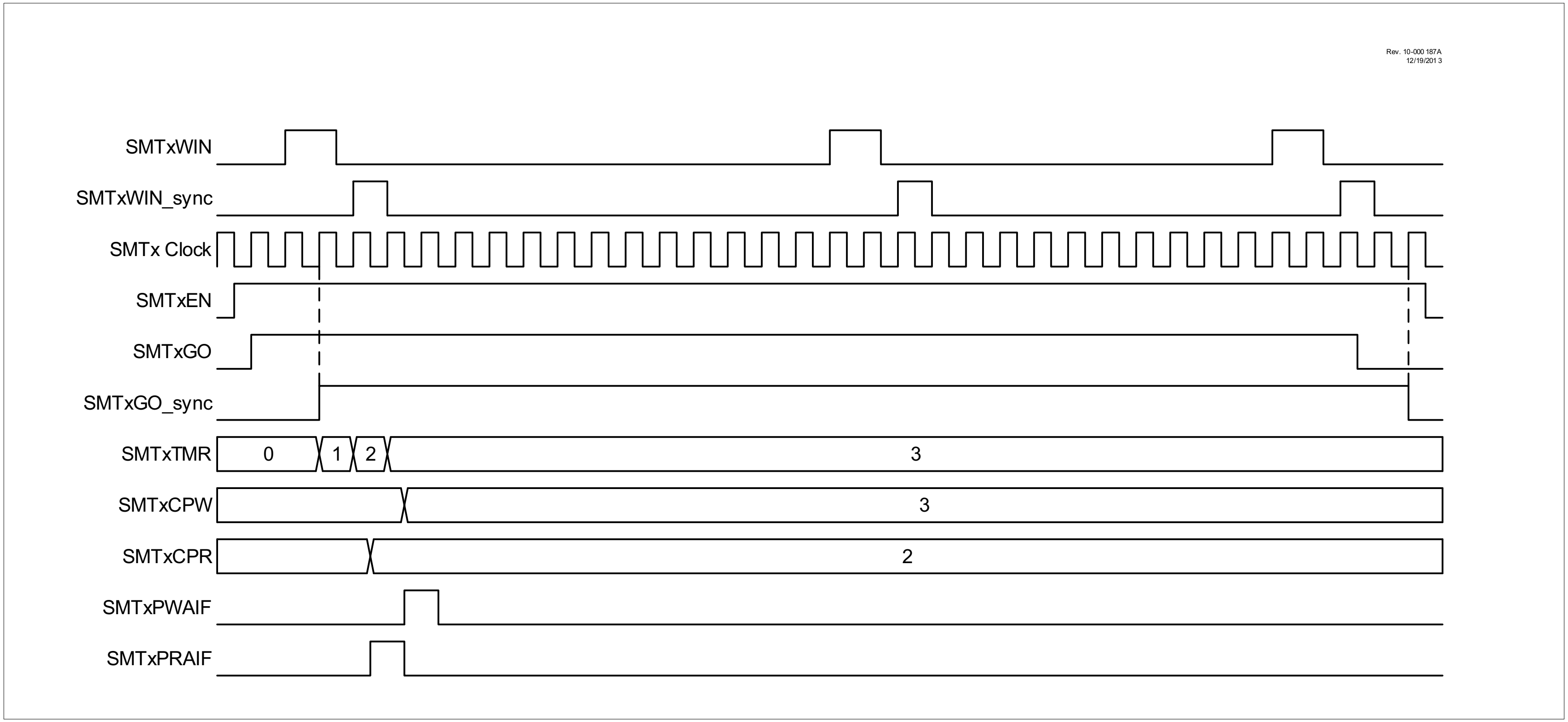


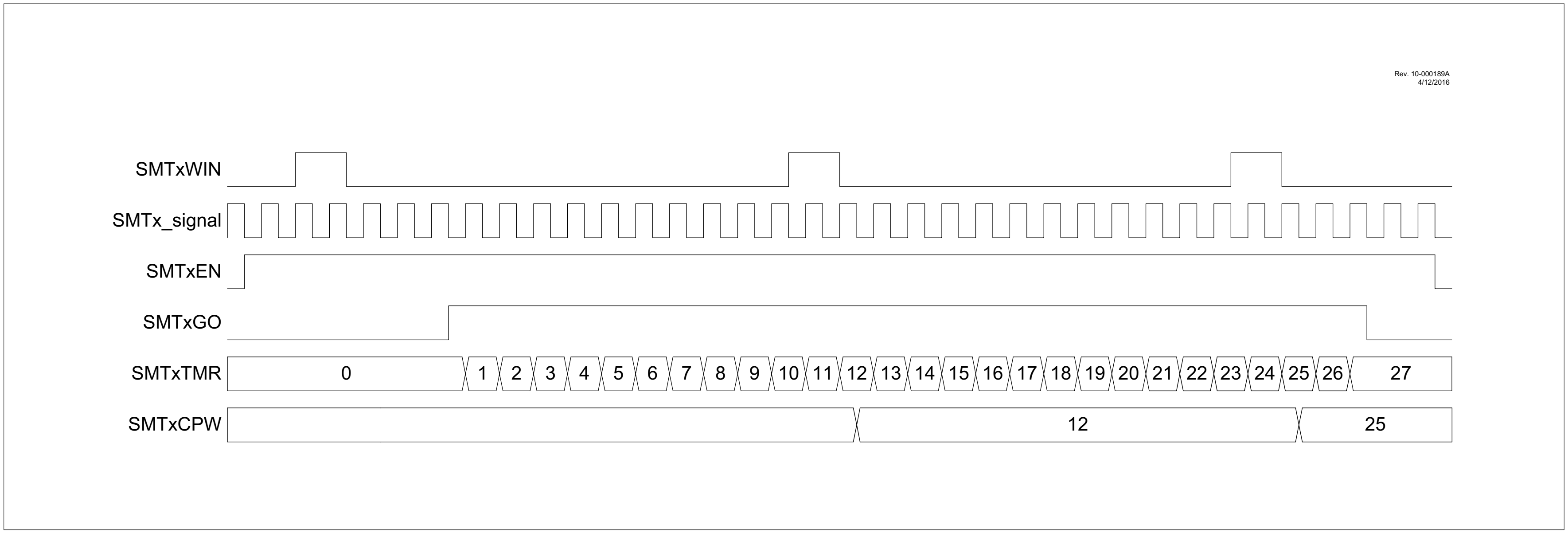
FIGURE 25-17: CAPTURE MODE SINGLE ACQUISITION TIMING DIAGRAM



25.6.9 COUNTER MODE

This mode increments the timer on each pulse of the SMTx_signal input. This mode is asynchronous to the SMT clock and uses the SMTx_signal as a time source. The SMTxCPW register will be updated with the current SMTxTMR value on the falling edge of the SMTxWIN input. See Figure 25-18.

FIGURE 25-18: COUNTER MODE TIMING DIAGRAM



25.6.10 GATED COUNTER MODE

This mode counts pulses on the SMTx_signal input, gated by the SMTxWIN input. It begins incrementing the timer upon seeing a rising edge of the SMTxWIN input and updates the SMTxCPW register upon a falling edge on the SMTxWIN input. See Figure 25-19 and Figure 25-20.

FIGURE 25-19: GATED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM

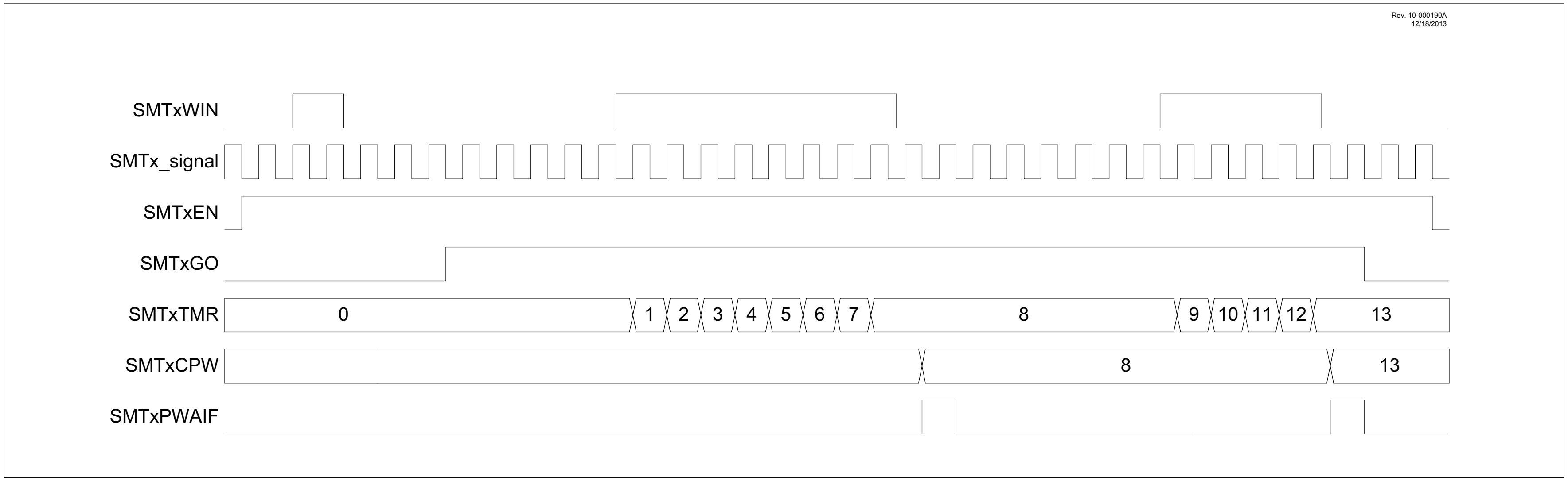
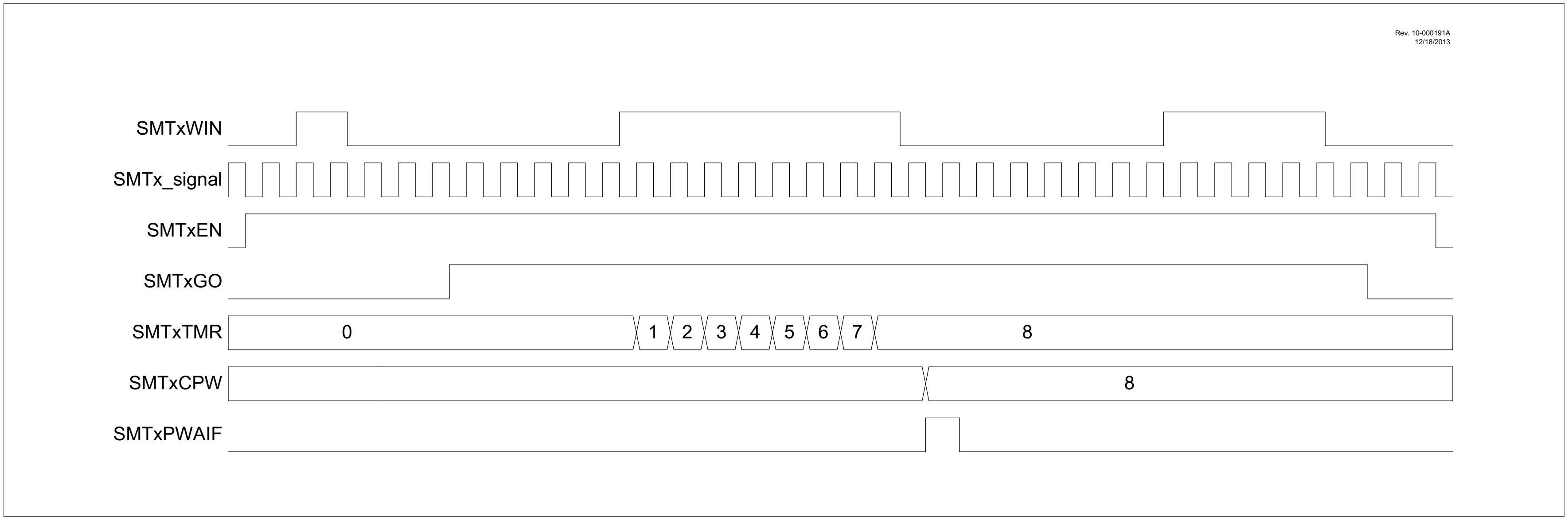


FIGURE 25-20: GATED COUNTER MODE SINGLE ACQUISITION TIMING DIAGRAM



25.6.11 WINDOWED COUNTER MODE

This mode counts pulses on the SMTx_signal input, within a window dictated by the SMTxWIN input. It begins counting upon seeing a rising edge of the SMTxWIN input, updates the SMTxCPW register on a falling edge of the SMTxWIN input, and updates the SMTxCPR register on each rising edge of the SMTxWIN input beyond the first. See Figure 25-21 and Figure 25-22.

FIGURE 25-21: WINDOWED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM

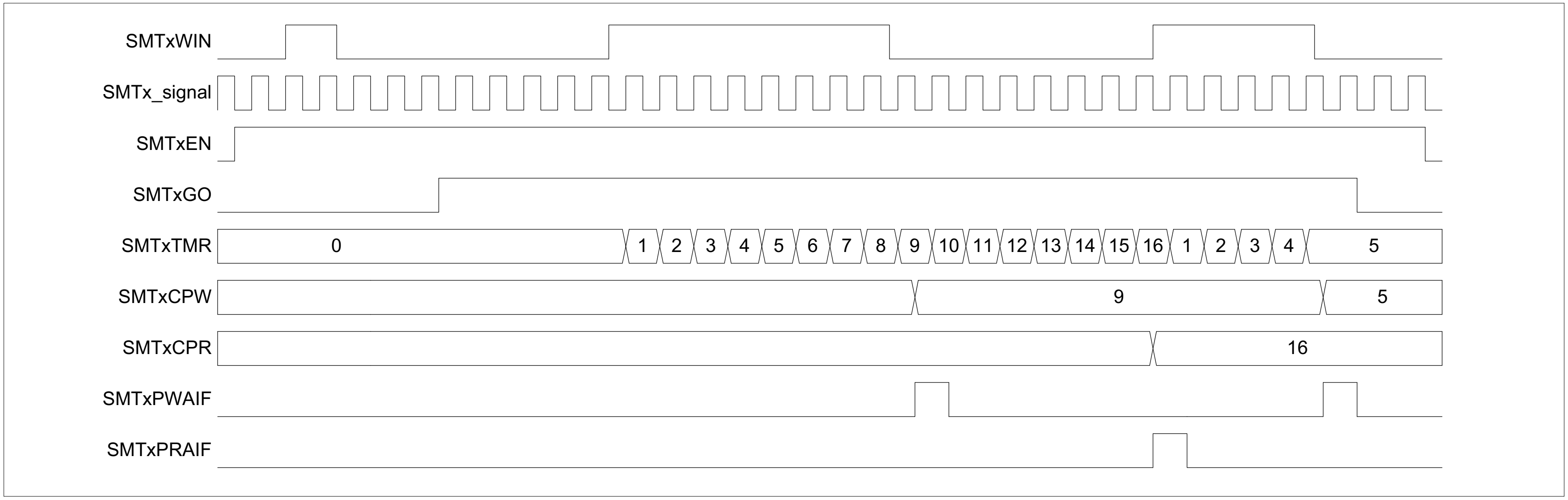
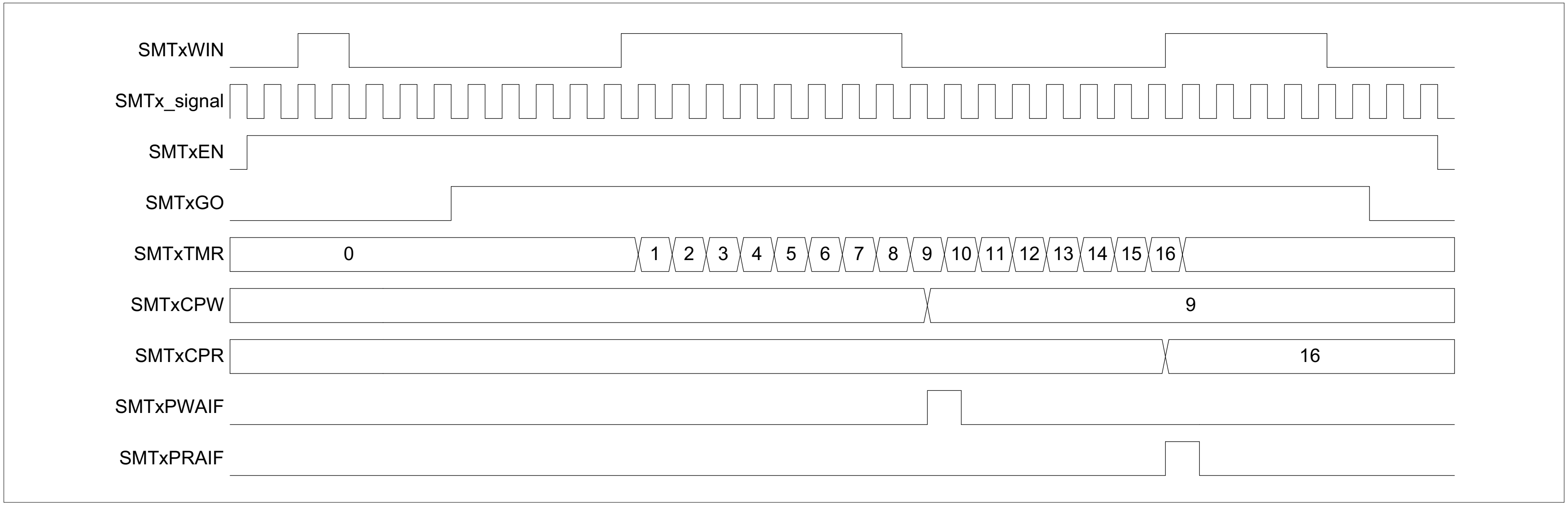


FIGURE 25-22: WINDOWED COUNTER MODE SINGLE ACQUISITION TIMING DIAGRAM



25.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

25.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMTxCPW and SMTxCPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMTxCPW interrupt is controlled by SMTxPWAIF and SMTxPWAIE bits in registers PIR4 and PIE4, respectively. The SMTxCPR interrupt is controlled by the SMTxPRAIF and SMTxPRAIE bits, also located in registers PIR4 and PIE4, respectively.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMTxCLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

25.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in **Section 25.1.2 "Period Match interrupt"**, the SMT will also interrupt upon SMTxTMR, matching SMTxPR with its period match limit functionality described in **Section25.3 "Halt Operation"**. The period match interrupt is controlled by SMTxIF and SMTxIE, located in registers PIR4 and PIE4, respectively.

25.8 Register Definitions: SMT Control

Long bit name prefixes for the Signal Measurement Timer peripherals are shown in Table 25-2. Refer to **Section 1.1 "Register and Bit Naming Conventions"** for more information.

TABLE 25-2:

Peripheral	Bit Name Prefix
SMT1	SMT1
SMT2	SMT2

REGISTER 25-1: SMTxCON0: SMT CONTROL REGISTER 0

R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN[1]	—	STP	WPOL	SPOL	CPOL	SMTxPS<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	**EN:** SMT Enable bit[1] 1 = SMT is enabled 0 = SMT is disabled; internal states are reset, clock requests are disabled
bit 6	**Unimplemented**: Read as '0'
bit 5	**STP:** SMT Counter Halt Enable bit When SMTxTMR = SMTxPR: 1 = Counter remains SMTxPR; period match interrupt occurs when clocked 0 = Counter resets to 24'h000000; period match interrupt occurs when clocked
bit 4	**WPOL:** SMTxWIN Input Polarity Control bit 1 = SMTxWIN signal is active-low/falling edge enabled 0 = SMTxWIN signal is active-high/rising edge enabled
bit 3	**SPOL:** SMTxSIG Input Polarity Control bit 1 = SMTx_signal is active-low/falling edge enabled 0 = SMTx_signal is active-high/rising edge enabled
bit 2	**CPOL:** SMT Clock Input Polarity Control bit 1 = SMTxTMR increments on the falling edge of the selected clock signal 0 = SMTxTMR increments on the rising edge of the selected clock signal
bit 1-0	**SMTxPS<1:0>:** SMT Prescale Select bits 11 = Prescaler = 1:8 10 = Prescaler = 1:4 01 = Prescaler = 1:2 00 = Prescaler = 1:1

Note 1: Setting EN to '0' does not affect the register contents.

REGISTER 25-2: SMTxCON1: SMT CONTROL REGISTER 1

R/W/HC-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxGO	REPEAT	—	—	MODE<3:0>			
bit 7							bit 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 **SMTxGO:** SMT GO Data Acquisition bit
`1` = Incrementing, acquiring data is enabled
`0` = Incrementing, acquiring data is disabled

bit 6 **REPEAT:** SMT Repeat Acquisition Enable bit
`1` = Repeat Data Acquisition mode is enabled
`0` = Single Acquisition mode is enabled

bit 5-4 **Unimplemented**: Read as ‘`0`’

bit 3-0 **MODE<3:0>** SMT Operation Mode Select bits
`1111` = Reserved
•
•
•
`1011` = Reserved
`1010` = Windowed counter
`1001` = Gated counter
`1000` = Counter
`0111` = Capture
`0110` = Time of flight
`0101` = Gated windowed measure
`0100` = Windowed measure
`0011` = High and low time measurement
`0010` = Period and Duty-Cycle Acquisition
`0001` = Gated Timer
`0000` = Timer

REGISTER 25-3: SMTxSTAT: SMT STATUS REGISTER

R/W/HC-0/0	R/W/HC-0/0	R/W/HC-0/0	U-0	U-0	R-0/0	R-0/0	R-0/0
CPRUP	CPWUP	RST	—	—	TS	WS	AS
bit 7							bit 0

Legend:		
HC = Bit is 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	**CPRUP:** SMT Manual Period Buffer Update bit `1` = Request update to SMTxCPRx registers `0` = SMTxCPRx registers update is complete
bit 6	**CPWUP:** SMT Manual Pulse Width Buffer Update bit `1` = Request update to SMTxCPW registers `0` = SMTxCPW registers update is complete
bit 5	**RST:** SMT Manual Timer Reset bit `1` = Request Reset to SMTxTMR registers `0` = SMTxTMR registers update is complete
bit 4-3	**Unimplemented**: Read as ‘`0`’
bit 2	**TS:** SMT GO Value Status bit `1` = SMT timer is incrementing `0` = SMT timer is not incrementing
bit 1	**WS:** SMTxWIN Value Status bit `1` = SMT window is open `0` = SMT window is closed
bit 0	**AS:** SMT_signal Value Status bit `1` = SMT acquisition is in progress `0` = SMT acquisition is not in progress

REGISTER 25-4: SMTxCLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<2:0>		
bit 7							bit 0

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

bit 7-3 **Unimplemented**: Read as ‘`0`’

bit 2-0 **CSEL<2:0>:** SMT Clock Selection bits

`111` = Reserved
`110` = Reserved
`101` = Reserved
`100` = MFINTOSC/16
`011` = LFINTOSC
`010` = HFINTOSC 16 MHz
`001` = FOSC/4
`000` = FOSC

REGISTER 25-5: SMTxWIN: SMTx WINDOW INPUT SELECT 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
—	—	—	—	WSEL<3:0>			
bit 7							bit 0

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

bit 7-4 **Unimplemented**: Read as ‘`0`’

bit 3-0 **WSEL<3:0>:** SMTx Window Selection bits

`1111` = Reserved
•
•
•
`1001` = Reserved
`1000` = TMR6_postscaled
`0111` = TMR4_postscaled
`0110` = TMR2_postscaled
`0101` = ZCD1_out
`0100` = CCP2_out
`0011` = CCP1_out
`0010` = C2OUT_sync[1]
`0001` = C1OUT_sync
`0000` = SMTWINx pin

Note 1: PIC16(L)F1613 only. Reserved on PIC12(L)F1612.

REGISTER 25-6: SMT1SIG: SMT1 SIGNAL INPUT SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	SSEL<2:0>		
bit 7							bit 0

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

bit 7-3 **Unimplemented**: Read as ‘`0`’

bit 2-0 **SSEL<2:0>:** SMT1 Signal Selection bits
`111` = Reserved
`110` = TMR6_postscaled
`101` = TMR4_postscaled
`100` = TMR2_postscaled
`011` = ZCD1_out
`010` = C2OUT_sync(1)
`001` = C1OUT_sync
`000` = SMTxSIG pin

Note 1: PIC16(L)F1613 only. Reserved on PIC12(L)F1612.

REGISTER 25-7: SMTxTMRL: SMT TIMER REGISTER – LOW BYTE

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxTMR<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	**SMTxTMR<7:0>**: Significant bits of the SMT Counter – Low Byte

REGISTER 25-8: SMTxTMRH: SMT TIMER REGISTER – HIGH BYTE

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxTMR<15:8>							
bit 7							bit 0

Legend:

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

bit 7-0	**SMTxTMR<15:8>**: Significant bits of the SMT Counter – High Byte

REGISTER 25-9: SMTxTMRU: SMT TIMER REGISTER – UPPER BYTE

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxTMR<23:16>							
bit 7							bit 0

Legend:

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

bit 7-0	**SMTxTMR<23:16>**: Significant bits of the SMT Counter – Upper Byte

REGISTER 25-10: SMTxCPRL: SMT CAPTURED PERIOD REGISTER – LOW BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPR<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	**SMTxCPR<7:0>**: Significant bits of the SMT Period Latch – Low Byte

REGISTER 25-11: SMTxCPRH: SMT CAPTURED PERIOD REGISTER – HIGH BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPR<15:8>							
bit 7							bit 0

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

bit 7-0	**SMTxCPR<15:8>**: Significant bits of the SMT Period Latch – High Byte

REGISTER 25-12: SMTxCPRU: SMT CAPTURED PERIOD REGISTER – UPPER BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPR<23:16>							
bit 7							bit 0

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

bit 7-0	**SMTxCPR<23:16>**: Significant bits of the SMT Period Latch – Upper Byte

REGISTER 25-13: SMTxCPWL: SMT CAPTURED PULSE WIDTH REGISTER – LOW BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPW<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	**SMTxCPW<7:0>**: Significant bits of the SMT PW Latch – Low Byte

REGISTER 25-14: SMTxCPWH: SMT CAPTURED PULSE WIDTH REGISTER – HIGH BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPW<15:8>							
bit 7							bit 0

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

bit 7-0	**SMTxCPW<15:8>**: Significant bits of the SMT PW Latch – High Byte

REGISTER 25-15: SMTxCPWU: SMT CAPTURED PULSE WIDTH REGISTER – UPPER BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPW<23:16>							
bit 7							bit 0

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

bit 7-0	**SMTxCPW<23:16>**: Significant bits of the SMT PW Latch – Upper Byte

REGISTER 25-16: SMTxPRL: SMT PERIOD REGISTER – LOW BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<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	**SMTxPR<7:0>**: Significant bits of the SMT Timer Value for Period Match – Low Byte

REGISTER 25-17: SMTxPRH: SMT PERIOD REGISTER – HIGH BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<15:8>							
bit 7							bit 0

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

bit 7-0	**SMTxPR<15:8>**: Significant bits of the SMT Timer Value for Period Match – High Byte

REGISTER 25-18: SMTxPRU: SMT PERIOD REGISTER – UPPER BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<23:16>							
bit 7							bit 0

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

bit 7-0	**SMTxPR<23:16>**: Significant bits of the SMT Timer Value for Period Match – Upper Byte

TABLE 25-3: SUMMARY OF REGISTERS ASSOCIATED WITH SMTx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	86
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	90
SMT1CLK	—	—	—	—	—	CSEL<2:0>			299
SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		296
SMT1CON1	SMT1GO	REPEAT	—	—	MODE<3:0>				297
SMT1CPRH	SMT1CPR<15:8>								302
SMT1CPRL	SMT1CPR<7:0>								302
SMT1CPRU	SMT1CPR<23:16>								302
SMT1CPWH	SMT1CPW<15:8>								303
SMT1CPWL	SMT1CPW<7:0>								303
SMT1CPWU	SMT1CPW<23:16>								303
SMT1PRH	SMT1PR<15:8>								304
SMT1PRL	SMT1PR<7:0>								304
SMT1PRU	SMT1PR<23:16>								304
SMT1SIG	—	—	—	—	—	SSEL<2:0>			300
SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	298
SMT1TMRH	SMT1TMR<15:8>								301
SMT1TMRL	SMT1TMR<7:0>								301
SMT1TMRU	SMT1TMR<23:16>								301
SMT1WIN	—	—	—	—	WSEL<3:0>				299
SMT2CLK	—	—	—	—	—	CSEL<2:0>			299
SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT2PS<1:0>		296
SMT2CON1	SMT2GO	REPEAT	—	—	MODE<3:0>				297
SMT2CPRH	SMT2CPR<15:8>								302
SMT2CPRL	SMT2CPR<7:0>								302
SMT2CPRU	SMT2CPR<23:16>								302
SMT2CPWH	SMT2CPW<15:8>								303
SMT2CPWL	SMT2CPW<7:0>								303
SMT2CPWU	SMT2CPW<23:16>								303
SMT2PRH	SMT2PR<15:8>								304
SMT2PRL	SMT2PR<7:0>								304
SMT2PRU	SMT2PR<23:16>								304
SMT2SIG	—	—	—	—	—	SSEL<2:0>			300
SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	298
SMT2TMRH	SMT2TMR<15:8>								301
SMT2TMRL	SMT2TMR<7:0>								301
SMT2TMRU	SMT2TMR<23:16>								301
SMT2WIN	—	—	—	WSEL<4:0>					299

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

26.0 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
- $\overline{\text{MCLR}}$/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 *"PIC12(L)F1612/PIC16(L)F161X Memory Programming Specification"* (DS40001720).

