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

# PIC16(L)F707

## 40/44-Pin Flash Microcontrollers with XLP and mTouch® Technology

### Devices included in this data sheet

- PIC16F707
- PIC16LF707

### High-Performance RISC CPU

- Only 35 Single-Word Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 20 MHz clock input
  - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes
- Processor Read Access to Program Memory
- Pinout Compatible to other 40-Pin PIC16CXXX and PIC16FXXX Microcontrollers

### Memory

- 8K x 14 Words of Flash Program Memory
- 363 Bytes of Data Memory (SRAM)

### Special Microcontroller Features

- Precision Internal Oscillator:
  - 16 MHz or 500 kHz operation
  - Factory calibrated to ±1%, typical
  - Software selectable ÷1, ÷2, ÷4 or ÷8 divider
- 31 kHz Low-Power Internal Oscillator
- External Oscillator Block with:
  - Three crystal/resonator modes up to 20 MHz
  - Three external clock modes up to 20 MHz
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Oscillator Start-Up Timer (OST)
- Brown-out Reset (BOR):
  - Selectable between two trip points
  - Disabled in Sleep option
- Watchdog Timer (WDT)
- Programmable Code Protection
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Multiplexed Master Clear with Pull-up/Input Pin
- Industrial and Extended Temperature Range
- Operating Voltage Range:
  - 1.8V to 3.6V (PIC16LF707)
  - 1.8V to 5.5V (PIC16F707)

### Extreme Low-Power Management PIC16LF707 with XLP

- Sleep mode: 20 nA @ 1.8V, typical
- Watchdog Timer: 500 nA @ 1.8V, typical
- Timer1 Oscillator: 600 nA @ 1.8V, typical @ 32 kHz

### mTouch® Technology Features

- Up to 32 Channels
- Two Capacitive Sensing modules:
  - Acquire two samples simultaneously
- Multiple Power modes:
  - Operation during Sleep
  - Proximity sensing with ultra low µA current
- Adjustable Waveform Min. and Max. for Optimal Noise Performance
- 1.8V to 5.5V Operation (3.6V max. for PIC16LF707)

### Analog Features

- A/D Converter:
  - 8-bit resolution and up to 14 channels
  - Conversion available during Sleep
  - Selectable 1.024V/2.048V/4.096V voltage reference
- On-chip 3.2V Regulator (PIC16F707 device only)

### Peripheral Highlights

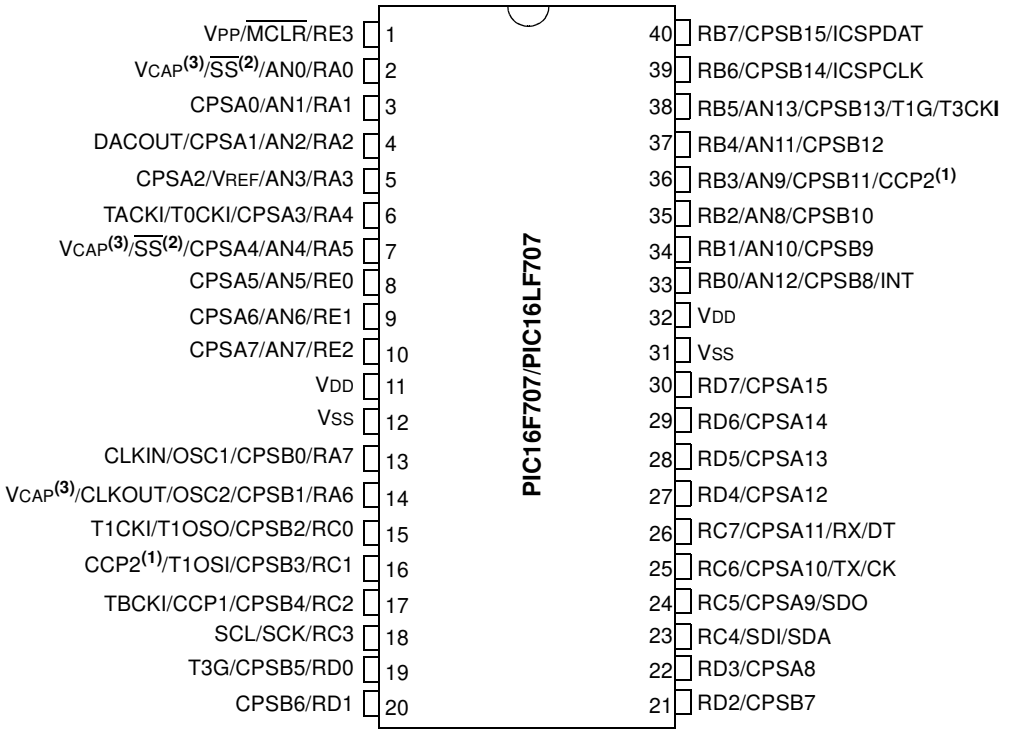
- Up to 35 I/O Pins and One Input-only Pin:
  - High current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
- Timer0/A/B: 8-Bit Timer/Counter with 8-Bit Prescaler
- Enhanced Timer1/3:
  - Dedicated low-power 32 kHz oscillator driver
  - 16-bit timer/counter with prescaler
  - External Gate Input mode with toggle and single shot modes
  - Interrupt-on-gate completion
- Timer2: 8-Bit Timer/Counter with 8-Bit Period Register, Prescaler and Postscaler

