Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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.

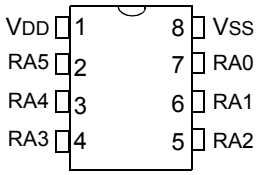
TABLE 2: PACKAGES

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

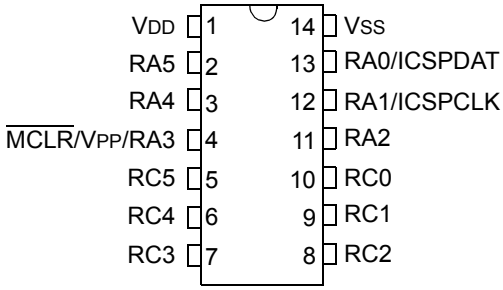
Note: Pin details are subject to change.

PIN DIAGRAMS

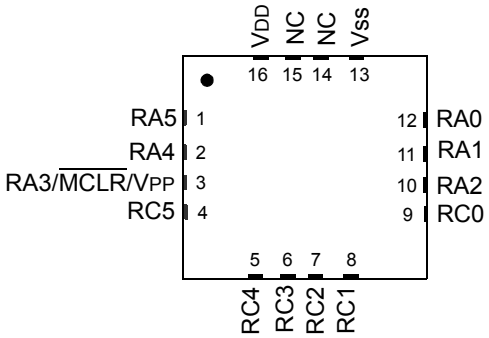
8-pin PDIP, SOIC, DFN, UDFN



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	$\overline{MCLR}$/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	$\overline{MCLR}$/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

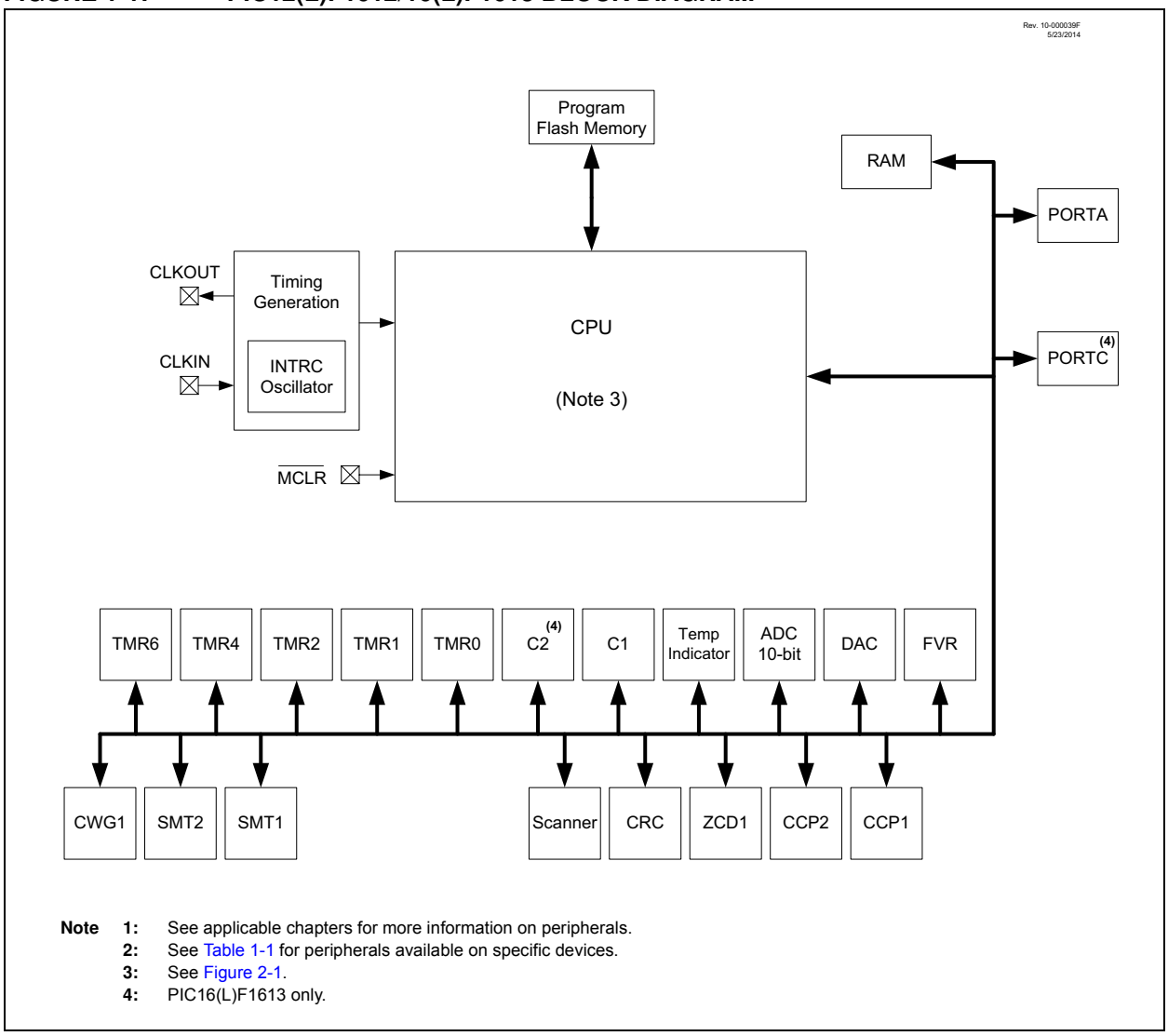
Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