26.1 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 $\overline{\text{MCLR}}$/VPP to VIHH.

26.2 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. $\overline{\text{MCLR}}$ is brought to VIL.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{\text{MCLR}}$ must be held at VIL for as long as Program/Verify mode is to be maintained.

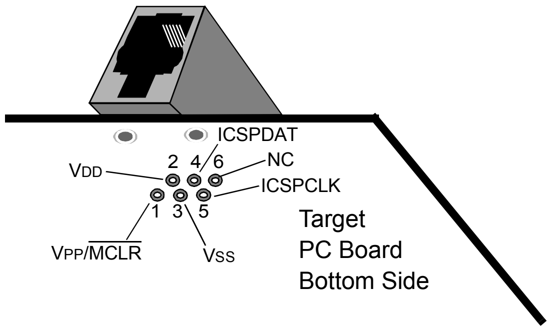
If low-voltage programming is enabled (LVP = `1`), the $\overline{\text{MCLR}}$ Reset function is automatically enabled and cannot be disabled. See **Section6.5 "MCLR"** for more information.

The LVP bit can only be reprogrammed to '`0`' by using the High-Voltage Programming mode.

26.3 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 26-1.

FIGURE 26-1: ICD RJ-11 STYLE CONNECTOR INTERFACE

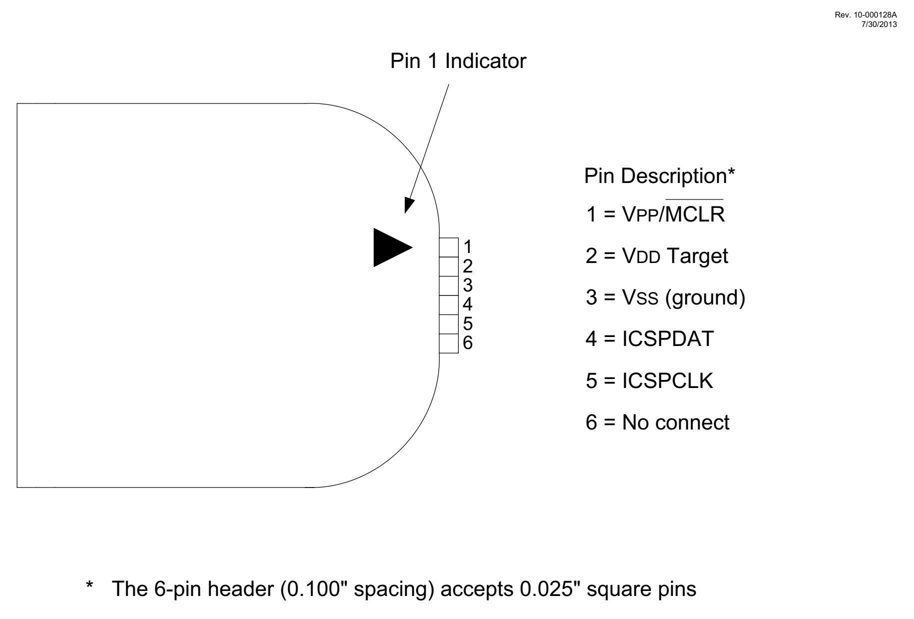


Pin Description*

1 = VPP/$\overline{\text{MCLR}}$
2 = VDD Target
3 = VSS (ground)
4 = ICSPDAT
5 = ICSPCLK
6 = No Connect

Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 26-2.

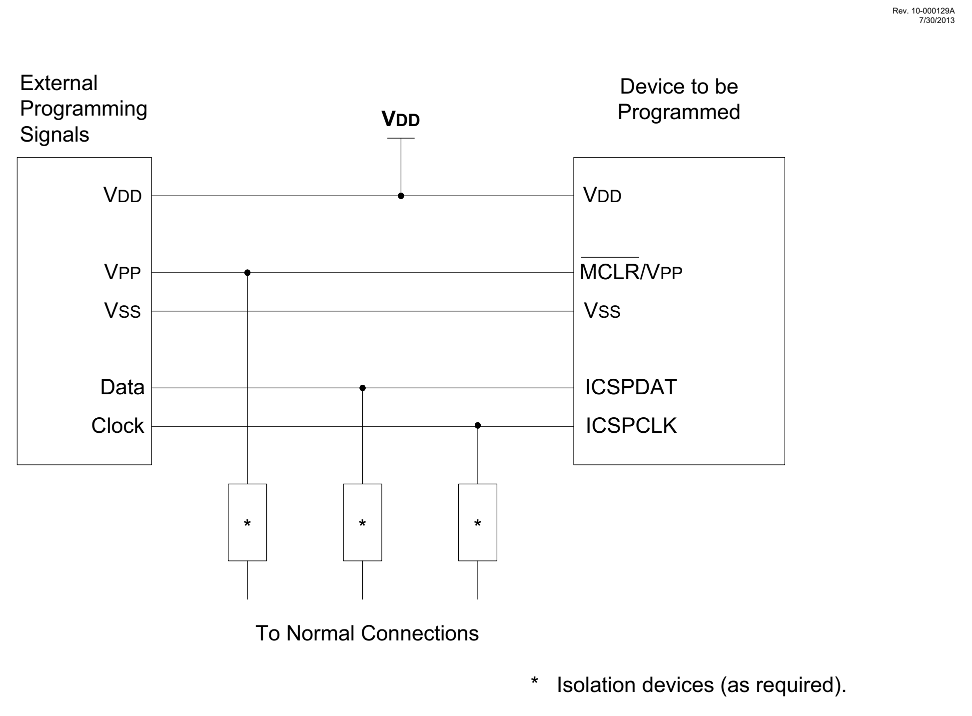
FIGURE 26-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE



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 26-3 for more information.

FIGURE 26-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



27.0 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 27-3 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.

TABLE 27-1: 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 27-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{\text{TO}}$	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
$\overline{\text{PD}}$	Power-Down bit

27.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator ‘d’. A read operation is performed on a register even if the instruction writes to that register.

FIGURE 27-1: GENERAL FORMAT FOR INSTRUCTIONS

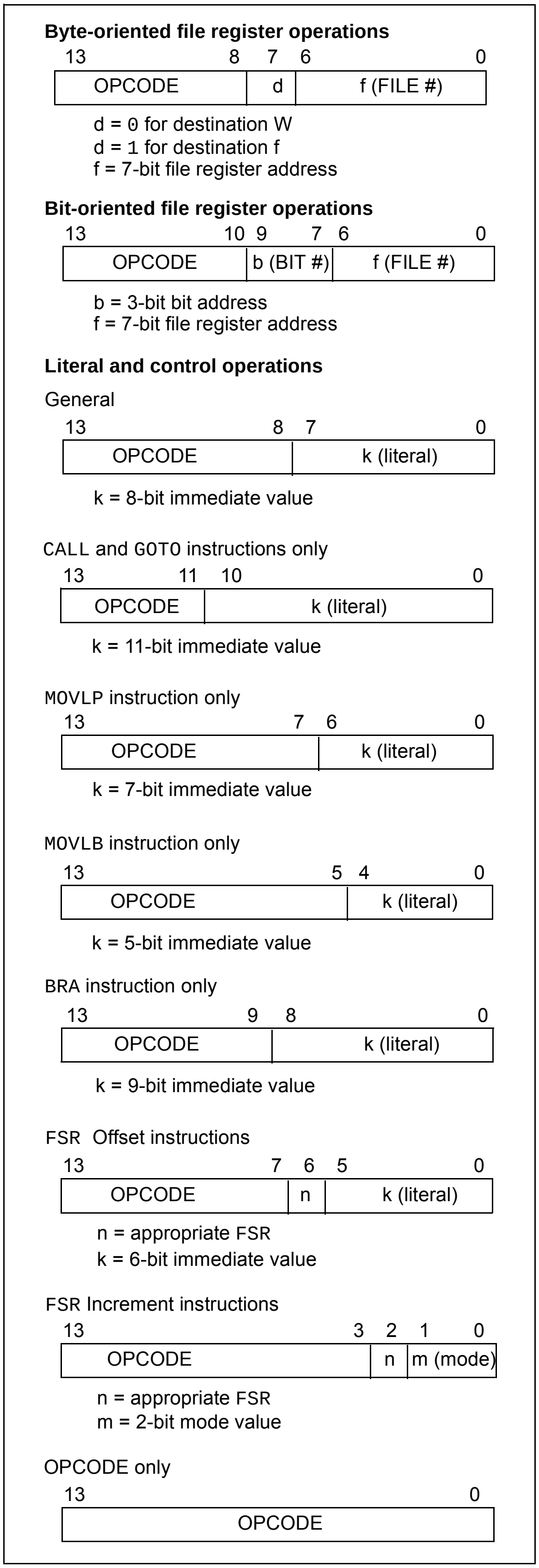


TABLE 27-3: ENHANCED MID-RANGE INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
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 27-3: 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`	$\overline{\text{TO}}$, $\overline{\text{PD}}$	
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`	$\overline{\text{TO}}$, $\overline{\text{PD}}$	
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.

27.2 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:	$-256 \le label - PC + 1 \le 255$ $-256 \le k \le 255$
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:	$0 \le k \le 2047$
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:	$0 \leq f \leq 127$
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:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	($\bar{f}$) → (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:	$0 \leq f \leq 127$ $d \in [0,1]$
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 ≤ k ≤ 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 ≤ f ≤ 127 d ∈ [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’.

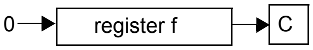
INCFSZ	**Increment f, Skip if 0**
Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, 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 ≤ k ≤ 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 ≤ f ≤ 127 d ∈ [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
Example:	`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 ≤ k ≤ 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 ≤ k ≤ 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 ≤ k ≤ 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 ≤ k ≤ 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
<u>Example:</u>	`MOVLW 0x5A` After Instruction W = 0x5A

MOVWF Move W to f

Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 127
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register ‘f’.
Words:	1
Cycles:	1
<u>Example:</u>	`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 ≤ k ≤ 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 $\overline{RI}$ 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 → PC, 1 → 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
Example:	`RETFIE` After Interrupt PC = TOS GIE = 1

RETLW	**Return with literal in W**
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → 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

Example:

```
      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 → 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:	$0 \leq f \leq 127$ $d \in$ [`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

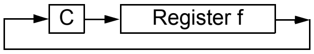
```
REG1    =   1110 0110
C       =   0
```

After Instruction

```
REG1    =   1110 0110
W       =   1100 1100
C       =   1
```

RRF — Rotate Right f through Carry

Syntax:	[*label*] RRF f,d
Operands:	$0 \leq f \leq 127$ $d \in$ [`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:	$0 \le k \le 255$
Operation:	$k - (W) \rightarrow (W)$
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:	$0 \le f \le 127$ $d \in$ [`0`,`1`]
Operation:	$(f) - (W) \rightarrow$ (destination)
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is '`0`', the result is stored in the W register. If 'd' is '`1`', the result is stored back in register 'f.

C = `0`	W > f
C = `1`	W ≤ 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:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(f) – (W) – (\overline{B}) \rightarrow$ dest
Status Affected:	C, DC, Z
Description:	Subtract W and the BORROW flag (CARRY) from register 'f' (2's 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:	$0 \le f \le 127$ $d \in$ [`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:	$5 \le f \le 7$
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:	$0 \leq k \leq 255$
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:	$0 \leq f \leq 127$ $d \in$ [`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'.

28.0 ELECTRICAL SPECIFICATIONS

28.1 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	
PIC12F1612/16F1613	-0.3V to +6.5V
PIC12LF1612/16F1613	-0.3V to +4.0V
on $\overline{\text{MCLR}}$ pin	-0.3V to +9.0V
on all other pins	-0.3V to (V_{DD} + 0.3V)
Maximum current	
on V_{SS} pin(1)	
$-40°C \leq T_A \leq +85°C$	250 mA
$+85°C \leq T_A \leq +125°C$	85 mA
on V_{DD} pin(1)	
$-40°C \leq T_A \leq +85°C$	250 mA
$+85°C \leq T_A \leq +125°C$	85 mA
Sunk by any standard I/O pin	50 mA
Sourced by any standard I/O pin	50 mA
Sunk by any High Current I/O pin	100 mA
Sourced by any High Current I/O pin	100 mA
Clamp current, I_K ($V_{PIN} < 0$ or $V_{PIN} > V_{DD}$)	±20 mA
Total power dissipation(2)	800 mW

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 28-6: “Thermal Characteristics” to calculate device specifications.

2: Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum(V_{Ol} \times I_{OL})$.

.

† 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.

28.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \leq T_A \leq T_{A\_MAX}$

VDD — Operating Supply Voltage(1)

PIC12LF1612/16F1613

VDDMIN (Fosc ≤ 16 MHz)	+1.8V
VDDMIN (Fosc ≤ 32 MHz)	+2.5V
VDDMAX	+3.6V

PIC12F1612/16F1613

VDDMIN (Fosc ≤ 16 MHz)	+2.3V
VDDMIN (Fosc ≤ 32 MHz)	+2.5V
VDDMAX	+5.5V

TA — Operating Ambient Temperature Range

Industrial Temperature

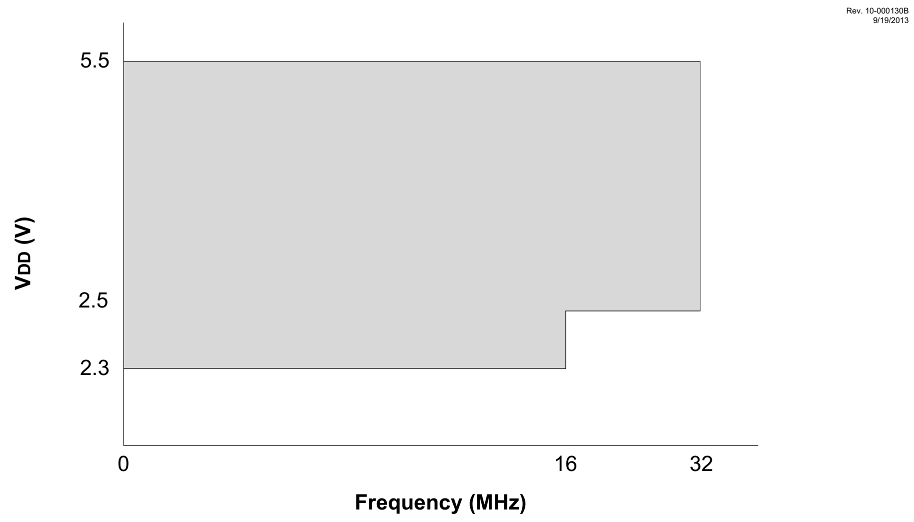
TA_MIN	-40°C
TA_MAX	+85°C

Extended Temperature

TA_MIN	-40°C
TA_MAX	+125°C

Note 1: See Parameter D001, DS Characteristics: Supply Voltage.

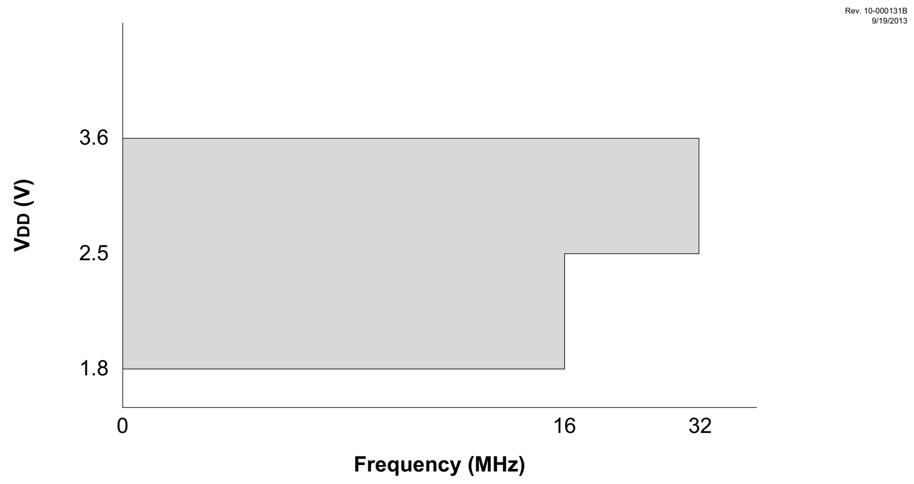
FIGURE 28-1: VOLTAGE FREQUENCY GRAPH, -40°C ≤ T_A ≤ +125°C, PIC12F1612/16F1613 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: Refer to Table 28-7 for each Oscillator mode's supported frequencies.

FIGURE 28-2: VOLTAGE FREQUENCY GRAPH, -40°C ≤ T_A ≤ +125°C, PIC12LF1612/16F1613 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: Refer to Table 28-7 for each Oscillator mode's supported frequencies.

28.3 DC Characteristics

TABLE 28-1: SUPPLY VOLTAGE

PIC12F1612/16F1613			Standard Operating Conditions (unless otherwise stated)				
PIC12F1612/16F1613							
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
D001			2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz
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
D003			—	1.024	—	V	-40°C ≤ TA ≤ +85°C
D003A	VADFVR	**FVR Gain Voltage Accuracy for ADC**					
			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V
D003A			-5	—	+5	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V 4x VFVR, VDD ≥ 4.75V
D003B	VCDAFVR	**FVR Gain Voltage Accuracy for Comparator/DAC**					
			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V
D003B			-7	—	+7	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V 4x VFVR, VDD ≥ 4.75V
D004*	SVDD	**VDD Rise Rate**(2)					
			0.05	—	—	V/ms	Ensures that the Power-on Reset signal is released properly.
D004*			0.05	—	—	V/ms	Ensures that the Power-on Reset signal is released properly.

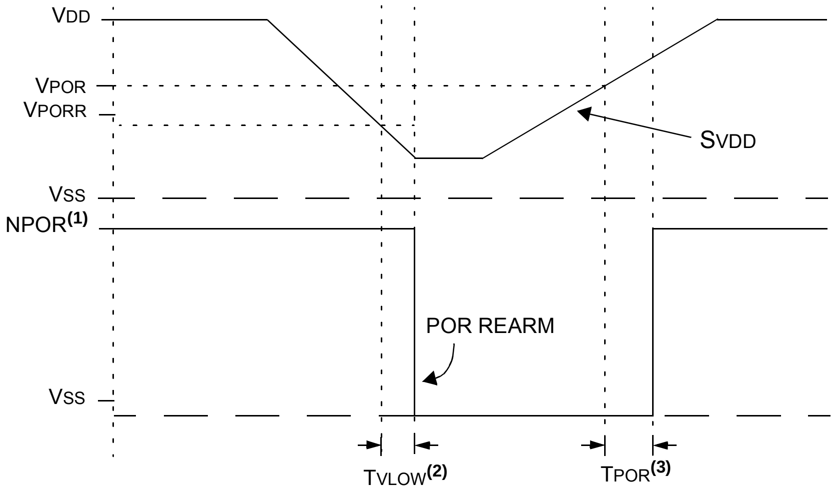
* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: See Figure 28-3, POR and POR REARM with Slow Rising VDD.

FIGURE 28-3: 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 28-2: SUPPLY CURRENT (IDD)(1,2)

PIC12LF1612/16F1613		Standard Operating Conditions (unless otherwise stated)					
PIC12F1612/16F1613							
Param. No.	**Device Characteristics**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**	
						VDD	**Note**
D013		—	30	90	µA	1.8	FOSC = 1 MHz, External Clock (ECM), Medium-Power mode
		—	55	110	µA	3.0	
D013		—	65	120	µA	2.3	FOSC = 1 MHz, External Clock (ECM), Medium-Power mode
		—	85	150	µA	3.0	
		—	115	200	µA	5.0	
D014		—	115	260	µA	1.8	FOSC = 4 MHz, External Clock (ECM), Medium-Power mode
		—	210	380	µA	3.0	
D014		—	180	310	µA	2.3	FOSC = 4 MHz, External Clock (ECM), Medium-Power mode
		—	240	410	µA	3.0	
		—	295	520	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: OSC1 = 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.

TABLE 28-2: SUPPLY CURRENT (IDD)(1,2) (CONTINUED)

PIC12LF1612/16F1613		Standard Operating Conditions (unless otherwise stated)					
PIC12F1612/16F1613							
Param. No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
D015		—	9.6	36	µA	1.8	FOSC = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	16.2	60	µA	3.0	
D015		—	39	84	µA	2.3	FOSC = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	45	90	µA	3.0	
		—	51	108	µA	5.0	
D016		—	215	360	µA	1.8	FOSC = 500 kHz, HFINTOSC
		—	275	480	µA	3.0	
D016		—	270	450	µA	2.3	FOSC = 500 kHz, HFINTOSC
		—	300	500	µA	3.0	
		—	350	620	µA	5.0	
D017*		—	410	800	µA	1.8	FOSC = 8 MHz, HFINTOSC
		—	630	1200	µA	3.0	
D017*		—	530	950	µA	2.3	FOSC = 8 MHz, HFINTOSC
		—	660	1300	µA	3.0	
		—	730	1400	µA	5.0	
D018		—	600	1200	µA	1.8	FOSC = 16 MHz, HFINTOSC
		—	970	1850	µA	3.0	
D018		—	780	1500	µA	2.3	FOSC = 16 MHz, HFINTOSC
		—	1000	1900	µA	3.0	
		—	1090	2100	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: OSC1 = 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.

TABLE 28-2: SUPPLY CURRENT (IDD)(1,2) (CONTINUED)

PIC12LF1612/16F1613		Standard Operating Conditions (unless otherwise stated)				
PIC12F1612/16F1613						
Param. No.	**Device Characteristics**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**	
						VDD	**Note**
D019		—	1.6	5.0	mA	3.0	FOSC = 32 MHz, HFINTOSC
		—	1.9	6.0	mA	3.6	
D019		—	1.6	5.0	mA	3.0	FOSC = 32 MHz, HFINTOSC
		—	1.9	6.0	mA	5.0	
D020A		—	1.6	5.0	mA	3.0	FOSC = 32 MHz, External Clock (ECH), High-Power mode
		—	1.9	6.0	mA	3.6	
D020A		—	1.6	5.0	mA	3.0	FOSC = 32 MHz, External Clock (ECH), High-Power mode
		—	1.9	6.0	mA	5.0	
D020B		—	6	16	μA	1.8	FOSC = 32 kHz, External Clock (ECL), Low-Power mode
		—	8	22	μA	3.0	
D020B		—	13	43	μA	2.3	FOSC = 32 kHz, External Clock (ECL), Low-Power mode
		—	15	55	μA	3.0	
		—	16	57	μA	5.0	
D020C		—	19	40	μA	1.8	FOSC = 500 kHz, External Clock (ECL), Low-Power mode
		—	32	60	μA	3.0	
D020C		—	31	60	μA	2.3	FOSC = 500 kHz, External Clock (ECL), Low-Power mode
		—	38	90	μA	3.0	