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

Note 1: 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 CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C levels
HV = High Voltage XTAL = Crystal

Note 1: 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 CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C levels
HV = High Voltage XTAL = Crystal

Note 1: 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^2C = Schmitt Trigger input with I^2C levels
HV = High Voltage XTAL = Crystal

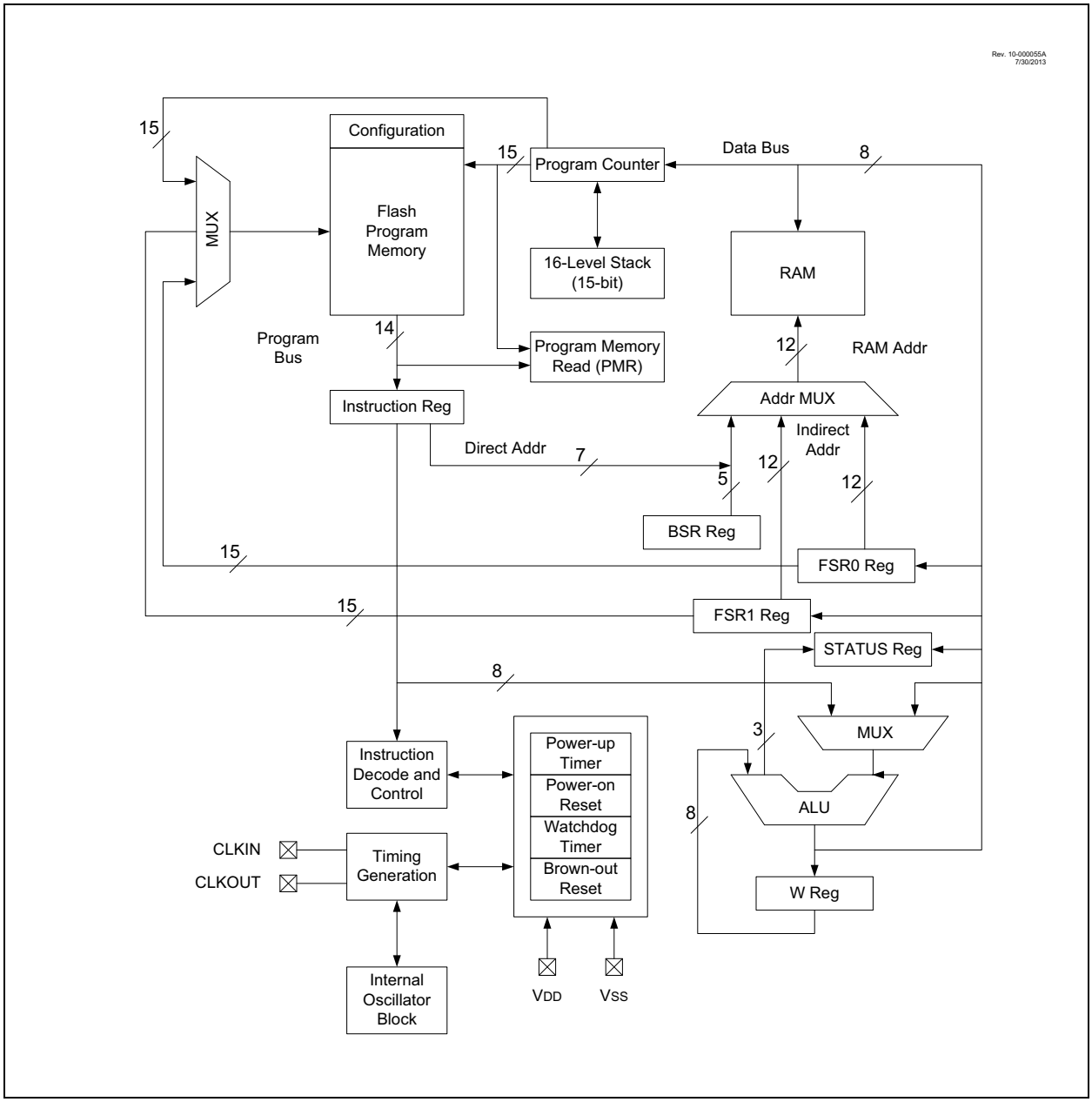
Note 1: Alternate pin function selected with the APFCON register (Register 12-1).