		—	44	100	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: OSC1 = 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.

TABLE 28-3: POWER-DOWN CURRENTS (IPD)(1,2)

PIC12LF1612/16F1613		Operating Conditions: (unless otherwise stated) Low-Power Sleep Mode						
PIC12F1612/16F1613		Low-Power Sleep Mode, VREGPM = 1						
Param. No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							VDD	Note
D022	Base IPD	—	0.020	1.0	8.0	μA	1.8	WDT, BOR, FVR disabled, all Peripherals inactive
		—	0.025	2.0	9.0	μA	3.0	
D022	Base IPD	—	0.25	3.0	10	μA	2.3	WDT, BOR, FVR disabled, all Peripherals inactive, Low-Power Sleep mode
		—	0.30	4.0	12	μA	3.0	
		—	0.40	6.0	15	μA	5.0	
D022A	Base IPD	—	9.8	16	18	μA	2.3	WDT, BOR, FVR disabled, all Peripherals inactive, Normal-Power Sleep mode, VREGPM = 0
		—	10.3	18	20	μA	3.0	
		—	11.5	21	26	μA	5.0	
D023		—	0.26	2.0	9.0	μA	1.8	WDT Current
		—	0.44	3.0	10	μA	3.0	
D023		—	0.43	6.0	15	μA	2.3	WDT Current
		—	0.53	7.0	20	μA	3.0	
		—	0.64	8.0	22	μA	5.0	
D023A		—	15	28	30	μA	1.8	FVR Current
		—	18	30	33	μA	3.0	
D023A		—	18	33	35	μA	2.3	FVR Current
		—	19	35	37	μA	3.0	
		—	20	37	39	μA	5.0	
D024		—	6.0	17	20	μA	3.0	BOR Current
D024		—	7.0	17	30	μA	3.0	BOR Current
		—	8.0	20	40	μA	5.0	
D24A		—	0.1	4.0	10	μA	3.0	LPBOR Current
D24A		—	0.35	5.0	14	μA	3.0	LPBOR Current
		—	0.45	8.0	17	μA	5.0	
D026		—	0.11	1.5	9.0	μA	1.8	ADC Current **(Note 3)**, No conversion in progress
		—	0.12	2.7	10	μA	3.0	
D026		—	0.30	4.0	11	μA	2.3	ADC Current **(Note 3)**, No conversion in progress
		—	0.35	5.0	13	μA	3.0	
		—	0.45	8.0	16	μA	5.0	
D026A*		—	250	—	—	μA	1.8	ADC Current **(Note 3)**, Conversion in progress
		—	250	—	—	μA	3.0	
D026A*		—	280	—	—	μA	2.3	ADC Current **(Note 3)**, Conversion in progress
		—	280	—	—	μA	3.0	
		—	280	—	—	μA	5.0	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Legend: TBD = To Be Determined

Note 1: The peripheral 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 28-3: POWER-DOWN CURRENTS (IPD)(1,2) (CONTINUED)

PIC12LF1612/16F1613		**Operating Conditions: (unless otherwise stated)** **Low-Power Sleep Mode**						
PIC12F1612/16F1613		**Low-Power Sleep Mode, VREGPM = 1**						
Param. No.	**Device Characteristics**	**Min.**	**Typ†**	**Max. +85°C**	**Max. +125°C**	**Units**	**Conditions**	
							VDD	**Note**
D027		—	7	22	25	μA	1.8	Comparator, CxSP = `0`
		—	8	23	27	μA	3.0	
D027		—	17	35	37	μA	2.3	Comparator, CxSP = `0`
		—	18	37	38	μA	3.0	
		—	19	38	40	μA	5.0	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Legend: TBD = To Be Determined

Note 1: The peripheral ∆ current can be determined by subtracting the base IPD current from this limit. Max. values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VSS.

3: ADC clock source is FRC.

TABLE 28-4: I/O PORTS

Standard Operating Conditions (unless otherwise stated)

Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
	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
D032		$\overline{\text{MCLR}}$	—	—	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
D042		$\overline{\text{MCLR}}$	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		$\overline{\text{MCLR}}$(3)	—	± 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(3)**					
D080		I/O Ports	—	—	0.6	V	IOL = 8.0 mA, VDD = 5.0V IOL = 6.0 mA, VDD = 3.3V IOL = 1.8 mA, VDD = 1.8V
	VOH	**Output High Voltage(3)**					
D090		I/O Ports	VDD - 0.7	—	—	V	IOH = 3.5 mA, VDD = 5.0V IOH = 3.0 mA, VDD = 3.3V IOH = 1.0 mA, VDD = 1.8V
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 $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Excluding OSC2 in CLKOUT mode.

TABLE 28-5: 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 $\leq T_A \leq$ +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 $\leq T_A \leq$ +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 28-6: THERMAL CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)

Param. No.	Sym.	Characteristic	Typ.	Units	Conditions
TH01	θ_{JA}	Thermal Resistance Junction to Ambient	62.2	°C/W	20-pin DIP package
			77.7	°C/W	20-pin SOIC package
			87.3	°C/W	20-pin SSOP package
			43	°C/W	20-pin QFN 4X4mm package
TH02	θ_{JC}	Thermal Resistance Junction to Case	27.5	°C/W	20-pin DIP package
			23.1	°C/W	20-pin SOIC package
			31.1	°C/W	20-pin SSOP package
			5.3	°C/W	20-pin QFN 4X4mm package
TH03	T_{JMAX}	Maximum Junction Temperature	150	°C	
TH04	P_D	Power Dissipation	—	W	$P_D = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD}$ x V_{DD}(1)
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	P_{DER}	Derated Power	—	W	$P_{DER} = PD_{MAX} (T_J - T_A)/\theta_{JA}$(2)

Note 1: I_{DD} is current to run the chip alone without driving any load on the output pins.

2: T_A = Ambient Temperature; T_J = Junction Temperature

28.4 AC Characteristics

Timing Parameter Symbology has been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

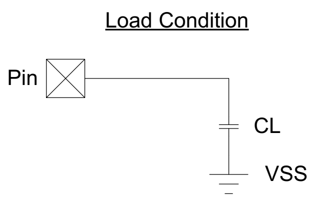
Lowercase letters (pp) and their meanings:

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

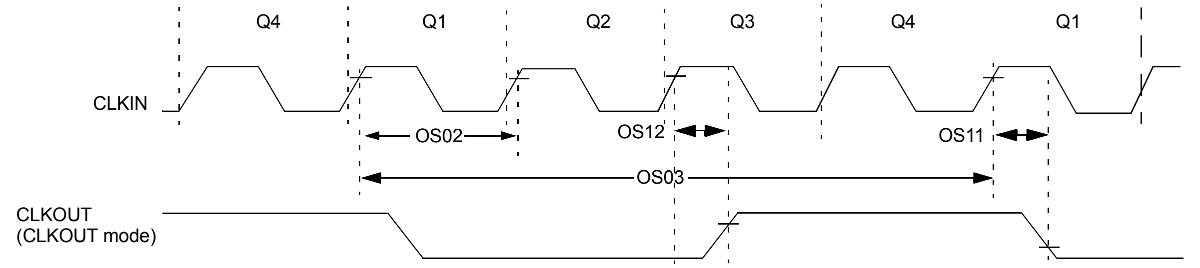
FIGURE 28-4: LOAD CONDITIONS



Rev. 10-000133A
8/1/2013

Legend: CL=50 pF for all pins

FIGURE 28-5: CLOCK TIMING



Note 1: See Table 28-10.

TABLE 28-7: 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 DC DC	— — —	0.5 4 32	MHz MHz MHz	External Clock (ECL) External Clock (ECM) External Clock (ECH)
OS02	TOSC	External CLKIN Period(1)	31.25	—	∞	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 28-8: OSCILLATOR PARAMETERS

Param. No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
Standard Operating Conditions (unless otherwise stated)								
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency[1]	—	—	16.0	—	MHz	**(Note 2)**
OS09	LFOSC	Internal LFINTOSC Frequency	—	—	31	—	kHz	**(Note 3)**
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 28-6: “HFINTOSC Frequency Accuracy over Device VDD and Temperature”,

3: See Figure 36-45: “LFINTOSC Frequency over VDD and Temperature, PIC12LF1612/16F1613 Only”, and Figure 36-46: “LFINTOSC Frequency over VDD and Temperature, PIC12F1612/16F1613 Only”.

FIGURE 28-6: HFINTOSC FREQUENCY ACCURACY OVER VDD AND TEMPERATURE

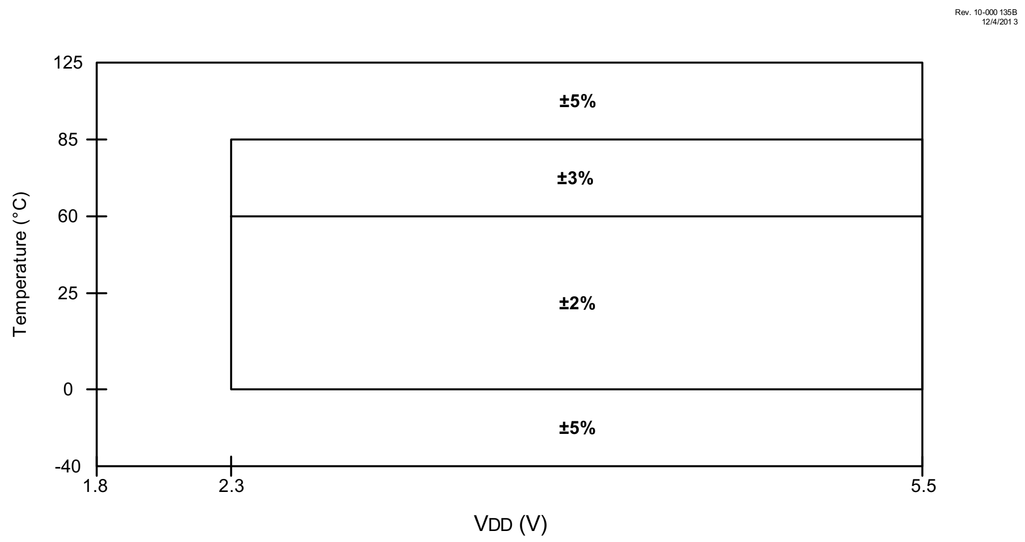


TABLE 28-9: PLL CLOCK TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)

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 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 28-7: CLKOUT AND I/O TIMING

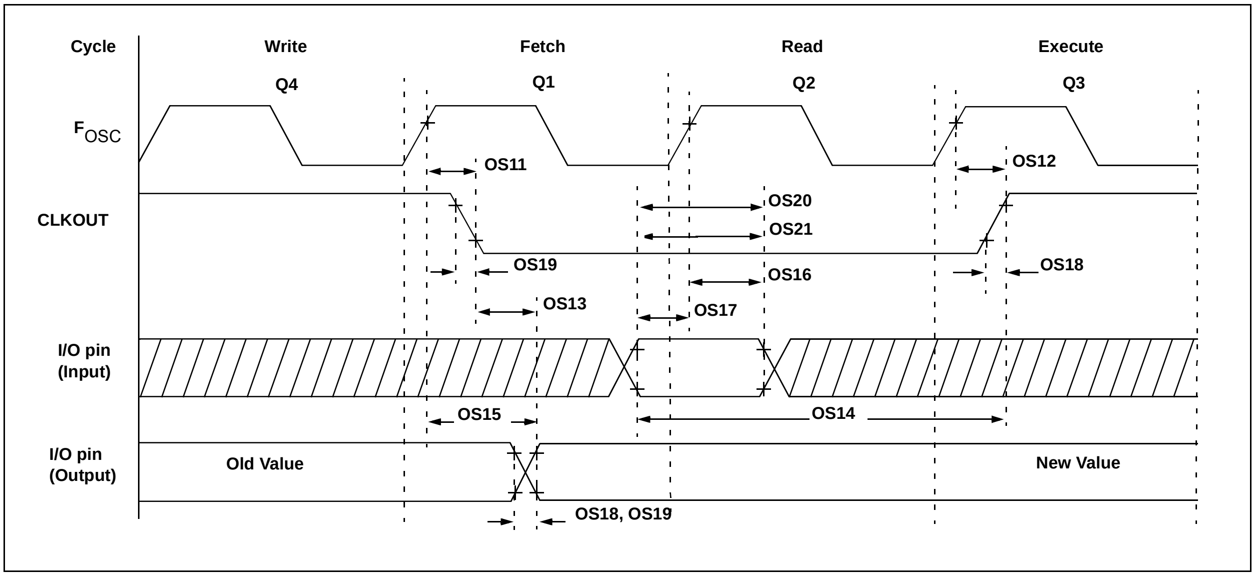


TABLE 28-10: CLKOUT AND I/O TIMING PARAMETERS

Standard Operating Conditions (unless otherwise stated)

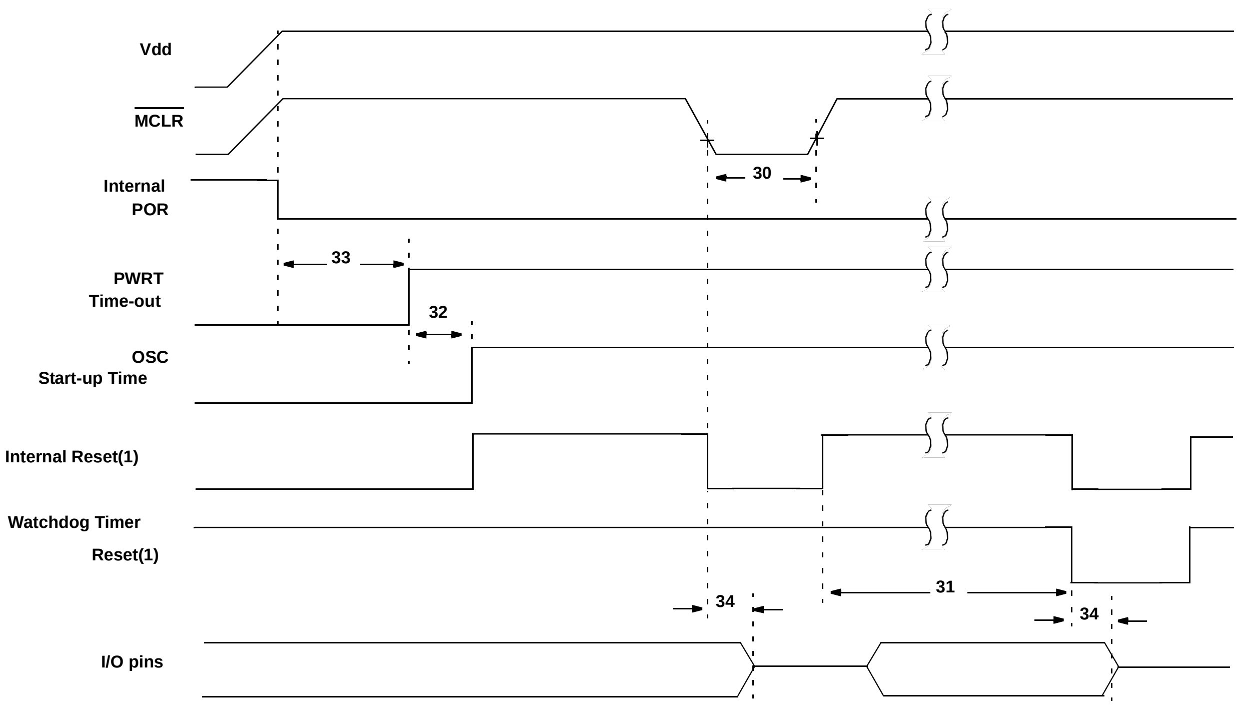
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS11	TosH2ckL	FOSC↑ to CLKOUT↓(1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	FOSC↑ to CLKOUT↑(1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid(1)	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑(1)	TOSC + 200 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 28-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

TABLE 28-11: 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	$\overline{\text{MCLR}}$ 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	$\overline{\text{PWRTE}}$ = `0`
34*	TIOZ	I/O high-impedance from $\overline{\text{MCLR}}$ 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` (PIC12F1612/16F1613) BORV = `1` (PIC12LF1612/16F1613)
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 28-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS

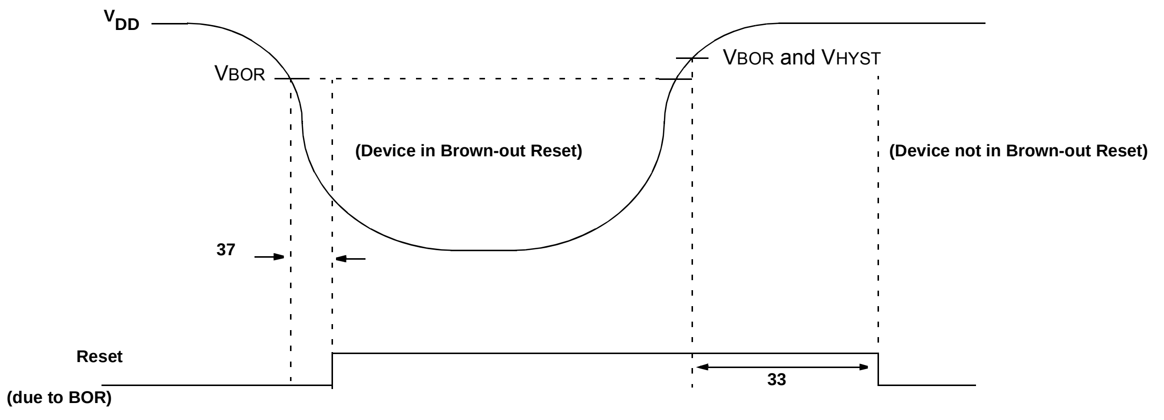


FIGURE 28-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

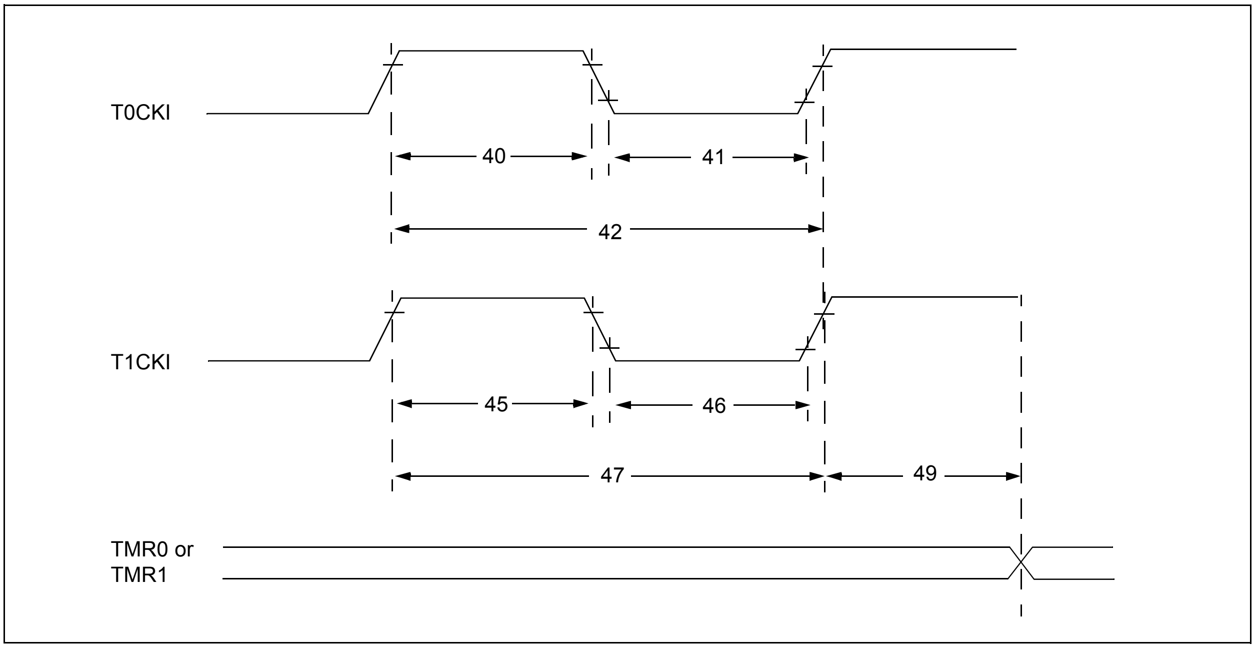


TABLE 28-12: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

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 $\frac{\text{TCY} + 40}{N}$	—	—	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 $\frac{\text{TCY} + 40}{N}$	—	—	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 28-13: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS(1,2,3)

Operating Conditions (unless otherwise stated)
VDD = 3.0V, TA = 25°C

Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD01	NR	Resolution	—	—	10	bit	
AD02	EIL	Integral Error	—	±1	±1.7	LSb	VREF = 3.0V
AD03	EDL	Differential Error	—	±1	±1	LSb	No missing codes VREF = 3.0V
AD04	EOFF	Offset Error	—	±1	±2.5	LSb	VREF = 3.0V
AD05	EGN	Gain Error	—	±1	±2.0	LSb	VREF = 3.0V
AD06	VREF	Reference Voltage	1.8	—	VDD	V	VREF = (VRPOS - VRNEG) (**Note 4**)
AD07	VAIN	Full-Scale Range	VSS	—	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	Can go higher if external 0.01μF 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 29.0 “DC and AC Characteristics Graphs and Charts”** for operating characterization.

4: ADC VREF is selected by ADPREF<0> bit.

FIGURE 28-11: ADC CONVERSION TIMING (ADC CLOCK FOSC-BASED)

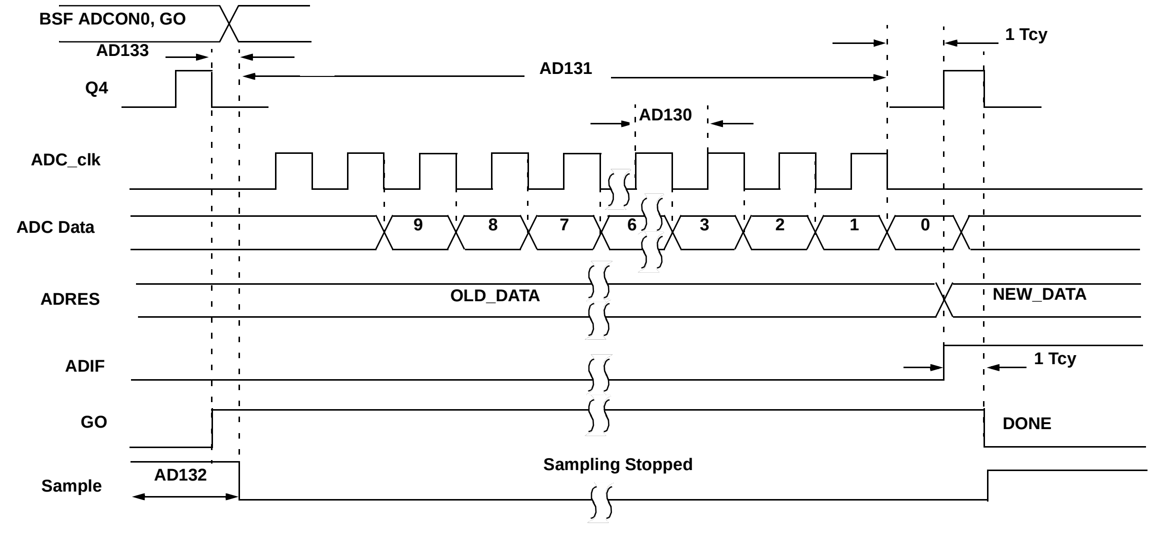
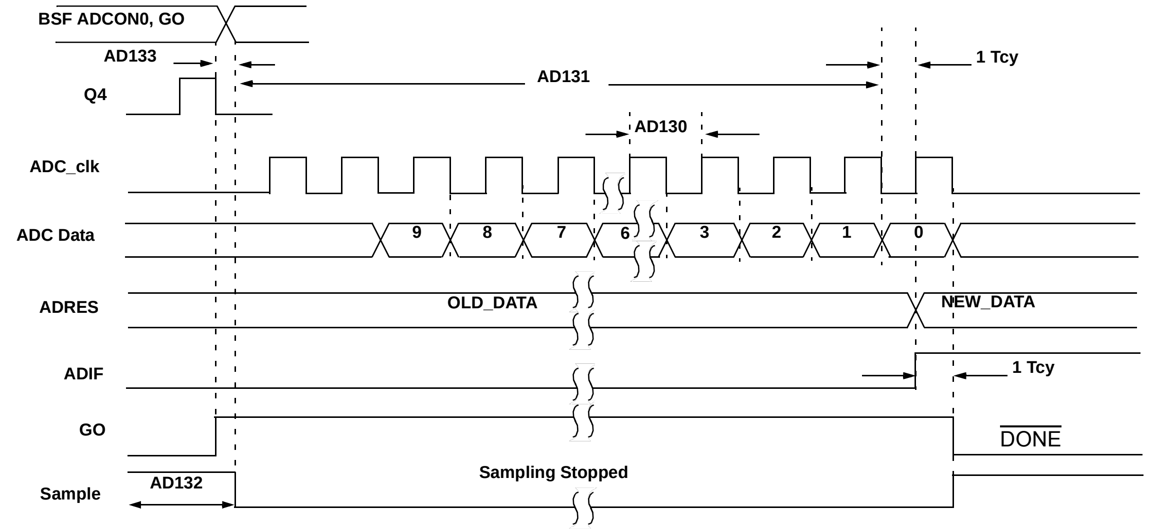


FIGURE 28-12: ADC CONVERSION TIMING (ADC CLOCK FROM FRC)



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 28-14: 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/$\overline{\text{DONE}}$ 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 28-15: 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 Ratio	—	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 29.0 “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 28-16: 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/256	—	V	
DAC02*	CACC	Absolute Accuracy	—	—	± 1.5	LSb	
DAC03*	CR	Unit Resistor Value (R)	—	—	—	Ω	
DAC04*	CST	Settling Time(2)	—	—	10	μs	

* These parameters are characterized but not tested.

Note 1: See **Section 29.0 “DC and AC Characteristics Graphs and Charts”** for operating characterization.

2: Settling time measured while DACR<4:0> transitions from ‘`0000`’ to ‘`1111`’.

TABLE 28-17: ZERO CROSS PIN SPECIFICATIONS

Operating Conditions (unless otherwise stated)
VDD = 3.0V, TA = 25°C

Param. No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
ZC01	ZCPINV	Voltage on Zero Cross Pin	—	0.75	—	V	
ZC02	ZCSRC	Source current	—	-300	-600	μA	
ZC03	ZCSNK	Sink current	—	300	600	μA	
ZC04	ZCISW	Response Time Rising Edge	—	1	—	μs	
		Response Time Falling Edge	—	1	—	μs	
ZC05	ZCOUT	Response Time Rising Edge	—	1	—	μs	
		Response Time Falling Edge	—	1	—	μs	

* These parameters are characterized but not tested.

29.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

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

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

Unless otherwise noted, all graphs apply to both the L and LF devices.

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

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

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

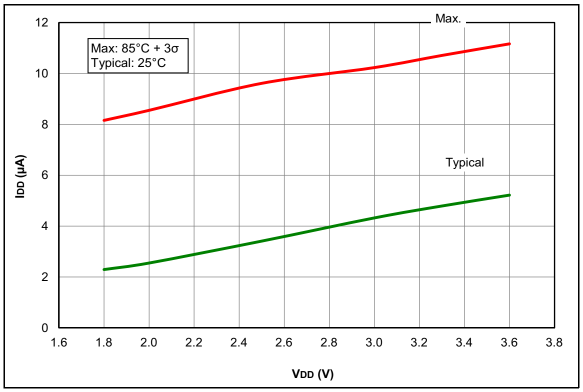


FIGURE 29-1: *IDD, EC Oscillator LP Mode, Fosc = 32 kHz, PIC12LF1612/16F1613 Only.*

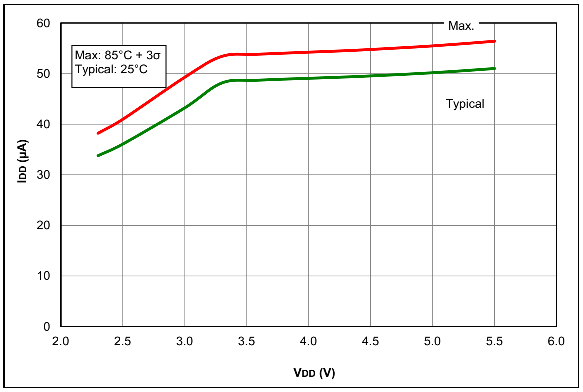


FIGURE 29-4: *IDD, EC Oscillator LP Mode, Fosc = 500 kHz, PIC12F1612/16F1613 Only.*

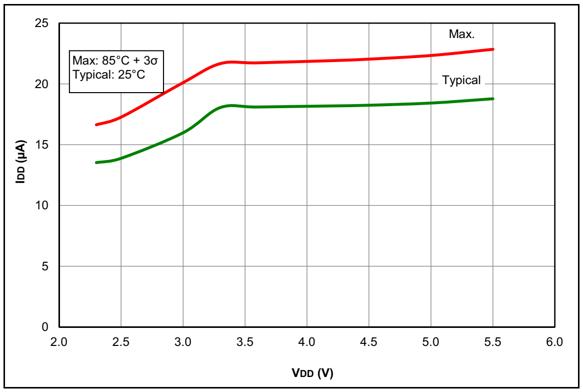


FIGURE 29-2: *IDD, EC Oscillator LP Mode, Fosc = 32 kHz, PIC12F1612/16F1613 Only.*

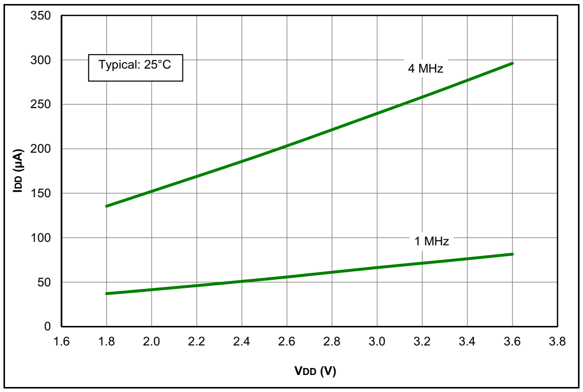


FIGURE 29-5: *IDD Typical, EC Oscillator MP Mode, PIC12LF1612/16F1613 Only.*

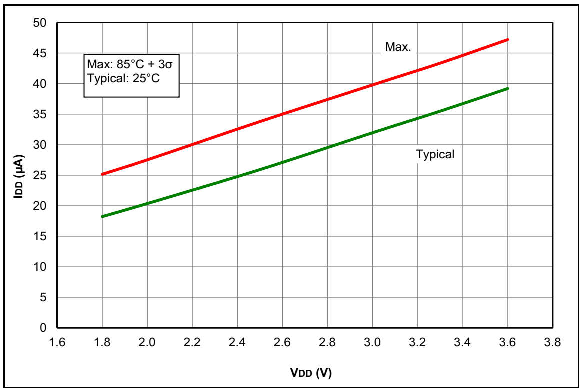


FIGURE 29-3: *IDD, EC Oscillator LP Mode, Fosc = 500 kHz, PIC12LF1612/16F1613 Only.*

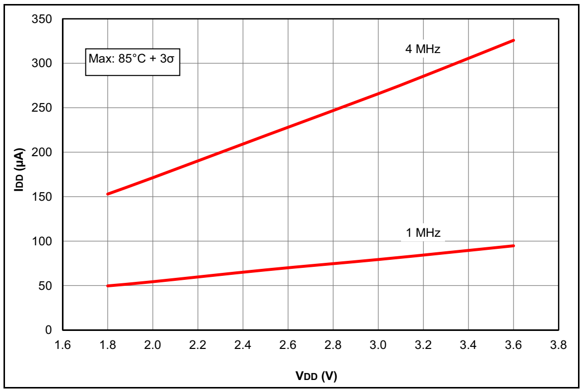


FIGURE 29-6: *IDD Maximum, EC Oscillator MP Mode, PIC12LF1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

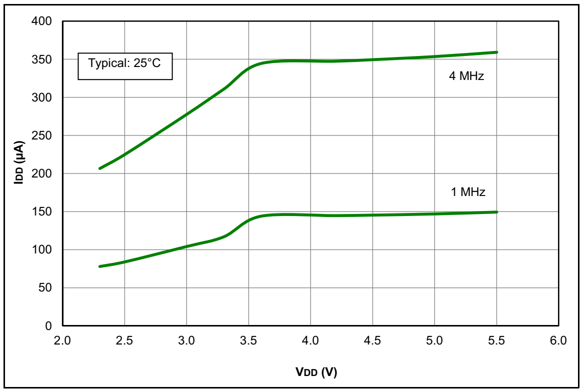


FIGURE 29-7: *IDD Typical, EC Oscillator MP Mode, PIC12F1612/16F1613 Only.*

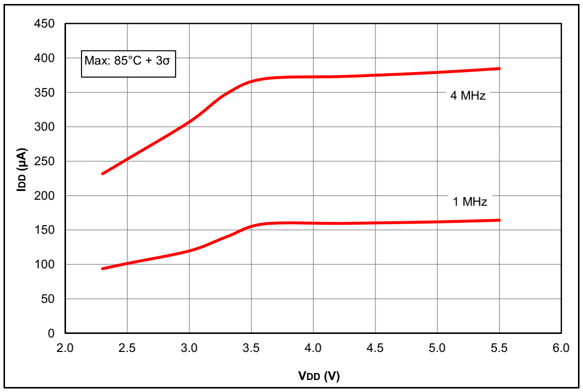


FIGURE 29-8: *IDD Maximum, EC Oscillator MP Mode, PIC12F1612/16F1613 Only.*

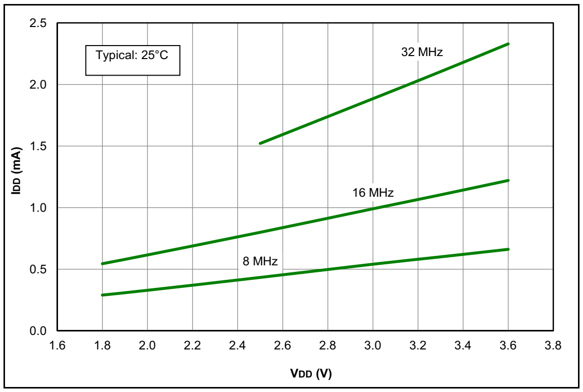


FIGURE 29-9: *IDD Typical, EC Oscillator HP Mode, PIC12LF1612/16F1613 Only.*

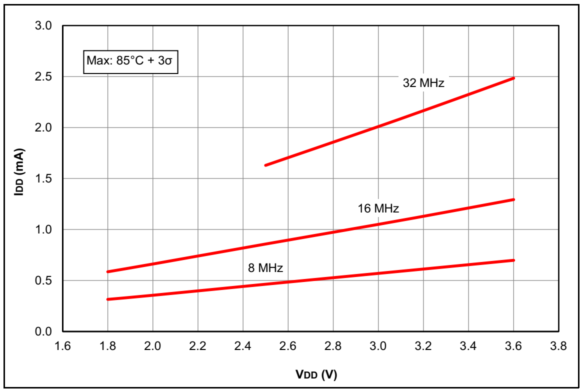


FIGURE 29-10: *IDD Maximum, EC Oscillator HP Mode, PIC12LF1612/16F1613 Only.*

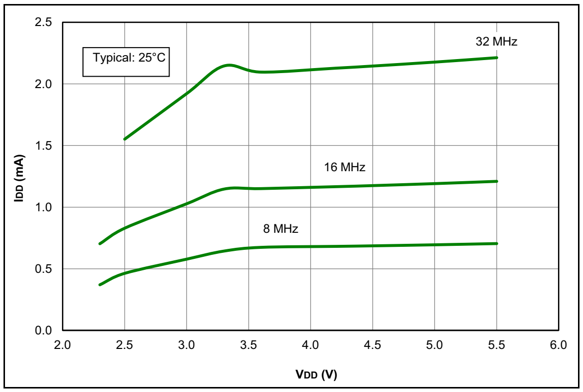


FIGURE 29-11: *IDD Typical, EC Oscillator HP Mode, PIC12F1612/16F1613 Only.*

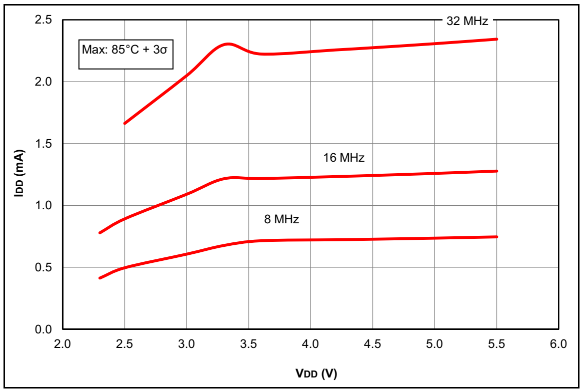


FIGURE 29-12: *IDD Maximum, EC Oscillator HP Mode, PIC12F1612/16F1613 Only.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

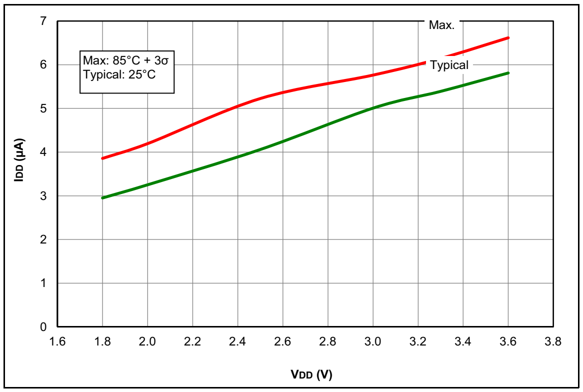


FIGURE 29-13: *I_{DD}, LFINTOSC Mode, F_{OSC} = 31 kHz, PIC12LF1612/16F1613 Only.*

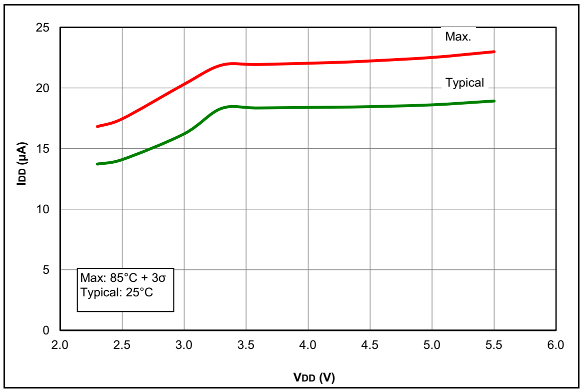


FIGURE 29-14: *I_{DD}, LFINTOSC Mode, F_{OSC} = 31 kHz, PIC12F1612/16F1613 Only.*

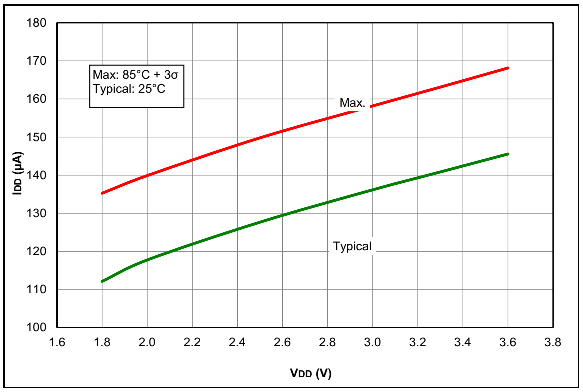


FIGURE 29-15: *I_{DD}, MFINTOSC Mode, F_{OSC} = 500 kHz, PIC12LF1612/16F1613 Only.*

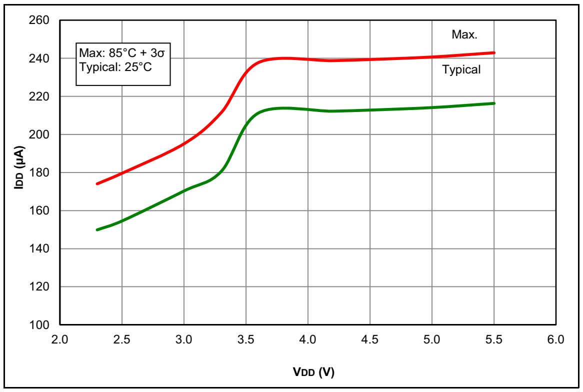


FIGURE 29-16: *I_{DD}, MFINTOSC Mode, F_{OSC} = 500 kHz, PIC12F1612/16F1613 Only.*

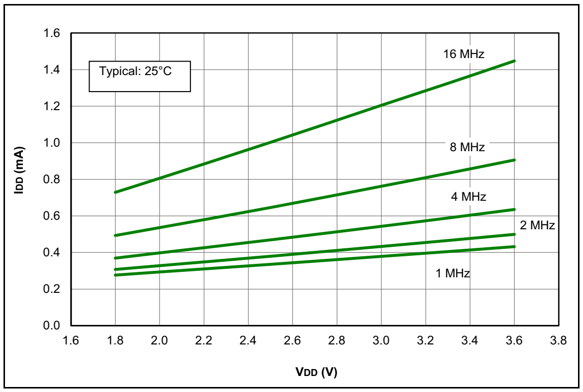


FIGURE 29-17: *I_{DD} Typical, HFINTOSC Mode, PIC12LF1612/16F1613 Only.*

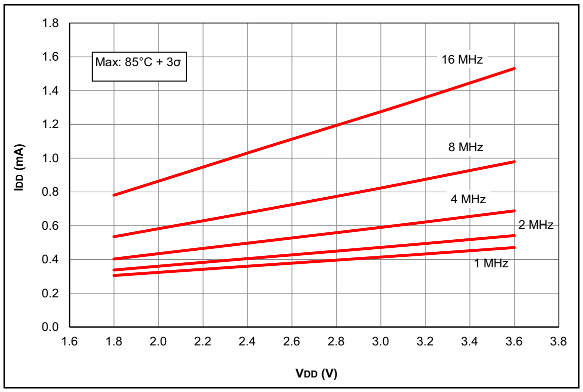


FIGURE 29-18: *I_{DD} Maximum, HFINTOSC Mode, PIC12LF1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

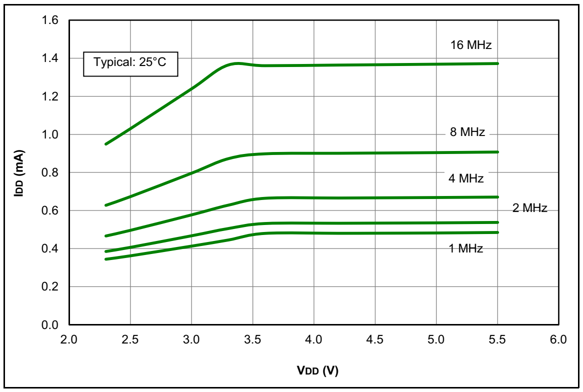


FIGURE 29-19: IDD Typical, HFINTOSC Mode, PIC12F1612/16F1613 Only.

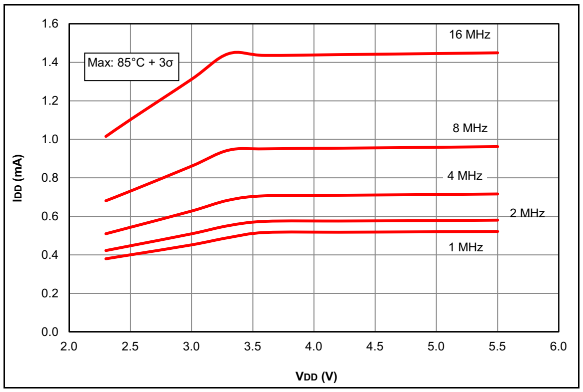


FIGURE 29-20: IDD Maximum, HFINTOSC Mode, PIC12F1612/16F1613 Only.

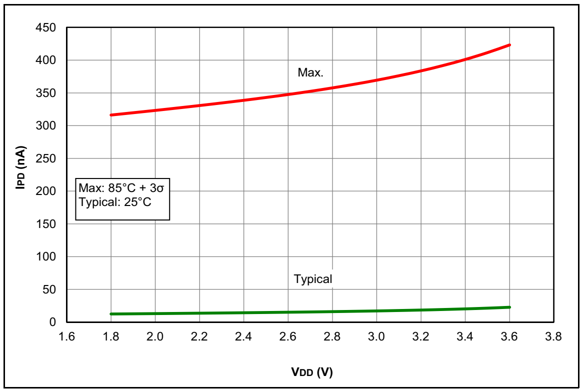


FIGURE 29-21: IPD Base, LP Sleep Mode, PIC12LF1612/16F1613 Only.

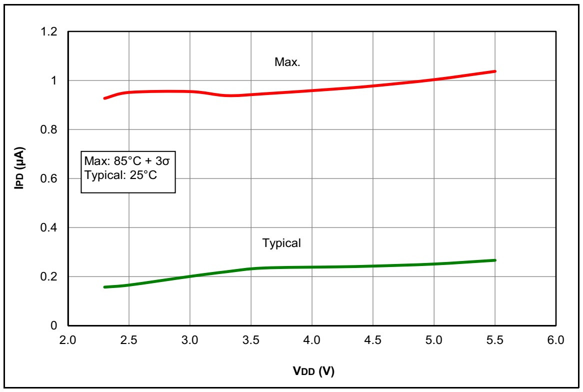


FIGURE 29-22: IPD Base, LP Sleep Mode (VREGPM = 1), PIC12F1612/16F1613 Only.

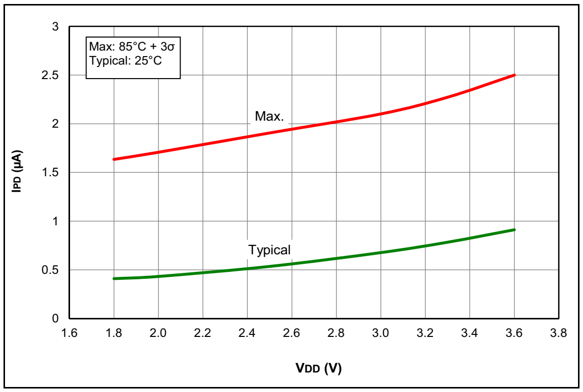


FIGURE 29-23: IPD, Watchdog Timer (WDT), PIC12LF1612/16F1613 Only.

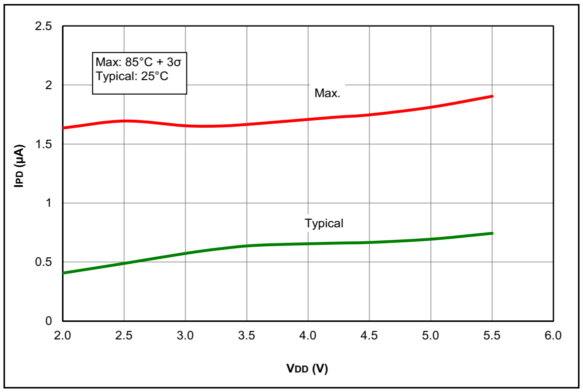


FIGURE 29-24: IPD, Watchdog Timer (WDT), PIC12F1612/16F1613 Only.

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

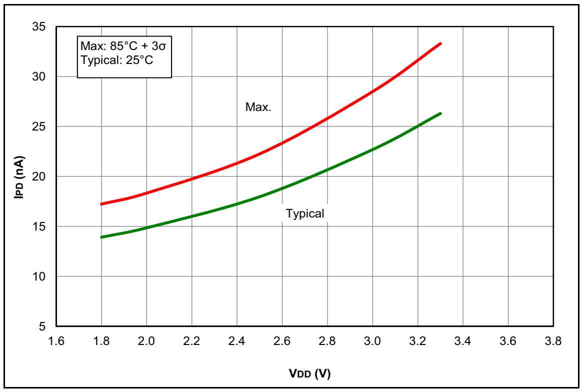


FIGURE 29-25: *I_{PD}, Fixed Voltage Reference (FVR), PIC12LF1612/16F1613 Only.*

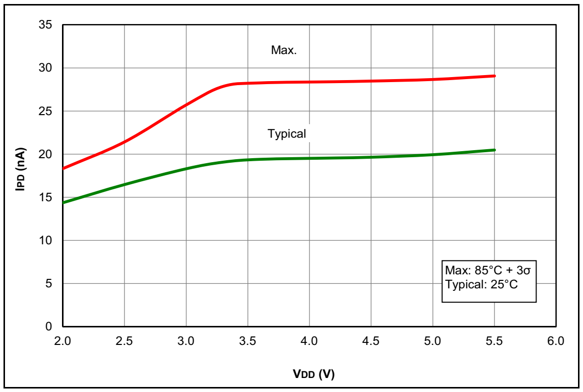


FIGURE 29-26: *I_{PD}, Fixed Voltage Reference (FVR), PIC12F1612/16F1613 Only.*

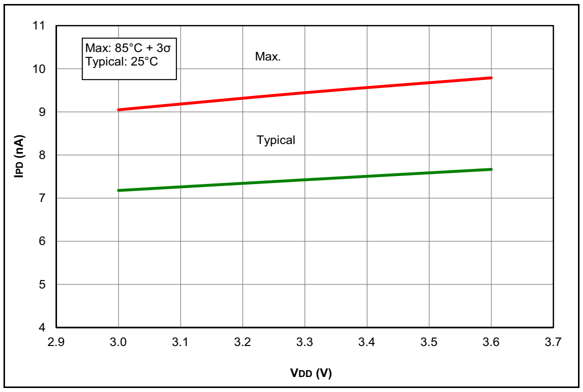


FIGURE 29-27: *I_{PD}, Brown-Out Reset (BOR), BORV = 1, PIC12LF1612/16F1613 Only.*

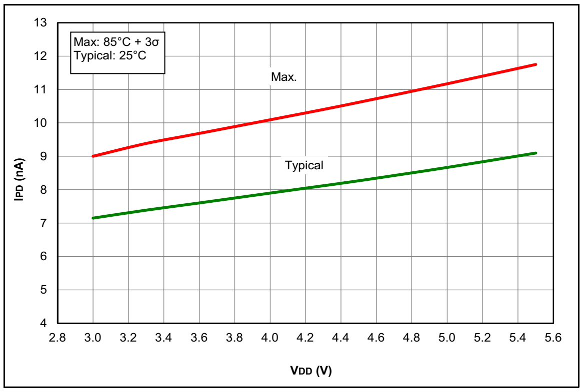


FIGURE 29-28: *I_{PD}, Brown-Out Reset (BOR), BORV = 1, PIC12F1612/16F1613 Only.*

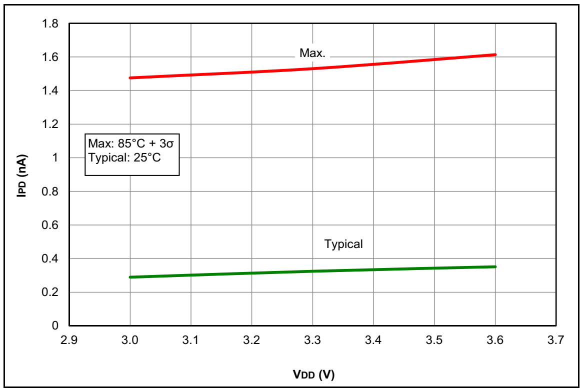


FIGURE 29-29: *I_{PD}, LP Brown-Out Reset (LPBOR = `0`), PIC12LF1612/16LF1613 Only.*

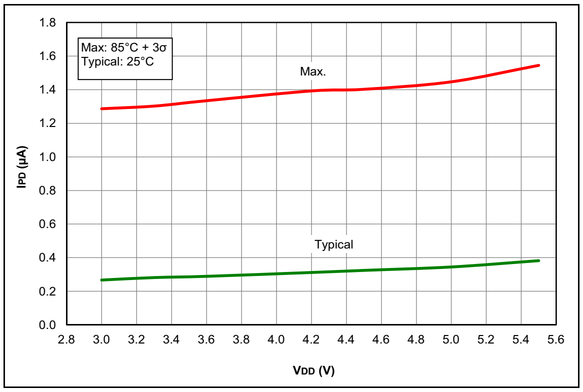


FIGURE 29-30: *I_{PD}, LP Brown-Out Reset (LPBOR = `0`), PIC12F1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 μF, TA = 25°C.

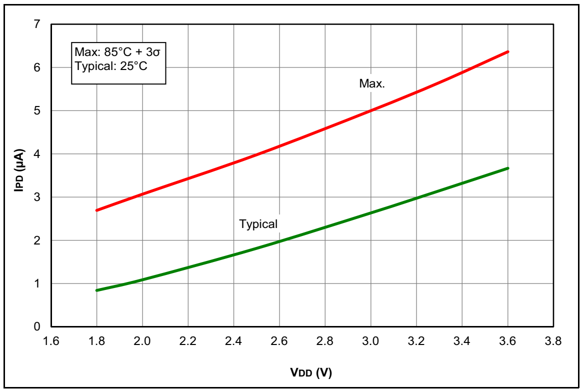