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).

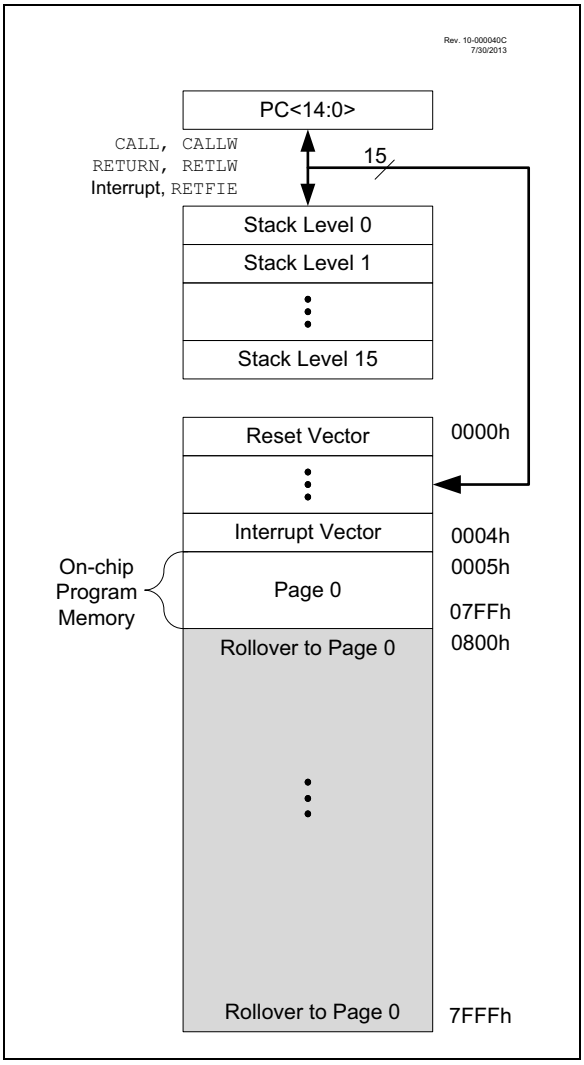
3.2 High-Endurance Flash

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

bit 3 **$\overline{PD}$:** 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/$\overline{Digit\ Borrow}$ 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/$\overline{Borrow}$ 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 $\overline{Borrow}$, 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