FIGURE 29-31: *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC12LF1612/16F1613 Only.*

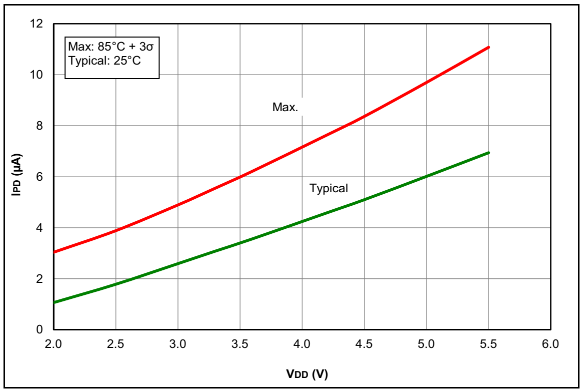


FIGURE 29-32: *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC12F1612/16F1613 Only.*

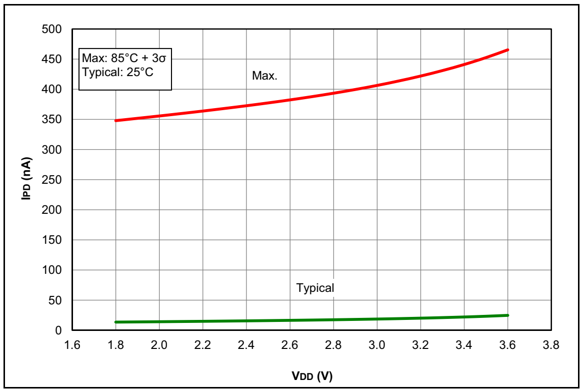


FIGURE 29-33: *IPD, ADC Non-Converting, PIC12LF1612/16F1613 Only.*

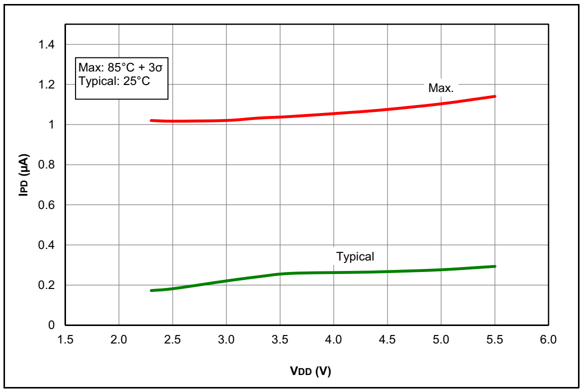


FIGURE 29-34: *IPD, ADC Non-Converting, PIC12F1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

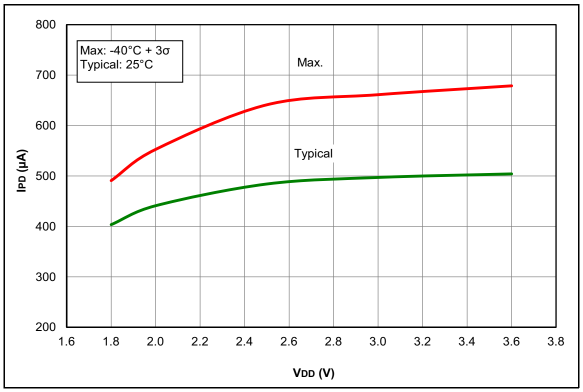


FIGURE 29-35: *IPD, Comparator, NP Mode (CxSP = 1), PIC12LF1612/16F1613 Only.*

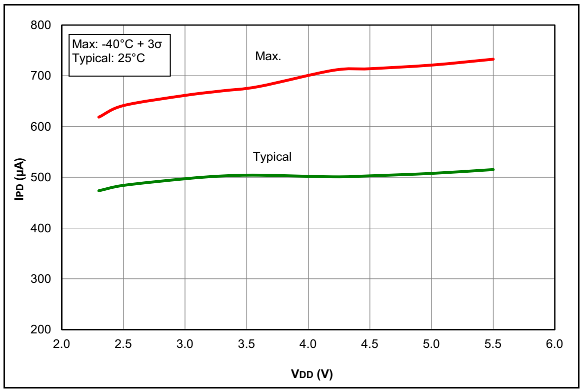


FIGURE 29-36: *IPD, Comparator, NP Mode (CxSP = 1), PIC12F1612/16F1613 Only.*

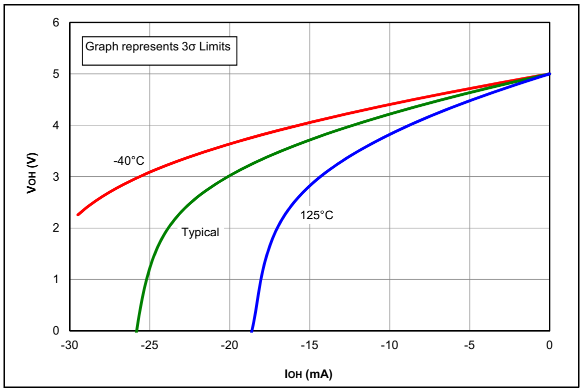


FIGURE 29-37: *VOH vs. IOH Over Temperature, VDD = 5.0V, PIC12F1612/16F1613 Only.*

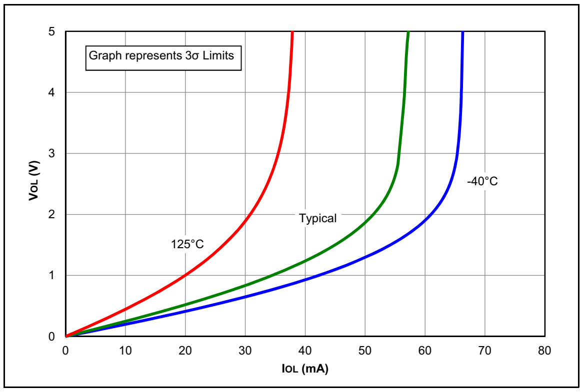


FIGURE 29-38: *VOL vs. IOL Over Temperature, VDD = 5.0V, PIC12F1612/16F1613 Only.*

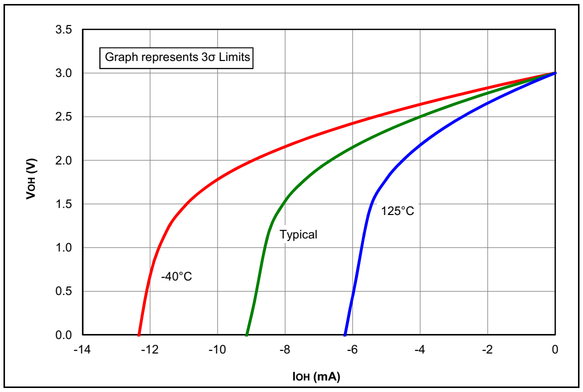


FIGURE 29-39: *VOH vs. IOH Over Temperature, VDD = 3.0V.*

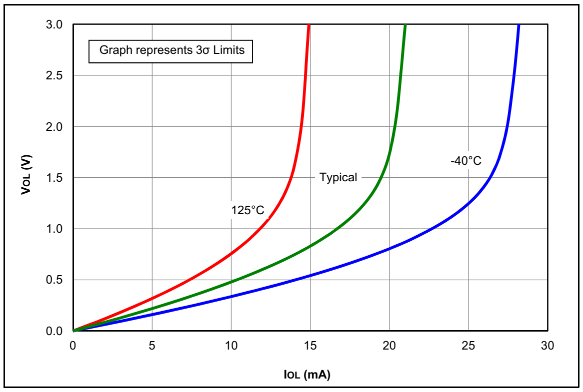


FIGURE 29-40: *VOL vs. IOL Over Temperature, VDD = 3.0V.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

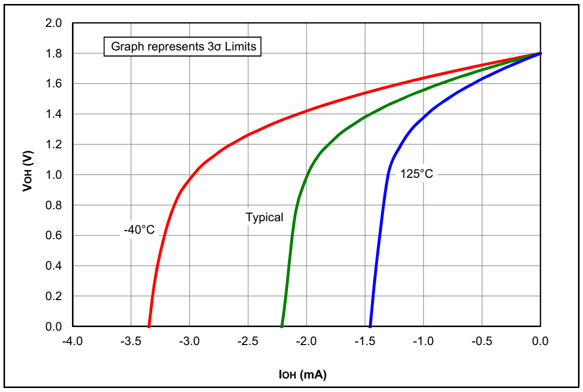


FIGURE 29-41: *V_{OH} vs. I_{OH} Over Temperature, V_{DD} = 1.8V, PIC12LF1612/16F1613 Only.*

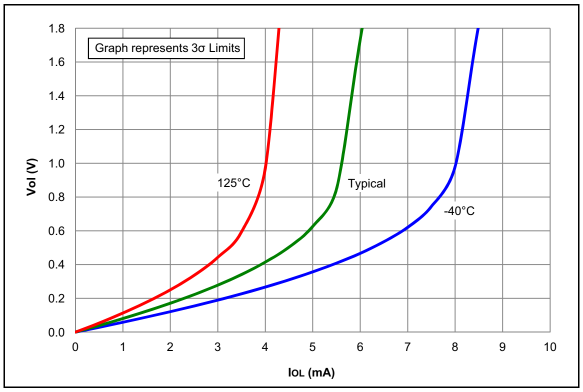


FIGURE 29-42: *V_{OL} vs. I_{OL} Over Temperature, V_{DD} = 1.8V, PIC12LF1612/16F1613 Only.*

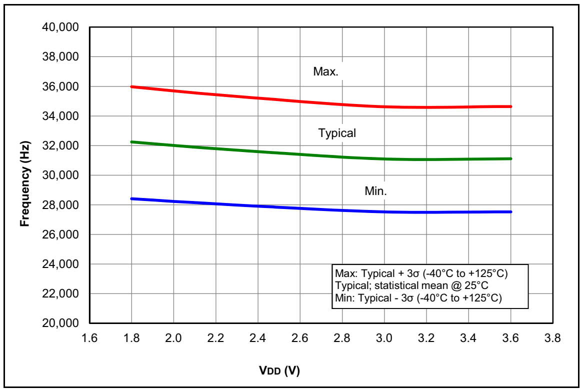


FIGURE 29-43: *LFINTOSC Frequency, PIC12LF1612/16F1613 Only.*

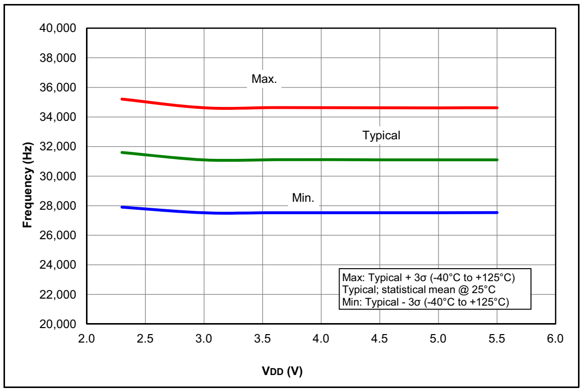


FIGURE 29-44: *LFINTOSC Frequency, PIC12F1612/16F1613 Only.*

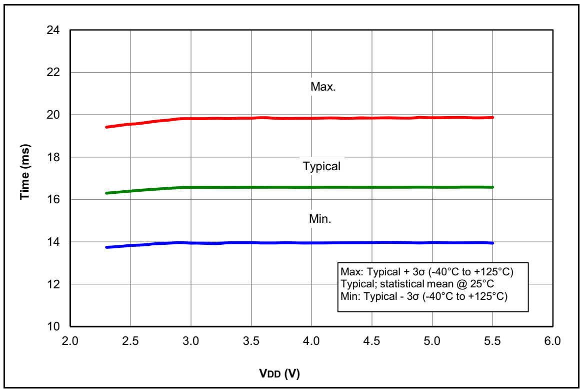


FIGURE 29-45: *WDT Time-Out Period, PIC12F1612/16F1613 Only.*

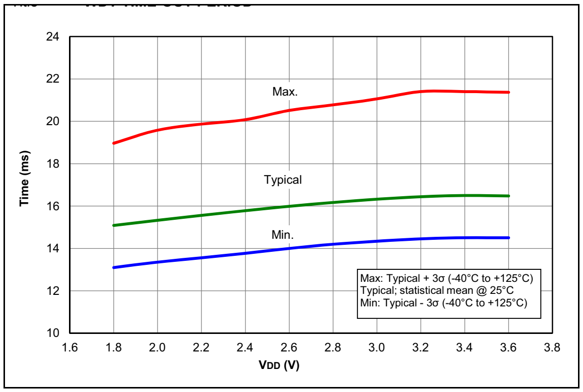


FIGURE 29-46: *WDT Time-Out Period, PIC12LF1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

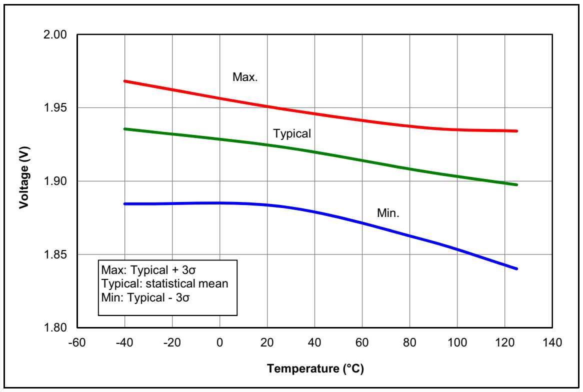


FIGURE 29-47: *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC12LF1612/16F1613 Only.*

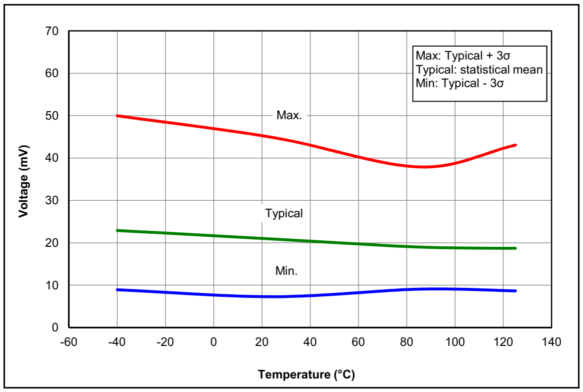


FIGURE 29-48: *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC12LF1612/16F1613 Only.*

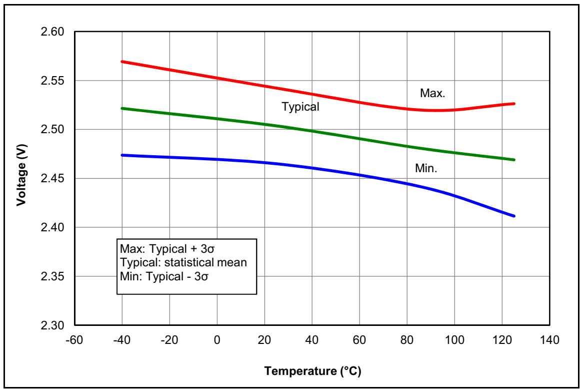


FIGURE 29-49: *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC12F1612/16F1613 Only.*

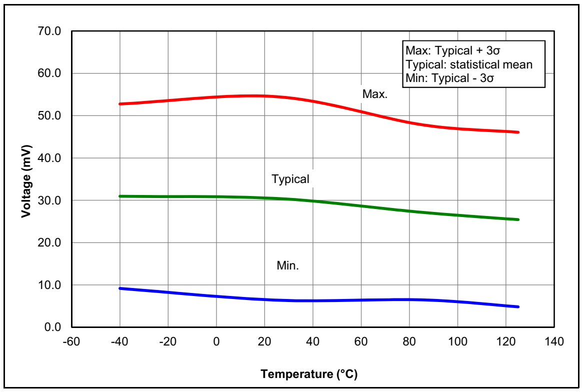


FIGURE 29-50: *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC12F1612/16F1613 Only.*

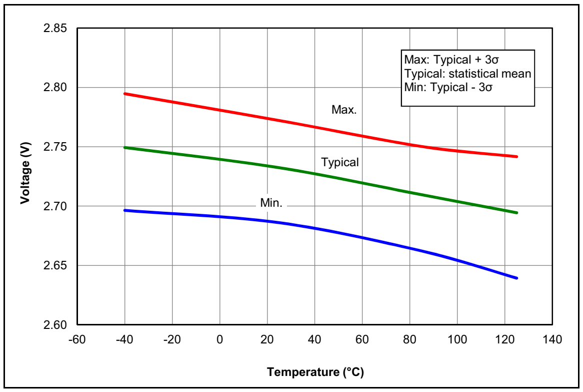


FIGURE 29-51: *Brown-Out Reset Voltage, High Trip Point (BORV =* `0`*).*

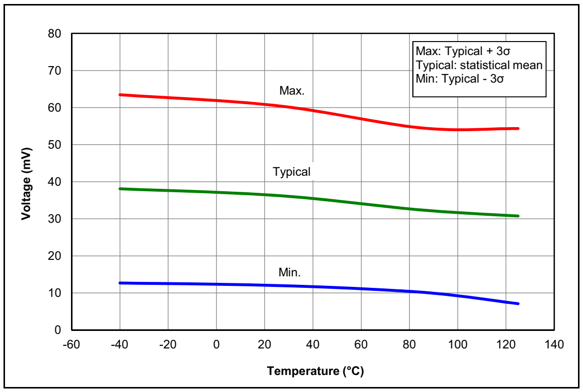


FIGURE 29-52: *Brown-Out Reset Hysteresis, High Trip Point (BORV =* `0`*).*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

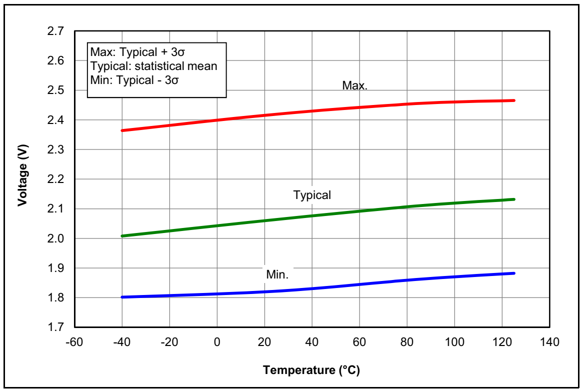


FIGURE 29-53: *LPBOR Reset Voltage.*

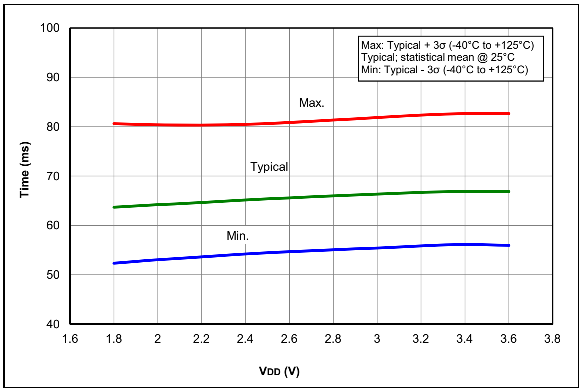


FIGURE 29-56: *PWRT Period, PIC12LF1612/16F1613 Only.*

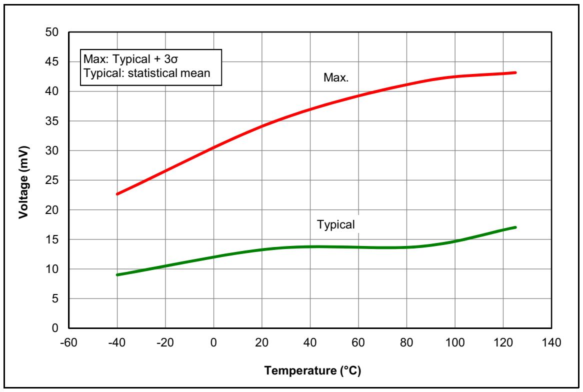


FIGURE 29-54: *LPBOR Reset Hysteresis.*

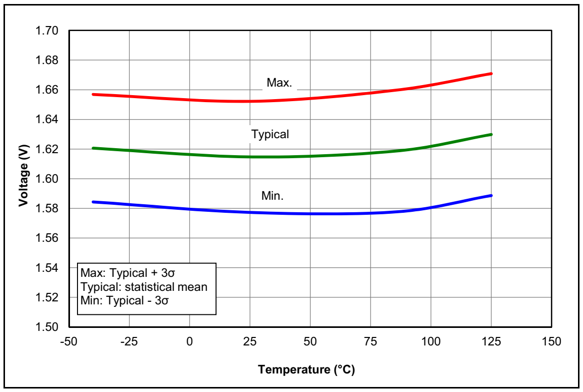


FIGURE 29-57: *POR Release Voltage.*

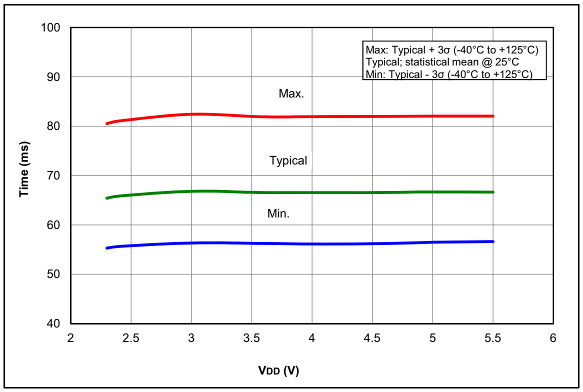


FIGURE 29-55: *PWRT Period, PIC12F1612/16F1613 Only.*

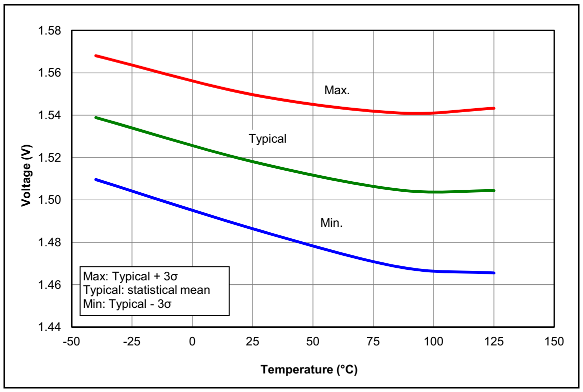


FIGURE 29-58: *POR Rearm Voltage, NP Mode (VREGPM1 = `0`), PIC12F1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 μF, TA = 25°C.

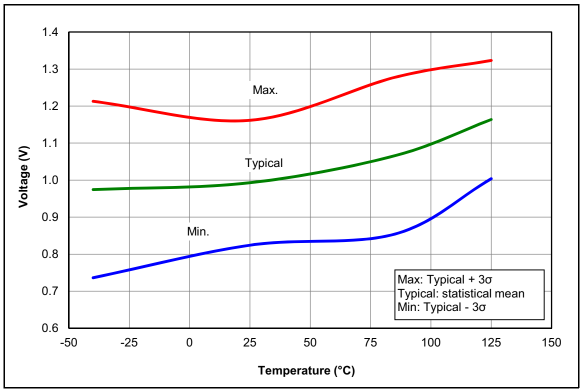


FIGURE 29-59: *POR Rearm Voltage, NP Mode, PIC12LF1612/16F1613 Only.*

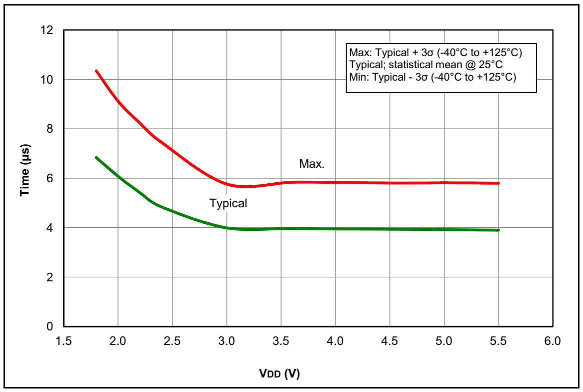


FIGURE 29-60: *Wake From Sleep, VREGPM = 0.*

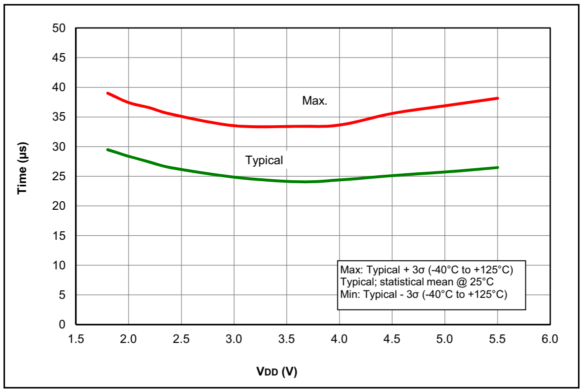


FIGURE 29-61: *Wake From Sleep, VREGPM = 1.*

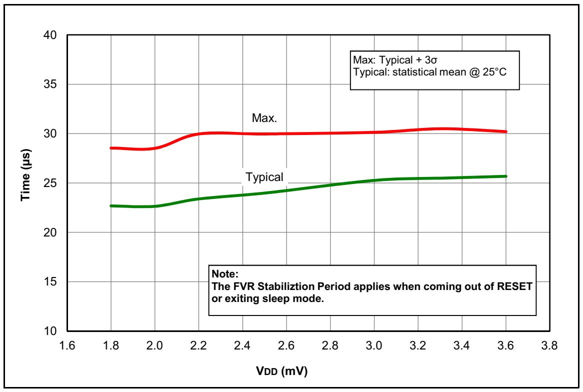


FIGURE 29-62: *FVR Stabilization Period, PIC12LF1612/16F1613 Only.*

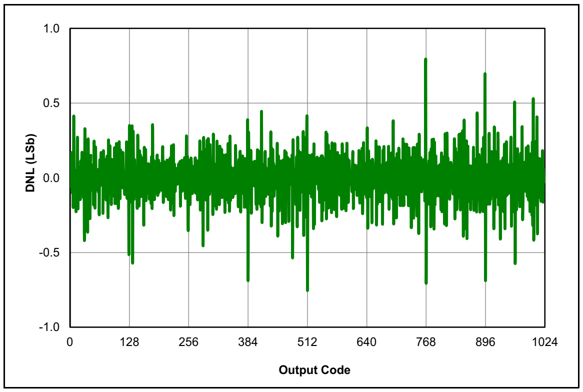


FIGURE 29-63: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS, 25°C.*

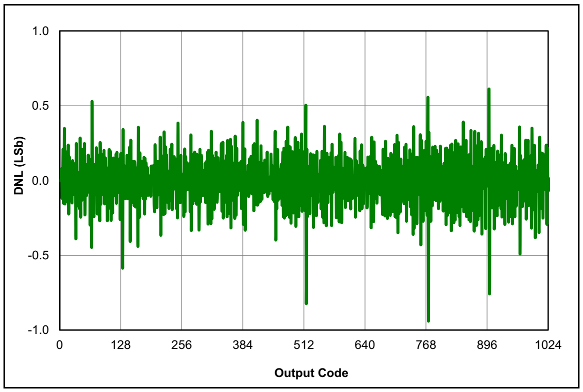


FIGURE 29-64: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 4 µS, 25°C.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

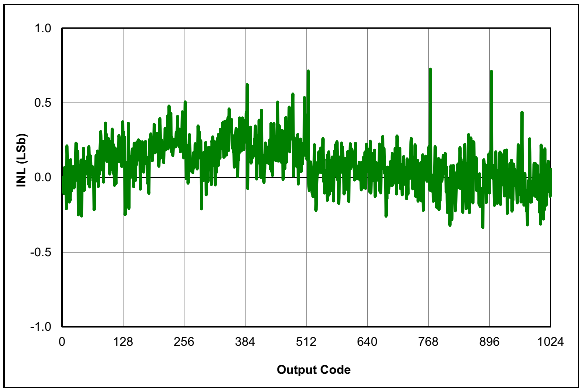


FIGURE 29-65: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS, 25°C.*

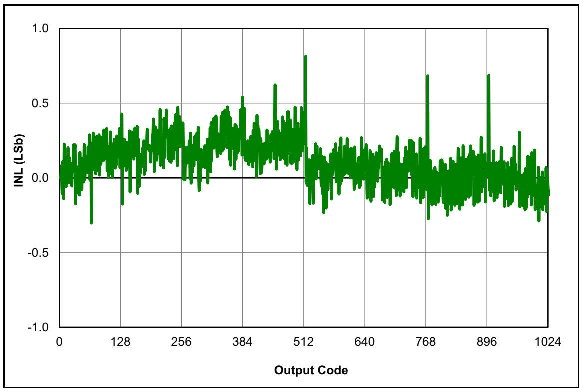


FIGURE 29-66: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4 µS, 25°C.*

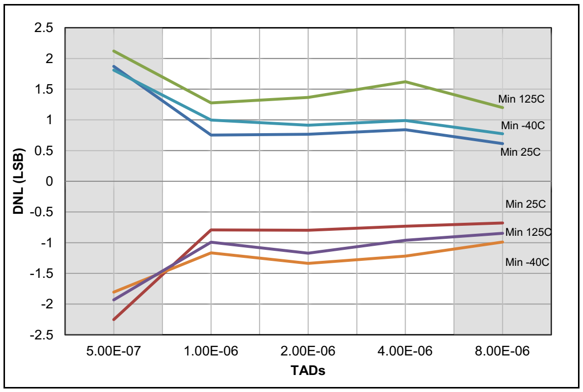


FIGURE 29-67: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.*

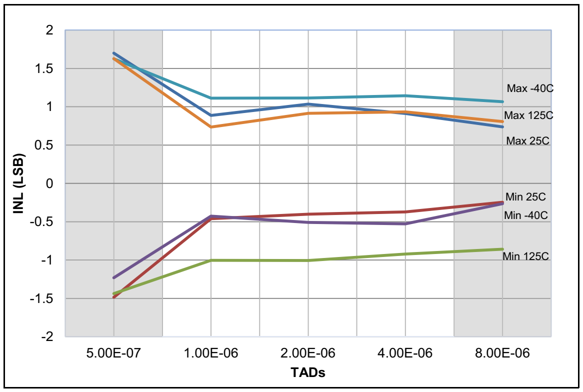


FIGURE 29-68: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, VREF = 3.0V.*

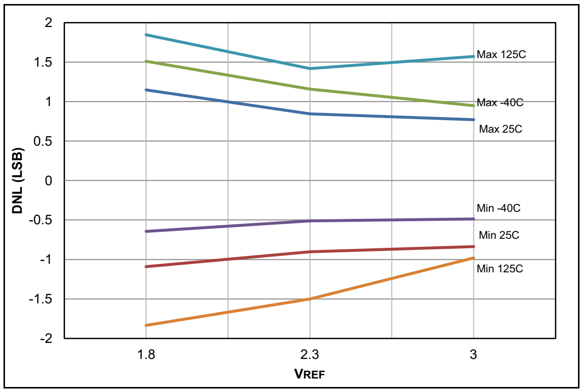


FIGURE 29-69: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS.*

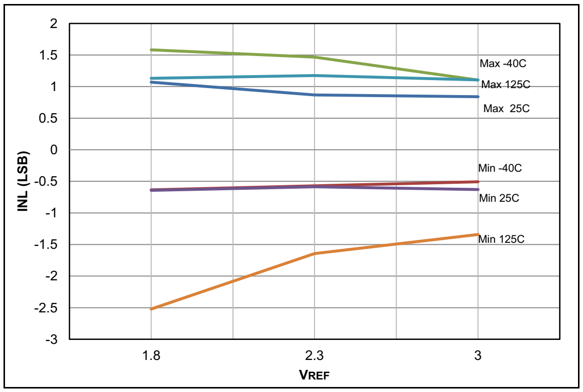


FIGURE 29-70: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

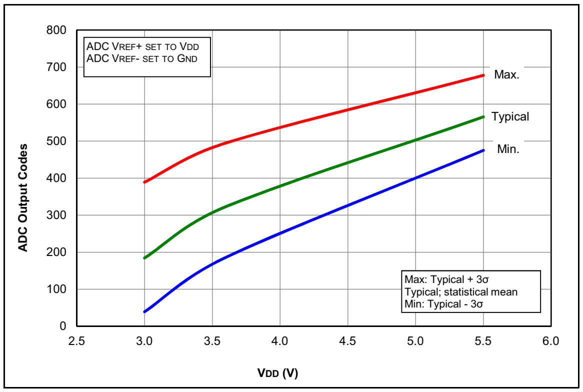


FIGURE 29-71: *Temp. Indicator Initial Offset, High Range, Temp. = 20°C, PIC12F1612/16F1613 Only.*

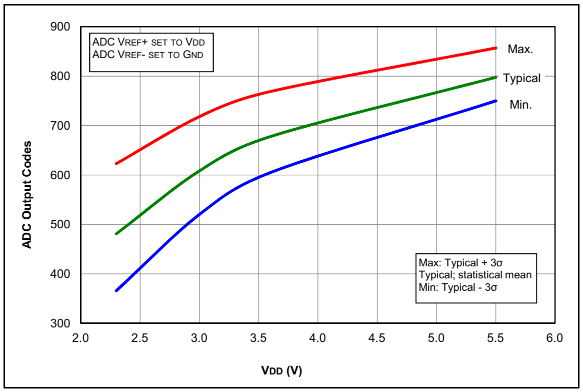


FIGURE 29-72: *Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC12F1612/16F1613 Only.*

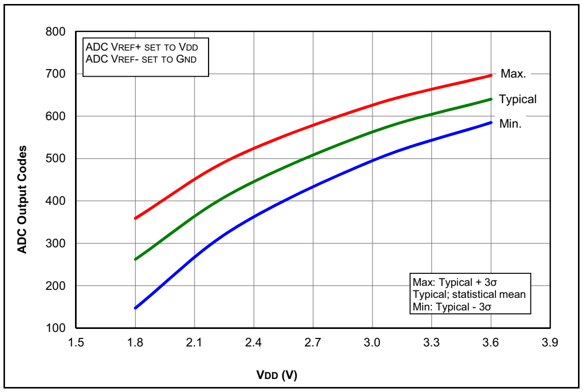


FIGURE 29-73: *Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC12LF1612/16F1613 Only.*

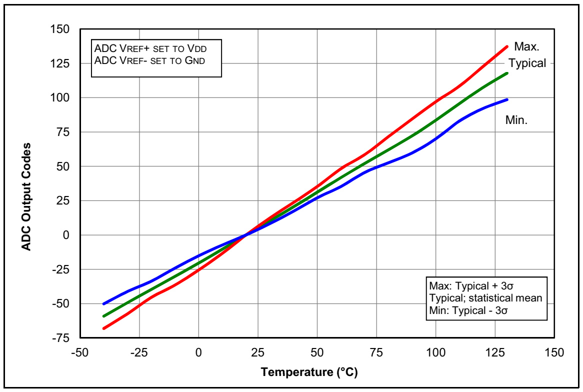


FIGURE 29-74: *Temp. Indicator Slope Normalized to 20°C, High Range, V_{DD} = 5.5V, PIC12F1612/16F1613 Only.*

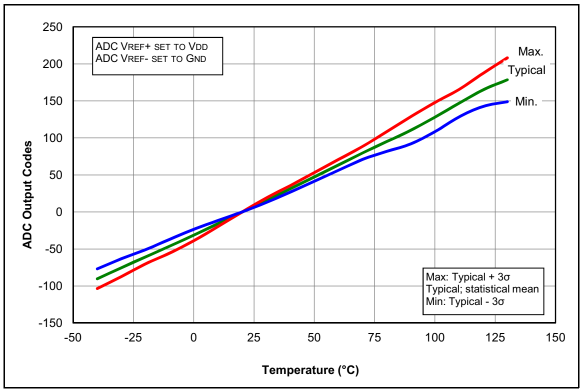


FIGURE 29-75: *Temp. Indicator Slope Normalized to 20°C, High Range, V_{DD} = 3.0V, PIC12F1612/16F1613 Only.*

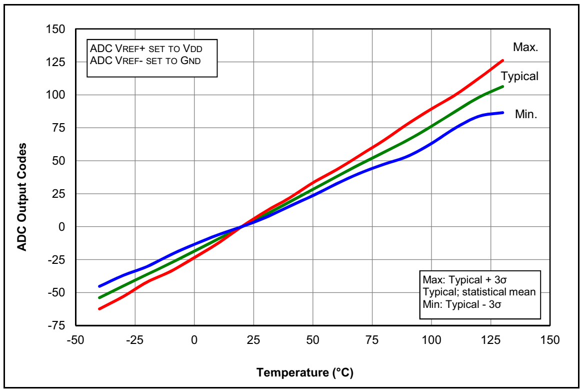


FIGURE 29-76: *Temp. Indicator Slope Normalized to 20°C, Low Range, V_{DD} = 3.0V, PIC12F1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

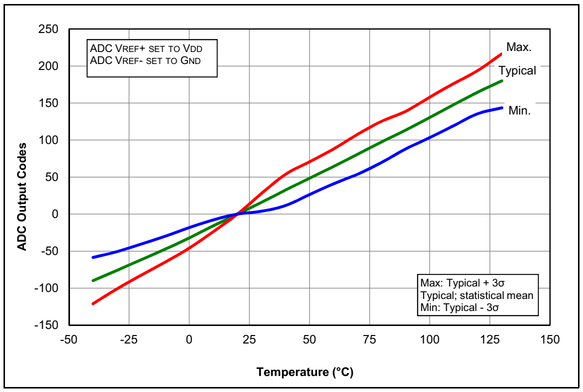


FIGURE 29-77: *Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 1.8V, PIC12LF1612/16F1613 Only.*

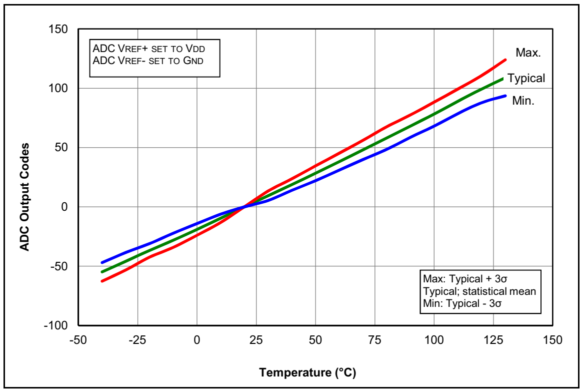


FIGURE 29-78: *Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0V, PIC12LF1612/16F1613 Only.*

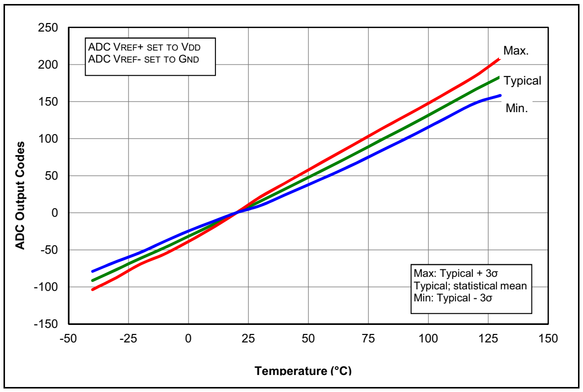


FIGURE 29-79: *Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 3.6V, PIC12LF1612/16F1613 Only.*

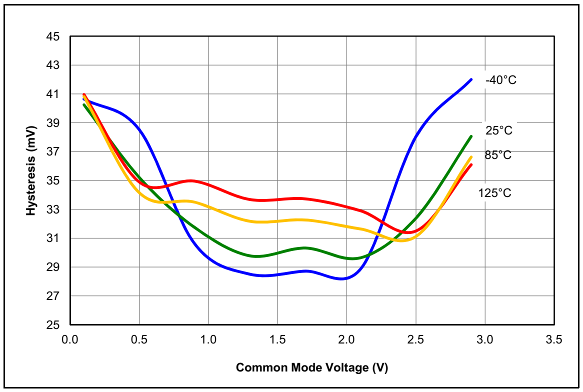


FIGURE 29-80: *Comparator Hysteresis, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values.*

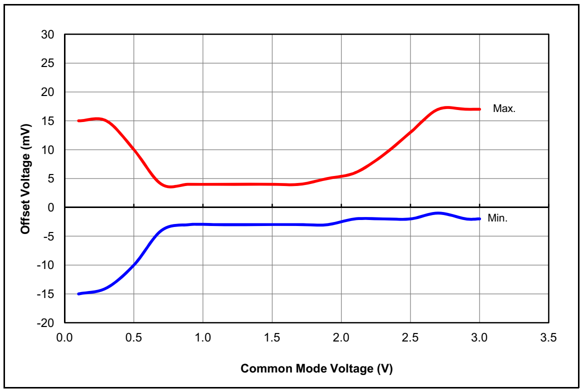


FIGURE 29-81: *Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values at 25°C.*

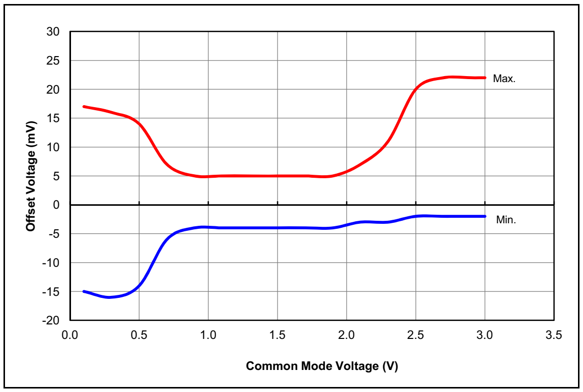


FIGURE 29-82: *Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values From -40°C to 125°C.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

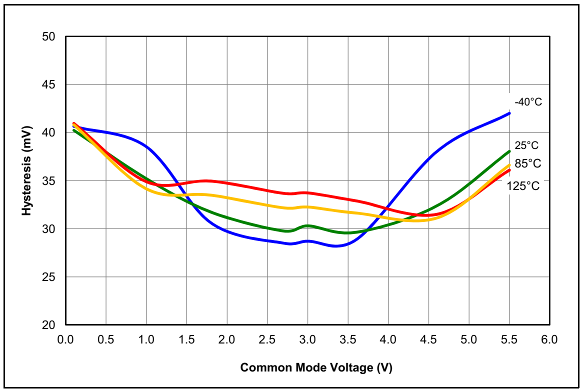


FIGURE 29-83: *Comparator Hysteresis, NP Mode (CxSP = 1), VDD = 5.5V, Typical Measured Values, PIC12F1612/16F1613 Only.*

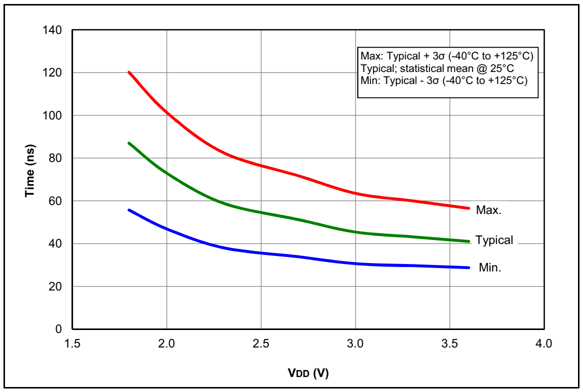


FIGURE 29-86: *Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values, PIC12LF1612/16F1613 Only.*

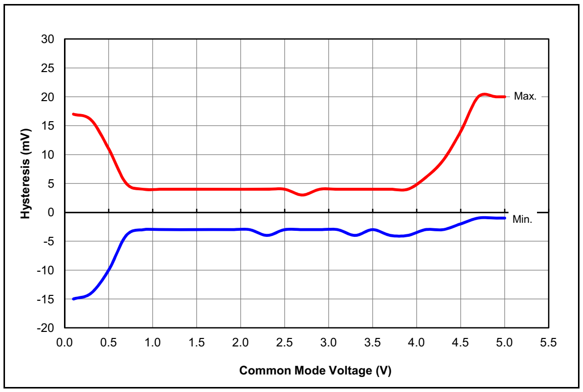


FIGURE 29-84: *Comparator Offset, NP Mode (CxSP = 1), VDD = 5.0V, Typical Measured Values at 25°C, PIC12F1612/16F1613 Only.*

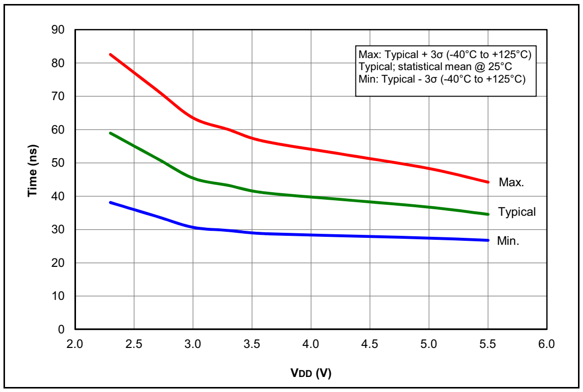


FIGURE 29-87: *Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values, PIC12F1612/16F1613 Only.*