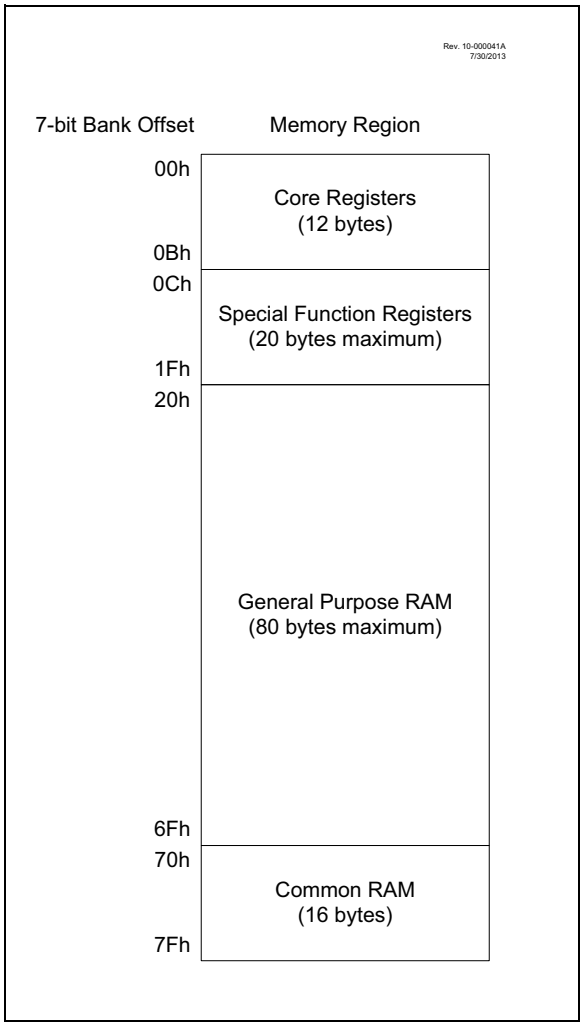


TABLE 3-2: PIC12(L)F1612 MEMORY MAP, BANK 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	Core Registers (Table 3-1)	080h	Core Registers (Table 3-1)	100h	Core Registers (Table 3-1)	180h	Core Registers (Table 3-1)	200h	Core Registers (Table 3-1)	280h	Core Registers (Table 3-1)	300h	Core Registers (Table 3-1)	380h	Core Registers (Table 3-1)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
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	General Purpose Register 80 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	Unimplemented Read as '0'	220h	Unimplemented Read as '0'	2A0h	Unimplemented Read as '0'	320h	Unimplemented Read as '0'	3A0h	Unimplemented Read as '0'
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h	Common RAM	0F0h	Common RAM (Accesses 70h – 7Fh)	170h	Common RAM (Accesses 70h – 7Fh)	1F0h	Common RAM (Accesses 70h – 7Fh)	270h	Common RAM (Accesses 70h – 7Fh)	2F0h	Common RAM (Accesses 70h – 7Fh)	370h	Common RAM (Accesses 70h – 7Fh)	3F0h	Common RAM (Accesses 70h – 7Fh)
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

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	Core Registers (Table 3-1)	080h	Core Registers (Table 3-1)	100h	Core Registers (Table 3-1)	180h	Core Registers (Table 3-1)	200h	Core Registers (Table 3-1)	280h	Core Registers (Table 3-1)	300h	Core Registers (Table 3-1)	380h	Core Registers (Table 3-1)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
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	General Purpose Register 80 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	Unimplemented Read as '0'	220h	Unimplemented Read as '0'	2A0h	Unimplemented Read as '0'	320h	Unimplemented Read as '0'	3A0h	Unimplemented Read as '0'
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h	Common RAM	0F0h	Common RAM (Accesses 70h – 7Fh)	170h	Common RAM (Accesses 70h – 7Fh)	1F0h	Common RAM (Accesses 70h – 7Fh)	270h	Common RAM (Accesses 70h – 7Fh)	2F0h	Common RAM (Accesses 70h – 7Fh)	370h	Common RAM (Accesses 70h – 7Fh)	3F0h	Common RAM (Accesses 70h – 7Fh)
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

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	Core Registers (Table 3-1)	480h	Core Registers (Table 3-1)	500h	Core Registers (Table 3-1)	580h	Core Registers (Table 3-1)	600h	Core Registers (Table 3-1)	680h	Core Registers (Table 3-1)	700h	Core Registers (Table 3-1)	780h	Core Registers (Table 3-1)
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh		78Bh	
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	Unimplemented Read as '0'	4A0h	Unimplemented Read as '0'	520h	Unimplemented Read as '0'	5A0h	Unimplemented Read as '0'	620h	Unimplemented Read as '0'	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh		7EFh	
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh	

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

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