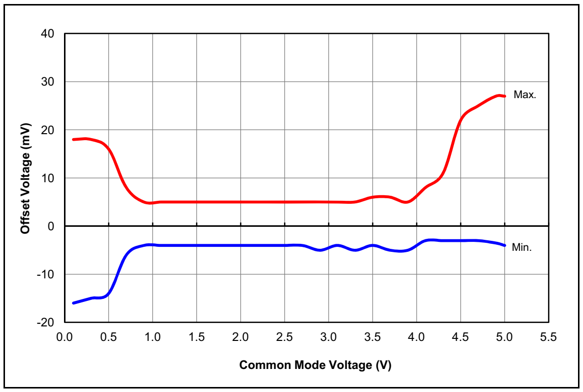


FIGURE 29-85: *Comparator Offset, NP Mode (CxSP = 1), VDD = 5.5V, Typical Measured Values From -40°C to 125°C, PIC12F1612/16F1613 Only.*

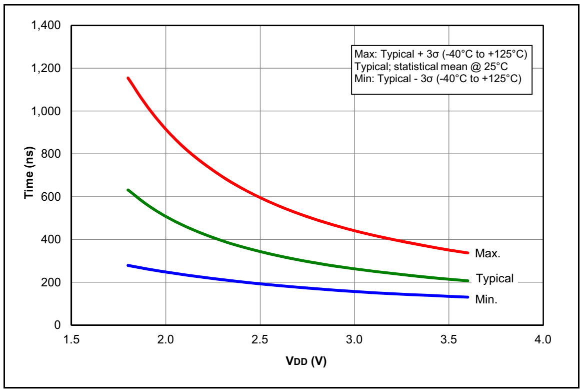


FIGURE 29-88: *Comparator Output Filter Delay Time Over Temp., NP Mode (CxSP = 1), Typical Measured Values, PIC12LF1612/16F1613 Only.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

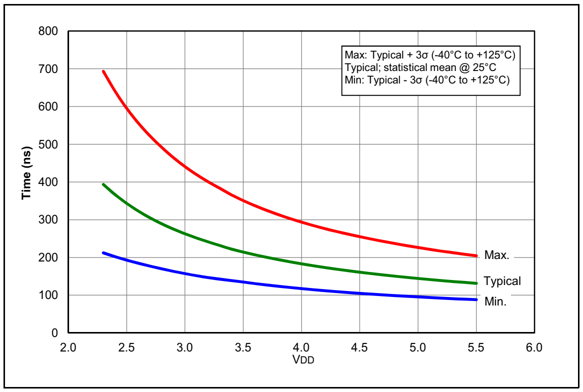


FIGURE 29-89: *Comparator Output Filter Delay Time Over Temp., NP Mode (CxSP = 1), Typical Measured Values, PIC12F1612/16F1613 Only.*

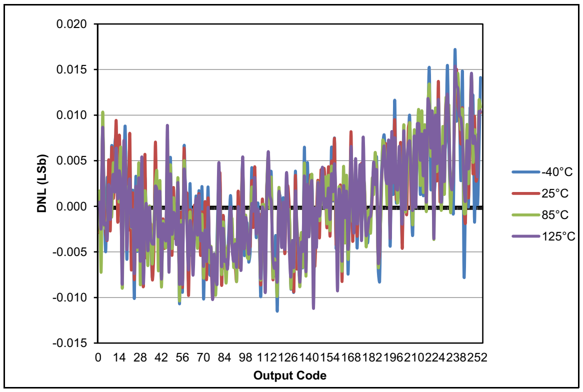


FIGURE 29-92: *Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC12F1612/16F1613 Only.*

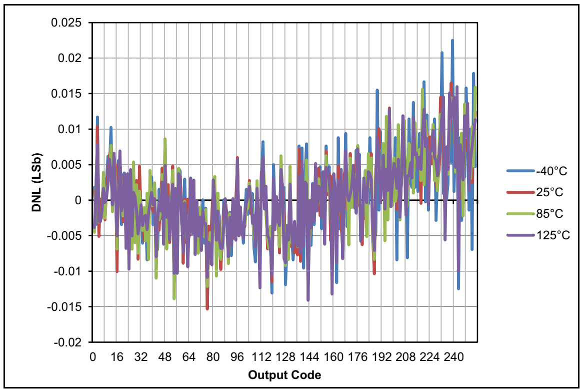


FIGURE 29-90: *Typical DAC DNL Error, VDD = 3.0V, VREF = External 3V.*

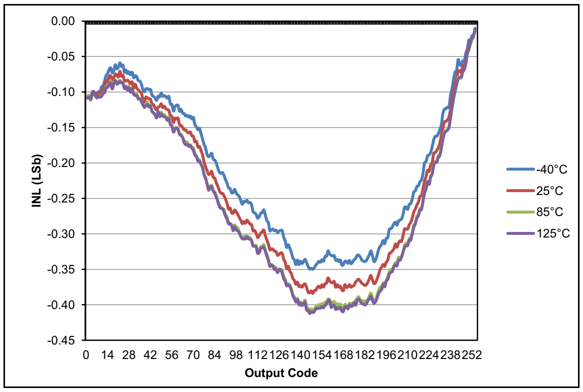


FIGURE 29-93: *Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC12F1612/16F1613 Only.*

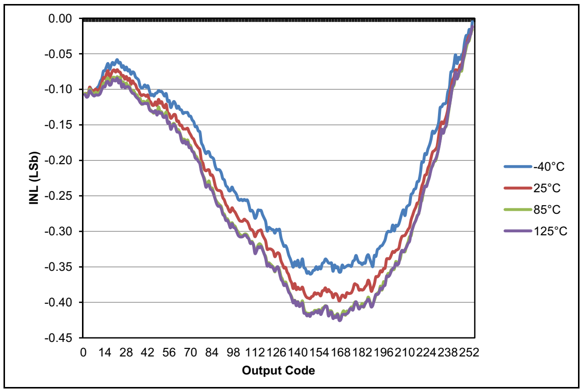


FIGURE 29-91: *Typical DAC INL Error, VDD = 3.0V, VREF = External 3V.*

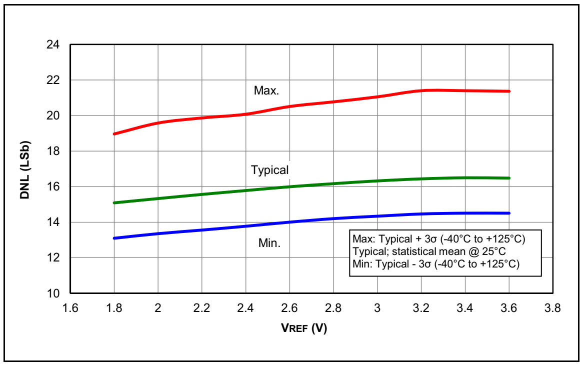


FIGURE 29-94: *DAC INL Error, VDD = 3.0V.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

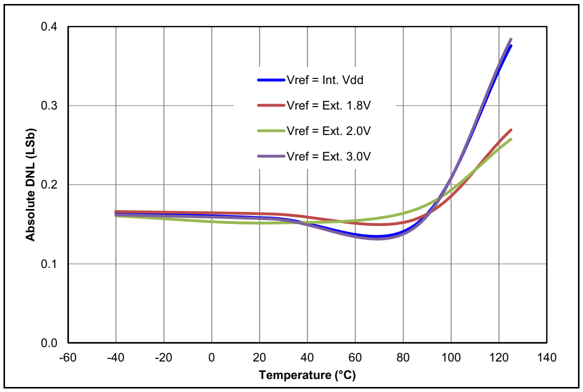


FIGURE 29-95: *Absolute Value of DAC DNL Error, V_{DD} = 3.0V, V_{REF} = V_{DD}.*

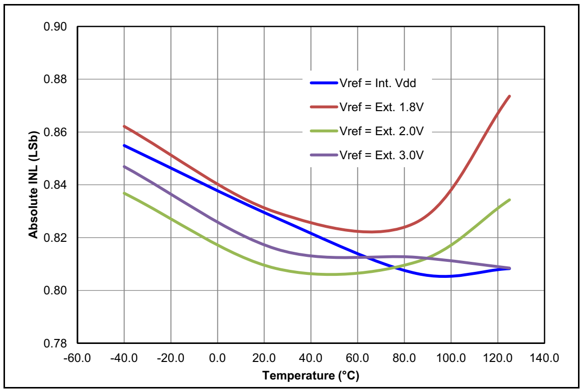


FIGURE 29-96: *Absolute Value of DAC INL Error, V_{DD} = 3.0V, V_{REF} = V_{DD}.*

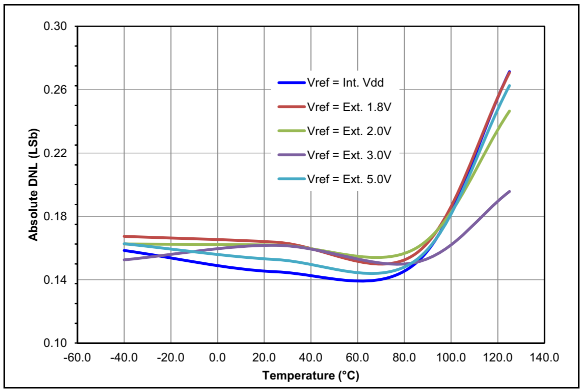


FIGURE 29-97: *Absolute Value of DAC DNL Error, V_{DD} = 5.0V, V_{REF} = V_{DD}, PIC12F1612/16F1613 Only.*

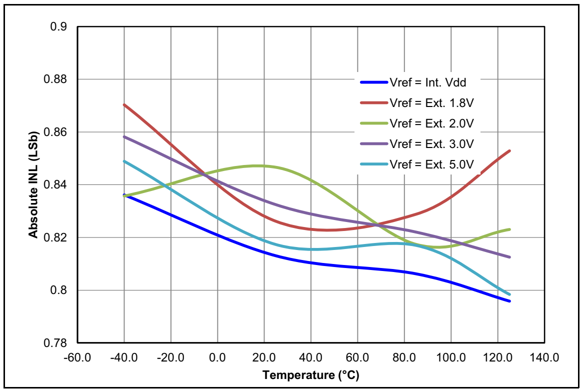


FIGURE 29-98: *Absolute Value of DAC INL Error, V_{DD} = 5.0V, V_{REF} = V_{DD}, PIC12F1612/16F1613 Only.*

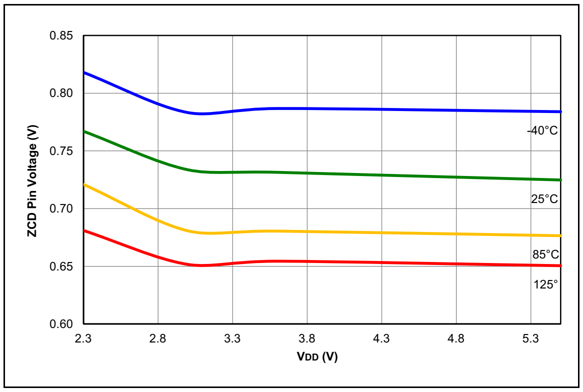


FIGURE 29-99: *ZCD Pin Voltage, Typical Measured Values.*

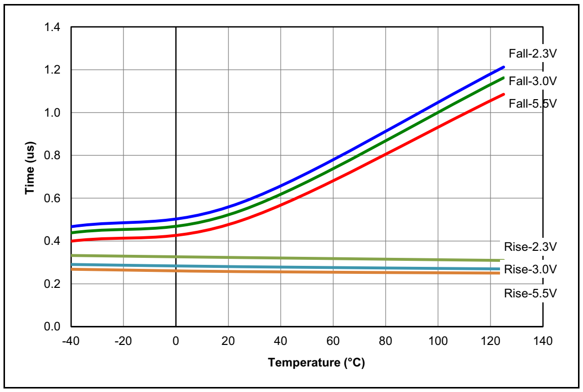


FIGURE 29-100: *ZCD Response time Over Voltage, Typical Measured Values.*

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

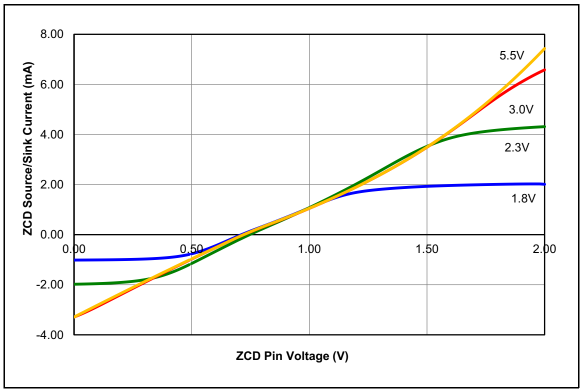


FIGURE 29-101: *ZCD Pin Current Over ZCD Pin Voltage, Typical Measured Values from -40°C to 125°C.*

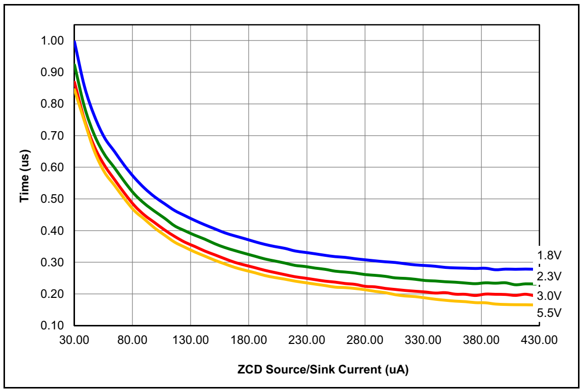


FIGURE 29-102: *ZCD Pin Response Time Over Current, Typical Measured Values from -40°C to 125°C.*

30.0 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

30.1 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

30.2 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

30.3 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

30.4 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

30.5 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

30.6 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.

30.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

30.8 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.

30.9 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™).

30.10 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.

30.11 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.

30.12 Third-Party Development Tools

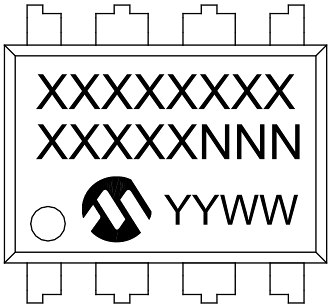
Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- 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®

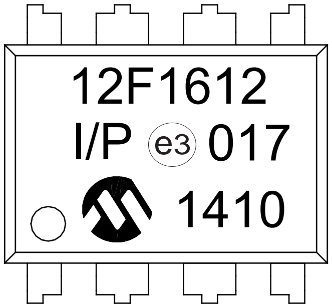
31.0 PACKAGING INFORMATION

31.1 Package Marking Information

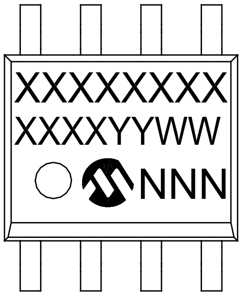
8-Lead PDIP (300 mil)



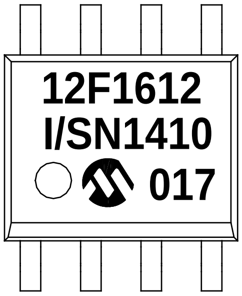
Example



8-Lead SOIC (3.90 mm)



Example



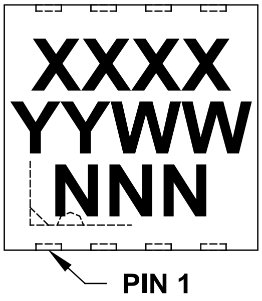
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free 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.

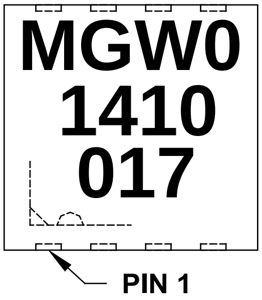
* Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

31.1 Package Marking Information (Continued)

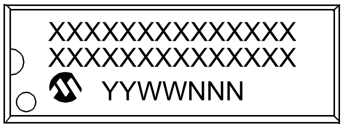
8-Lead DFN (3x3x0.9 mm)
8-Lead UDFN (3x3x0.5 mm)



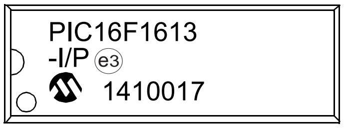
Example



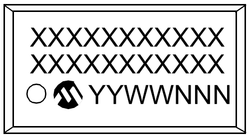
14-Lead PDIP



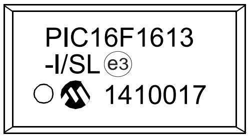
Example



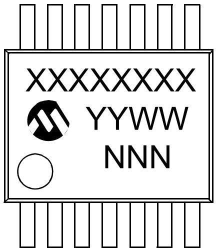
14-Lead SOIC (.150”)



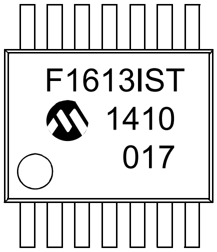
Example



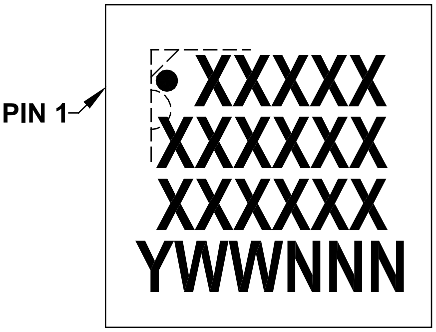
14-Lead TSSOP



Example



16-Lead QFN (4x4x0.9 mm)



Example

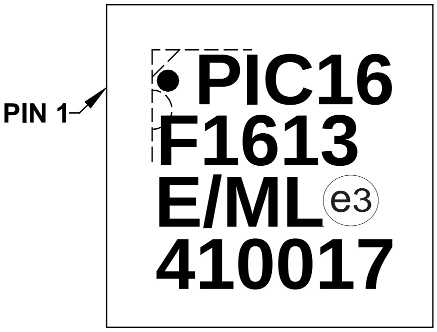


TABLE 31-1: 8-LEAD 3x3 DFN (MF) TOP MARKING

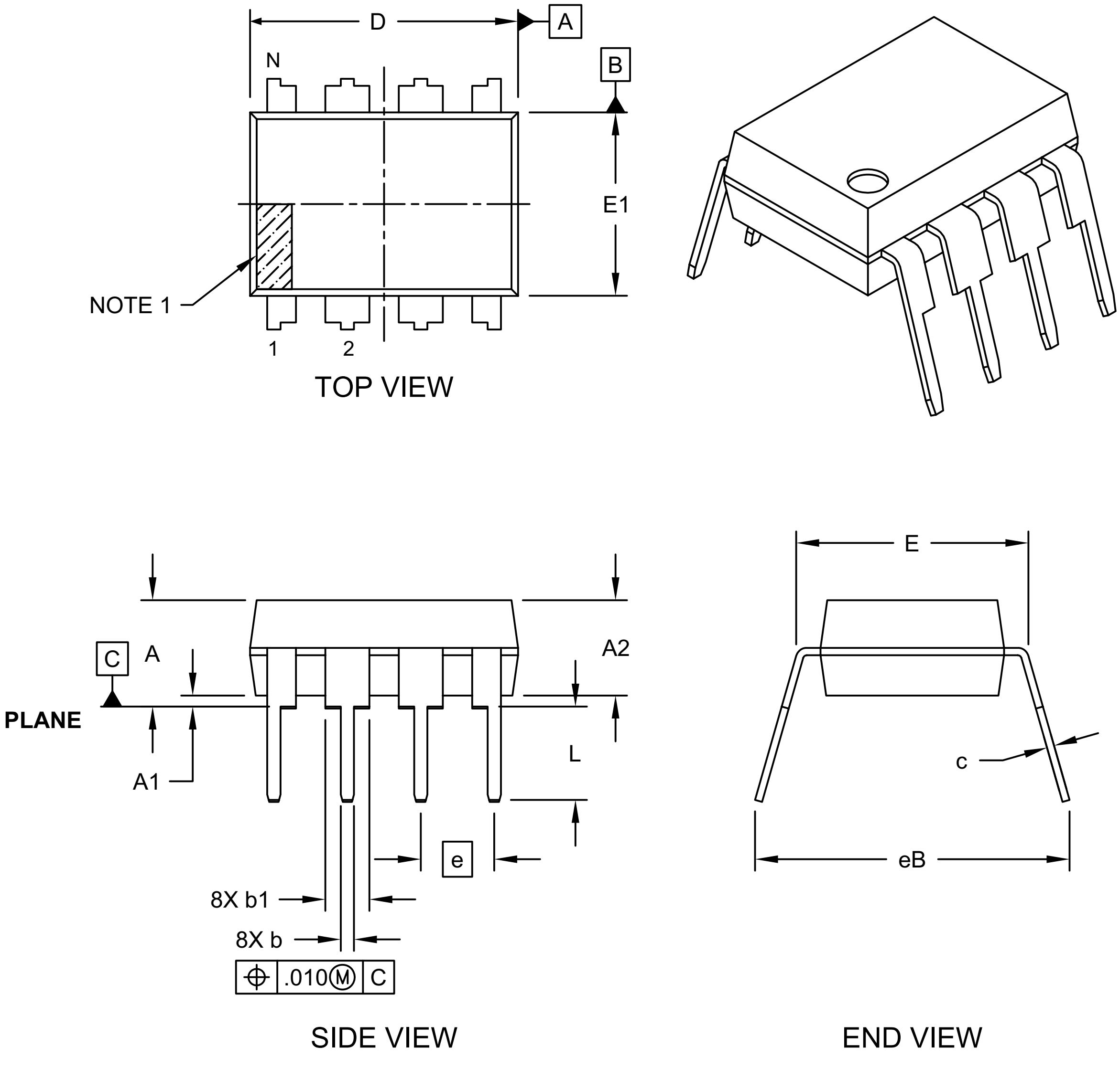
Part Number	Marking
PIC12F1612-E/MF	MGU0
PIC12LF1612-E/MF	MGW0
PIC12F1612-I/MF	MGV0
PIC12LF1612-I/MF	MGX0
PIC12F1612T-I/MF	MGV0
PIC12LF1612T-I/MF	MGX0

31.2 Package Details

The following sections give the technical details of the packages.

8-Lead Plastic Dual In-Line (P) - 300 mil Body [PDIP]

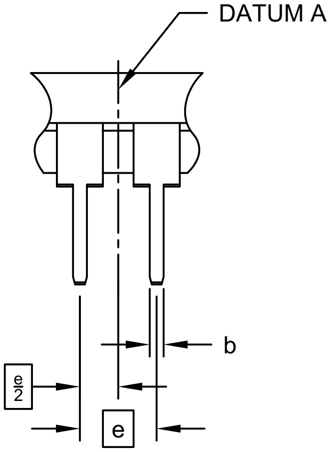
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

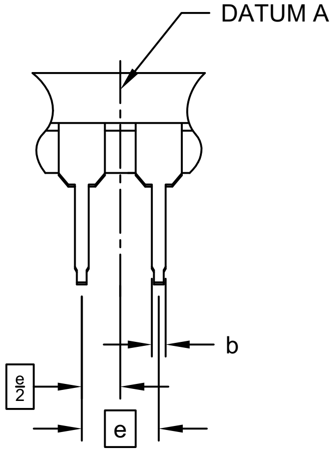


Microchip Technology Drawing No. C04-018D Sheet 1 of 2

8-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





ALTERNATE LEAD DESIGN
(VENDOR DEPENDENT)

	Units	INCHES		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N	8		
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	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	-	-	.430

Notes:

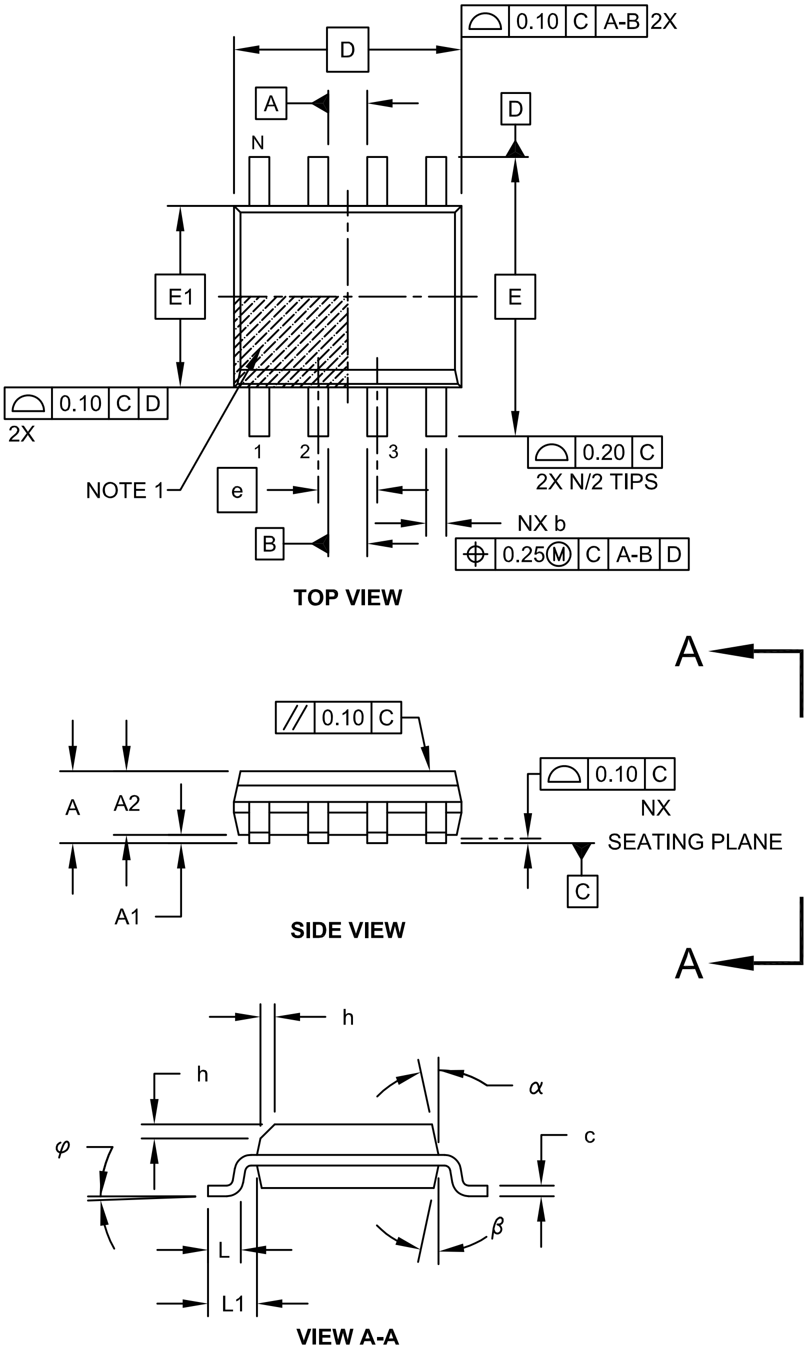
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-018D Sheet 2 of 2

8-Lead Plastic Small Outline (SN) - Narrow, 3.90 mm Body [SOIC]

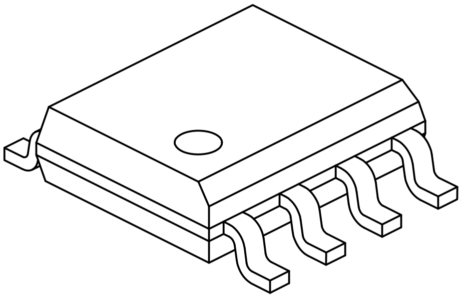
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing No. C04-057C Sheet 1 of 2

8-Lead Plastic Small Outline (SN) - 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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
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	4.90 BSC		
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1	1.04 REF		
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.17	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

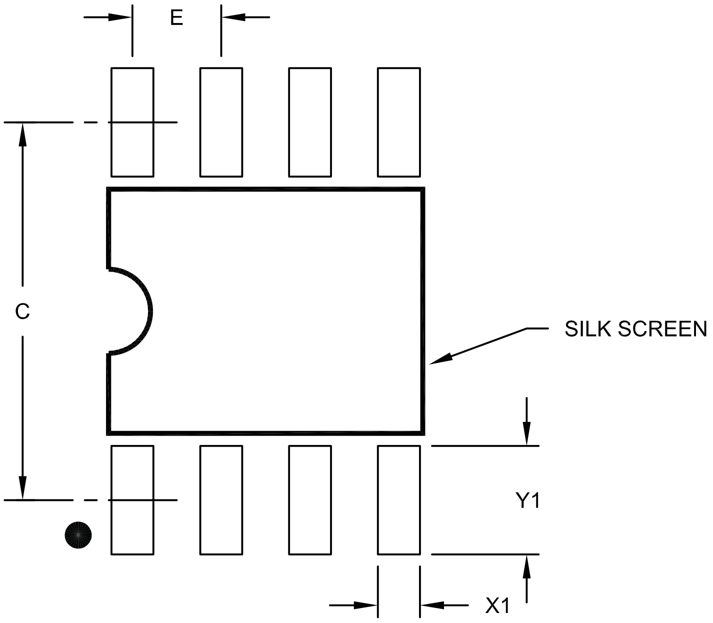
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm 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.

Microchip Technology Drawing No. C04-057C Sheet 2 of 2

8-Lead Plastic Small Outline (SN) – 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 (X8)	X1			0.60
Contact Pad Length (X8)	Y1			1.55

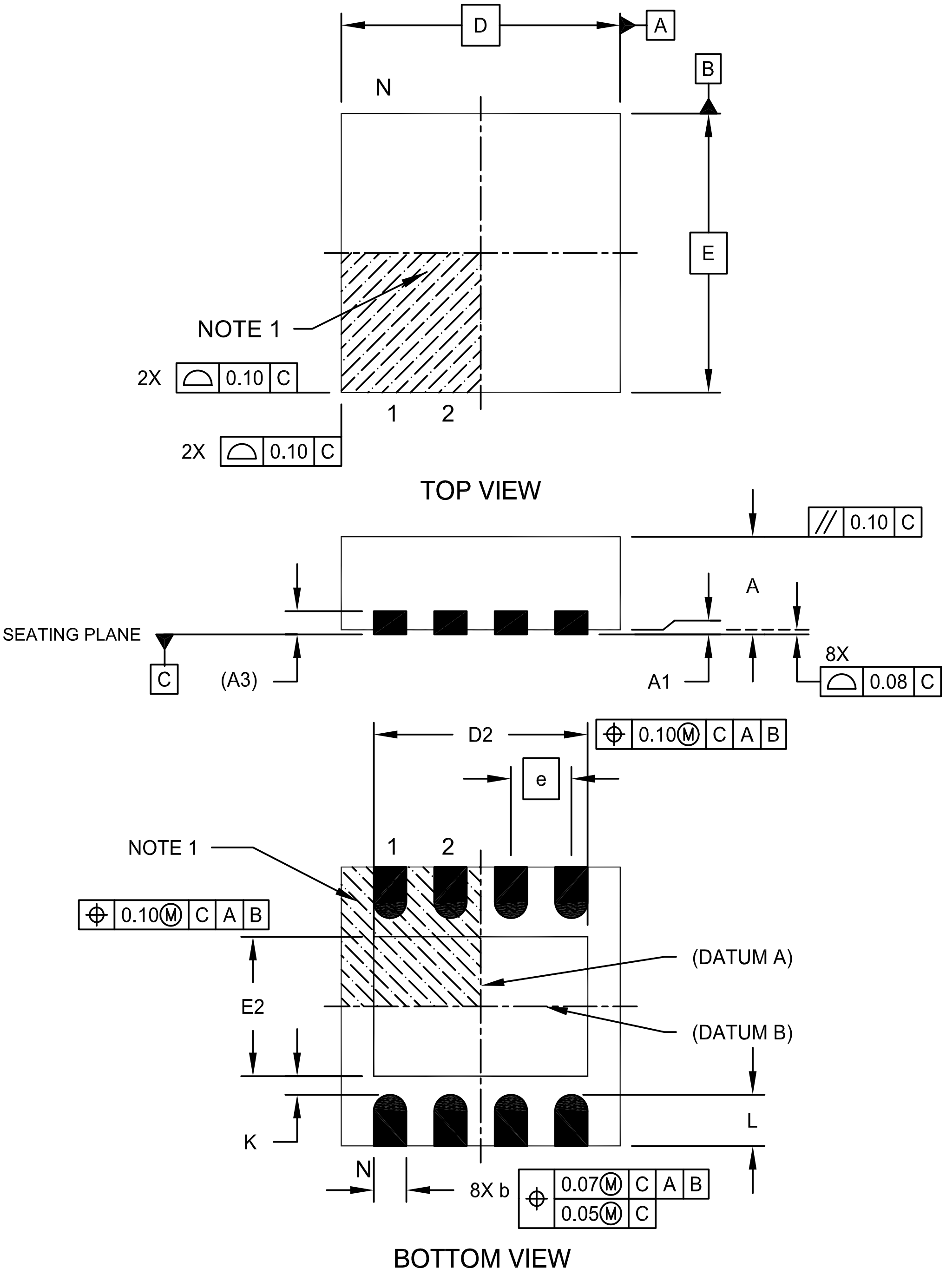
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2057A

8-Lead Plastic Dual Flat, No Lead Package (MF) - 3x3x0.9mm Body [DFN]

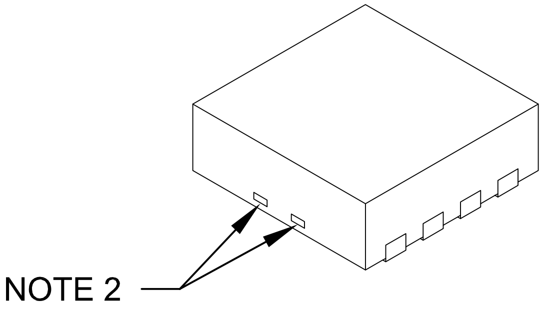
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing No. C04-062C Sheet 1 of 2

8-Lead Plastic Dual Flat, No Lead Package (MF) - 3x3x0.9mm Body [DFN]

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	8		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Length	D	3.00 BSC		
Exposed Pad Width	E2	1.34	-	1.60
Overall Width	E	3.00 BSC		
Exposed Pad Length	D2	1.60	-	2.40
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.20	0.30	0.55
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

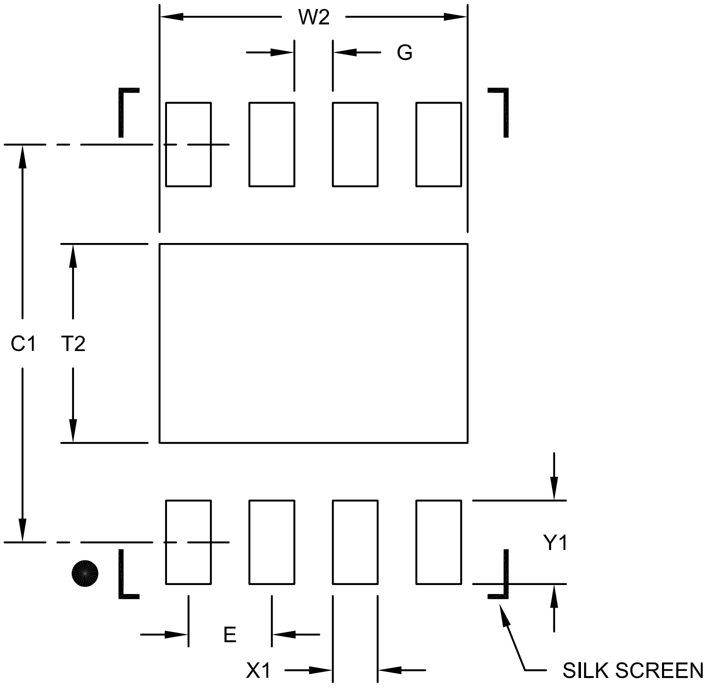
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated
4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-062C Sheet 2 of 2

8-Lead Plastic Dual Flat, No Lead Package (MF) - 3x3x0.9mm Body [DFN]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

		MILLIMETERS		
	Units			
	Dimension Limits	MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			2.40
Optional Center Pad Length	T2			1.55
Contact Pad Spacing	C1		3.10	
Contact Pad Width (X8)	X1			0.35
Contact Pad Length (X8)	Y1			0.65
Distance Between Pads	G	0.30		

Notes:

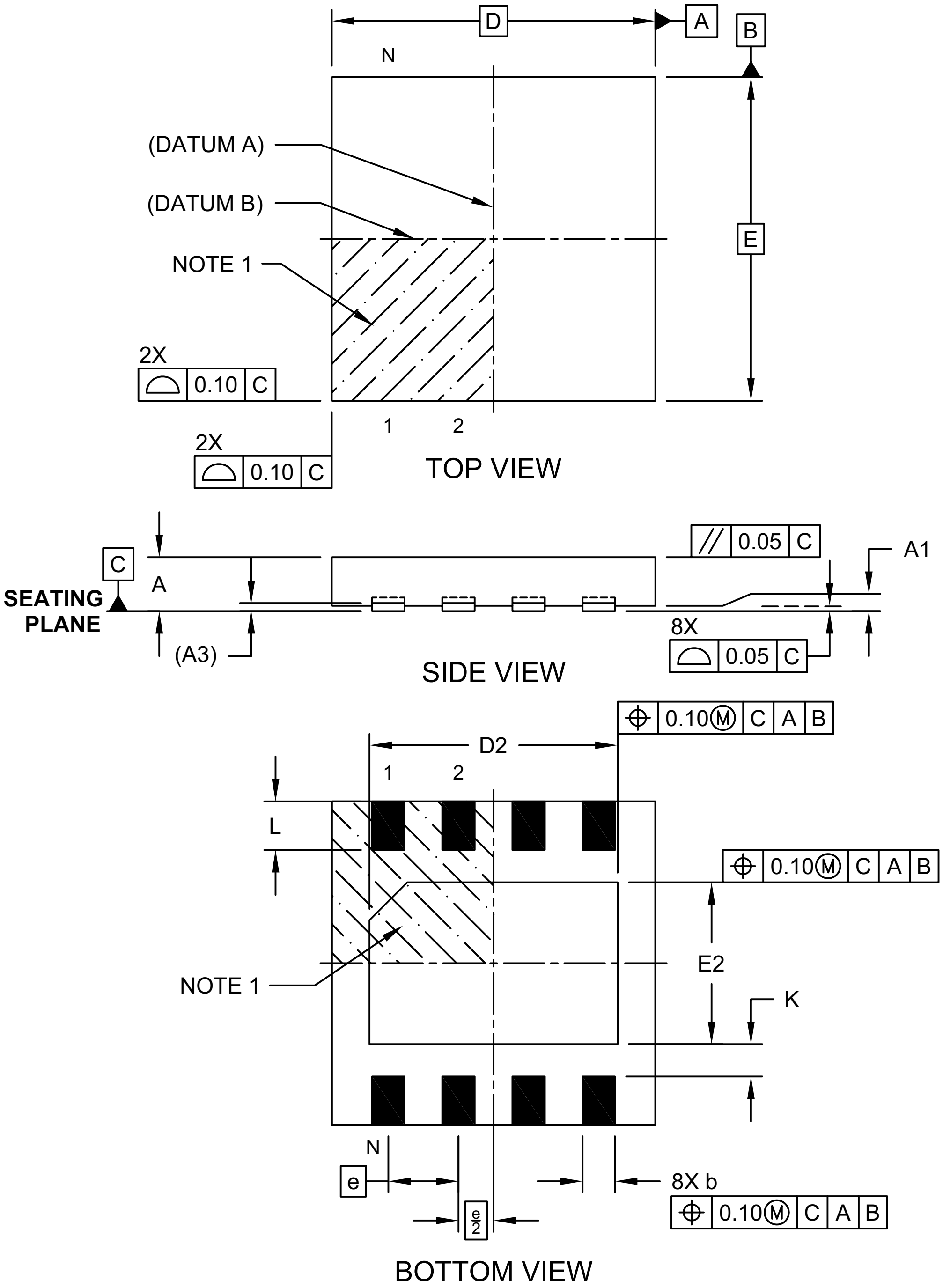
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2062B

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (RF) - 3x3x0.50 mm Body [UDFN]

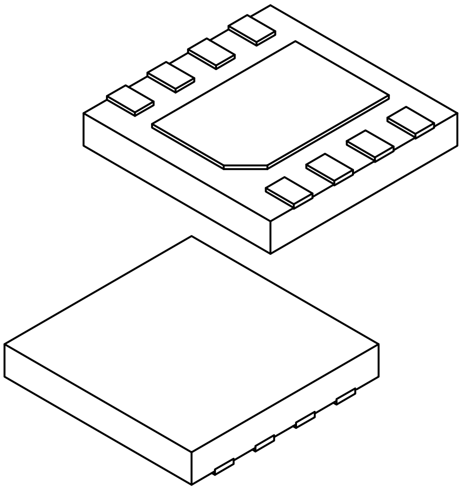
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-254A Sheet 1 of 2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (RF) - 3x3x0.50 mm Body [UDFN]

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	8		
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	0.065 REF		
Overall Width	E	3.00 BSC		
Exposed Pad Width	E2	1.40	1.50	1.60
Overall Length	D	3.00 BSC		
Exposed Pad Length	D2	2.20	2.30	2.40
Terminal Width	b	0.25	0.30	0.35
Terminal Length	L	0.35	0.45	0.55
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

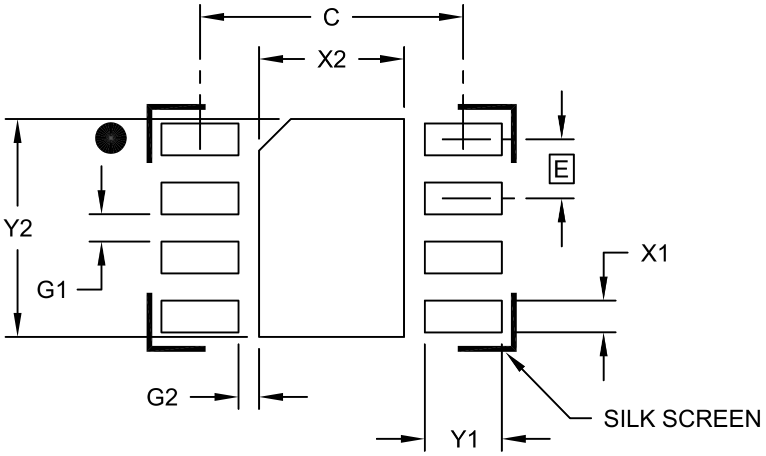
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-254A Sheet 2 of 2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (RF) - 3x3x0.50 mm Body [UDFN]

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			1.60
Optional Center Pad Length	Y2			2.40
Contact Pad Spacing	C		2.90	
Contact Pad Width (X8)	X1			0.35
Contact Pad Length (X8)	Y1			0.85
Contact Pad to Contact Pad (X6)	G1	0.20		
Contact Pad to Center Pad (X8)	G2	0.30		

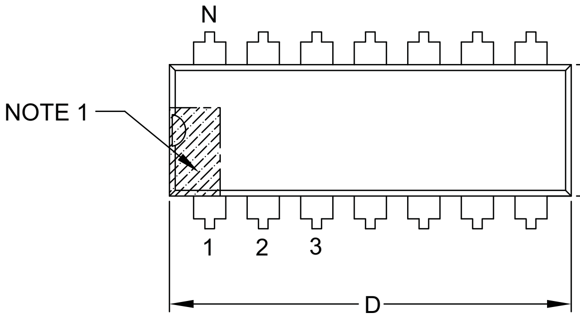
Notes:

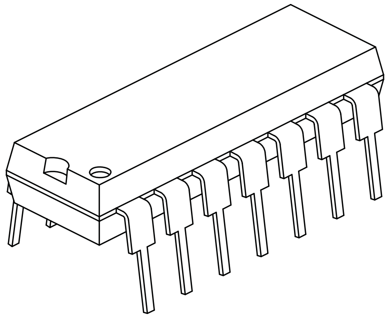
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

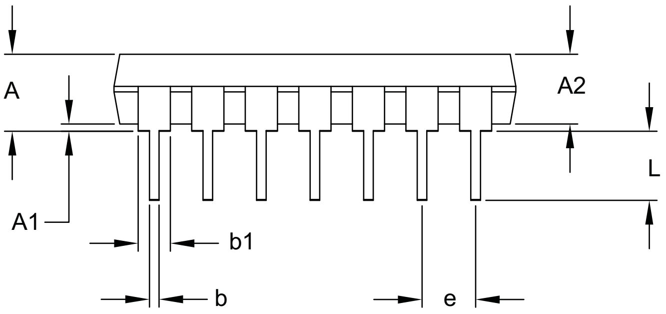
Microchip Technology Drawing C04-2254A

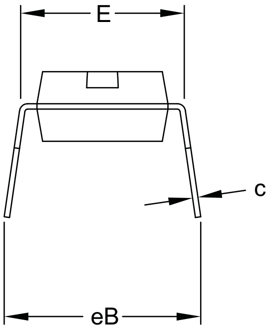
14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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:

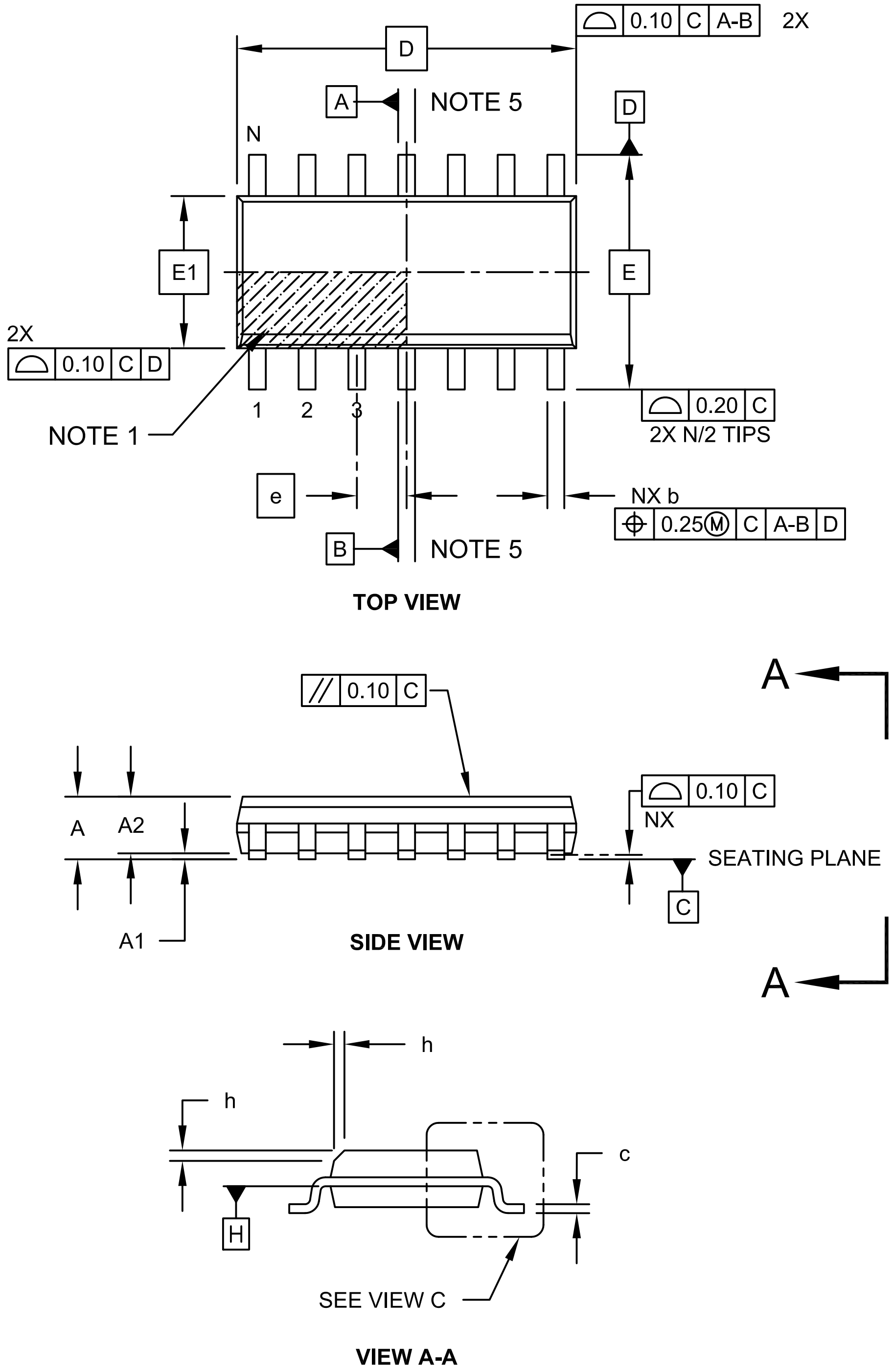
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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]

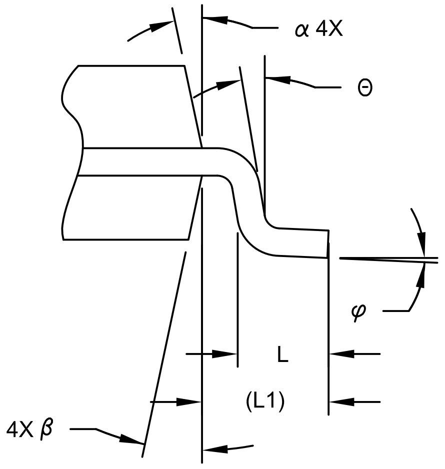
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



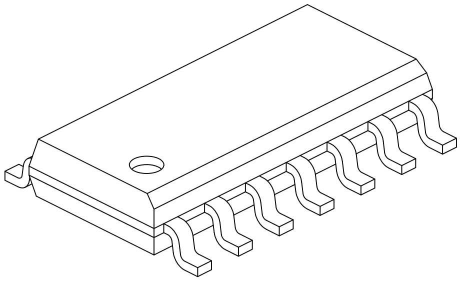
Microchip Technology Drawing No. C04-065C Sheet 1 of 2

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

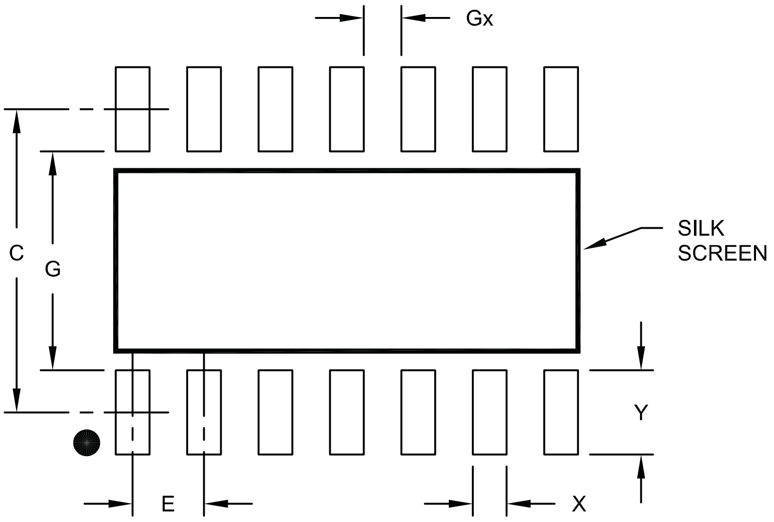
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   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:

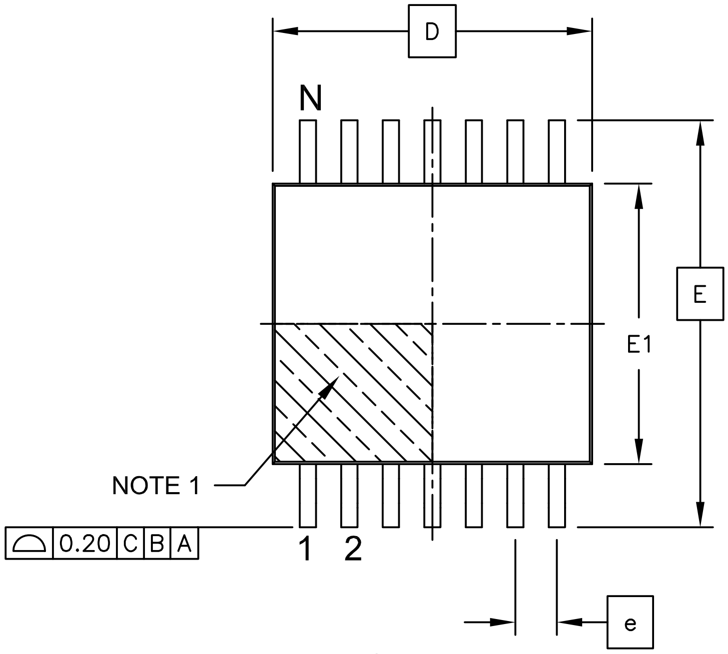
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

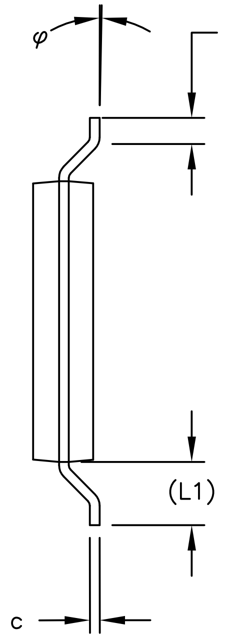
Microchip Technology Drawing No. C04-2065A

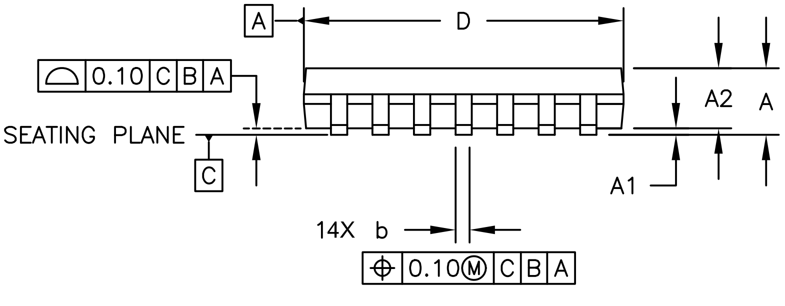
14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



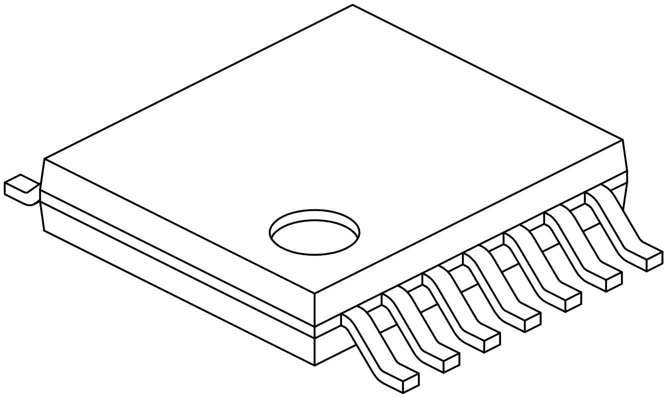


SIDE VIEW

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:

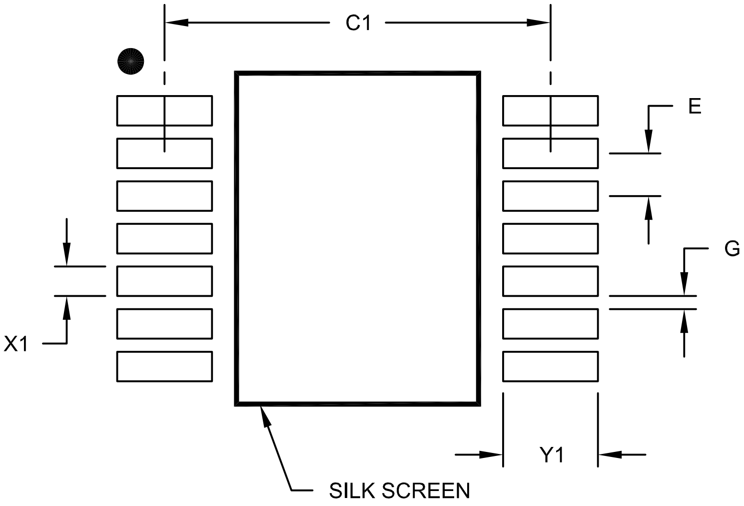
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

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:

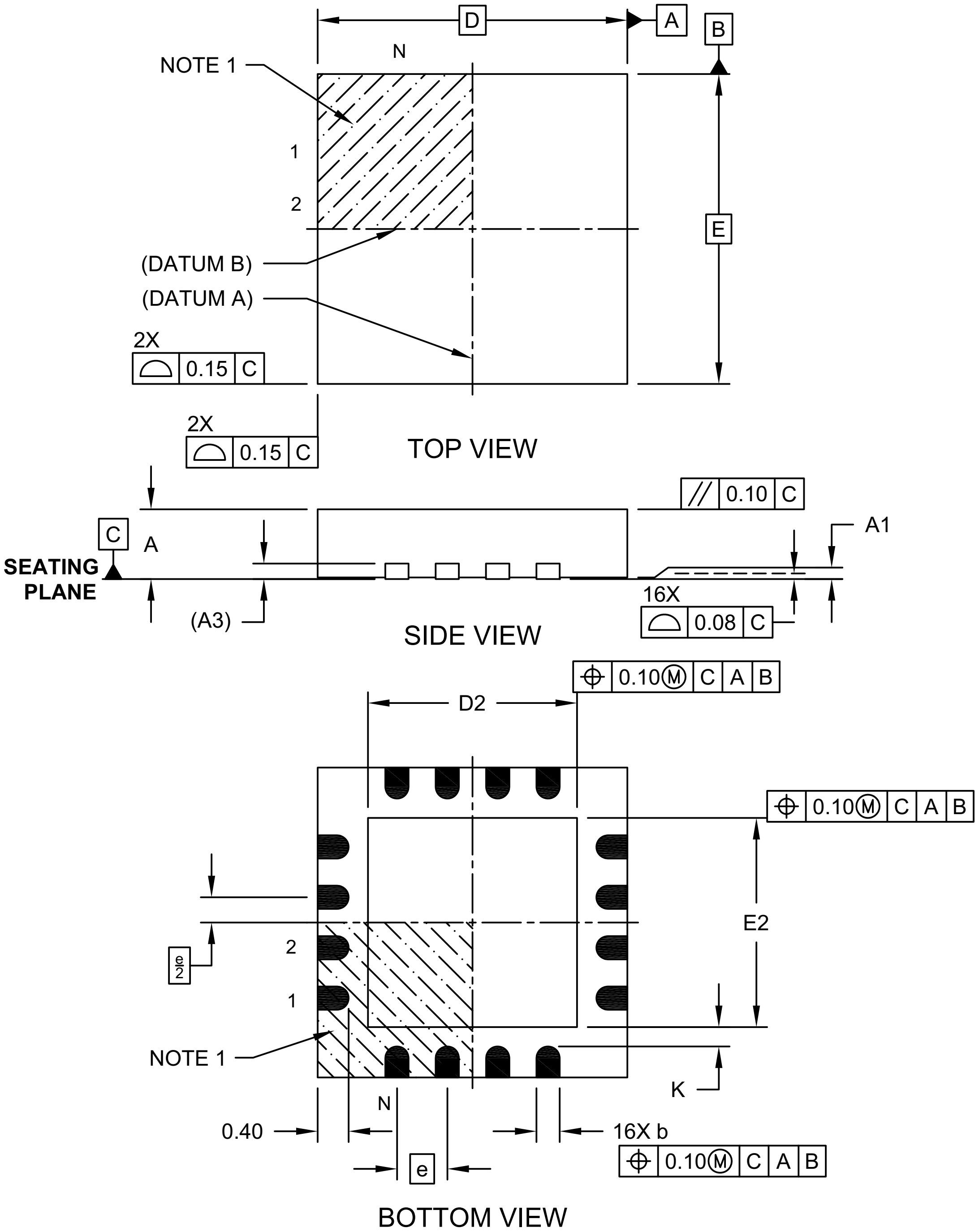
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

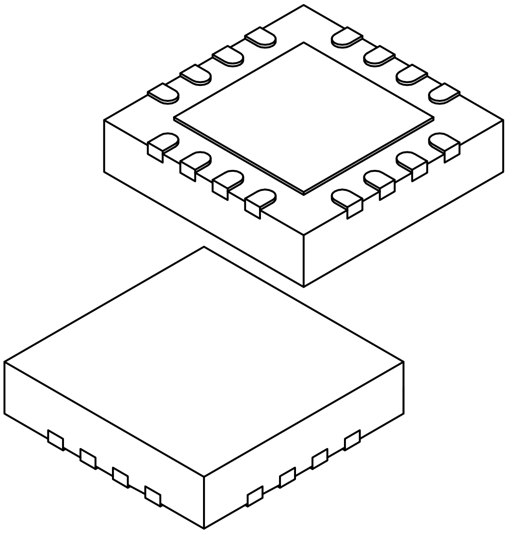
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-127D Sheet 1 of 2

16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N	16		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.50	2.65	2.80
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.50	2.65	2.80
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	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

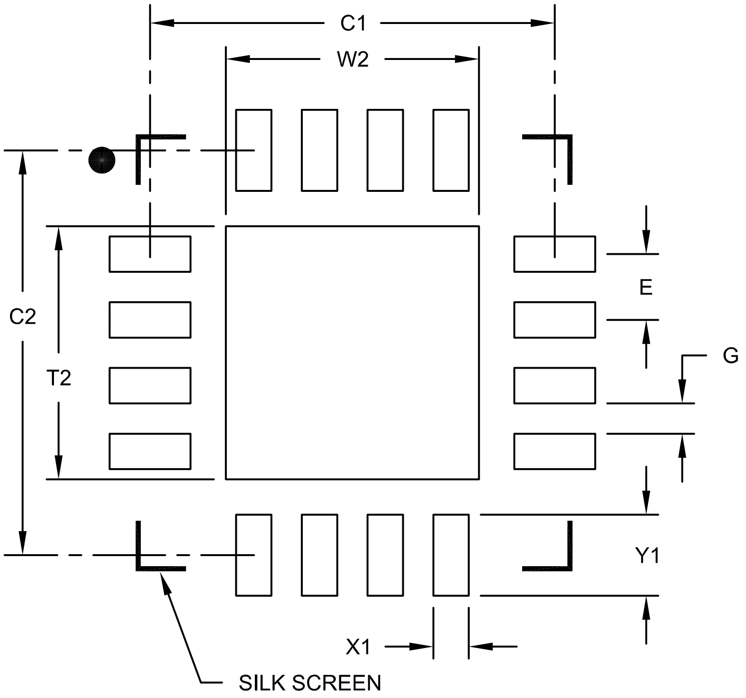
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127D Sheet 2 of 2

16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X28)	X1			0.35
Contact Pad Length (X28)	Y1			0.80
Distance Between Pads	G	0.30		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2127A

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (01/2014)

Original release.

Revision B (05/2016)

Added Section 1.1 Register and Bit Naming Conventions.

Added Register 12-14 WPUC register. Updated SMT Chapter.

Minor typos corrected.

Added High endurance column to Table 1: PIC12/16(L)F161x Family Types. Added Sections 22.1.1 and 22.1.2. Added Tables 22-1 and 22-3.

Updated the High-Endurance Flash data memory information on the cover page. Updated Figures 18-2, 21-1, 22-8, 23-2, and 23-3; Registers 19-1, 21-1, 22-3, 22-4, and 25-6; Sections 18,6, 18.7, 22.0, 22.1, 22.4, 22.5, 22.5.1, 22.5.2, 22.5.4, 22.5.5, 22.5.8, 23.1.7, 23.2.6, and 25.0; Tables 5-1, 7-1, 8-1, 22-1 and 25-3.

Updated Package Drawings C04-018, C04-127.

Deleted Section 24.1.1 and Registers 22-5 and 22-6.

Revision C (01/2017)

Updated Figure 16-1; Updated Registers 18-1, 19-1, and 23-1; Sections 11.3, 11.5, and 23.1; Tables 18-3, 22-4, 23-3, and 24-2. Added Register 24-8.

THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://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:	PIC12LF1612, PIC12F1612, PIC16LF1613, PIC16F1613
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]	MF = DFN (8-Lead) ML = QFN (16-Lead) P = Plastic DIP RF = Micro Lead Frame (UDFN) 3x3x0.5mm SL = SOIC (14-Lead SN = SOIC (8-Lead) ST = TSSOP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC12LF1612T - I/SN
Tape and Reel,
Industrial temperature,
SOIC package

b) PIC16F1613 - I/P
Industrial temperature
PDIP package

c) PIC16F1613 - E/ML 298
Extended temperature,
QFN package
QTP pattern #298

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.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, 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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-1259-5

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

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 - Guangzhou
Tel: 86-20-8755-8029

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: 86-21-3326-8000
Fax: 86-21-3326-8021

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: 886-7-213-7830

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

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700
Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

11/07/16

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

PIC16LF1613-E/ML PIC16LF1613-I/ML PIC16LF1613T-I/ML PIC16LF1613-E/SL PIC16LF1613-E/ST PIC16LF1613-I/P PIC16LF1613-I/SL PIC16LF1613-I/ST PIC16LF1613T-I/SL PIC16LF1613T-I/ST PIC12F1612-E/MF PIC12F1612-I/MF PIC12F1612T-I/MF PIC12LF1612-E/MF PIC12LF1612-E/P PIC12LF1612-E/SN PIC12LF1612-I/MF PIC12LF1612-I/P PIC12LF1612-I/SN PIC12LF1612T-I/MF PIC12LF1612T-I/SN PIC16F1613-E/SL PIC16F1613-I/SL PIC16F1613T-I/SL PIC16LF1613-E/P PIC16F1613-I/ML PIC16F1613-E/P PIC16F1613-I/P PIC12F1612-E/SN PIC16F1613T-I/ST PIC12F1612-I/P PIC12F1612-I/SN PIC16F1613-E/ML PIC16F1613-E/ST PIC16F1613T-I/ML PIC12F1612-E/P PIC16F1613-I/ST PIC12F1612T-I/SN

OCEAN CHIPS
Океан Электроники
Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения **«JONHON»**, а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов **«FORSTAR»**.

AVIC JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).

FORSTAR®

Телефон: 8 (812) 309-75-97 (многоканальный)
Факс: 8 (812) 320-03-32
Электронная почта: ocean@oceanchips.ru
Web: http://oceanchips.ru/
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А