- Two Capture, Compare, PWM modules (CCP):
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART)
- Synchronous Serial Port (SSP):
  - SPI (Master/Slave)
  - $I^2C$ (Slave) with Address Mask
- Voltage Reference module:
  - Fixed Voltage Reference (FVR) with 1.024V, 2.048V and 4.096V output levels
  - 5-bit rail-to-rail resistive DAC with positive reference selection

| Device | Program Memory Flash (words) | SRAM (bytes) | High Endurance Flash (bytes) | I/Os | Capacitive Touch Channels | 8-bit A/D (ch) | AUSART | CCP | Timers 8/16-bit |
|---|---|---|---|---|---|---|---|---|---|
| PIC16(L)F707 | 8192 | 363 | 128 | 36 | 32 | 14 | Yes | 2 | 4/2 |

## PIN DIAGRAMS

**FIGURE 1: 40-PIN PDIP**

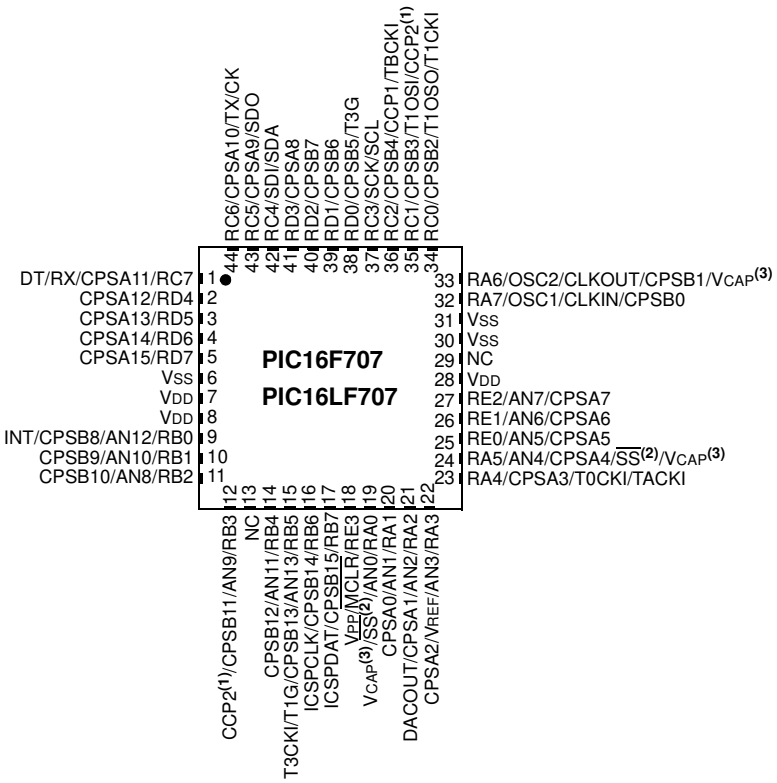



**Note 1:** CCP2 pin location may be selected as RB3 or RC1.

**2:** $\overline{SS}$ pin location may be selected as RA5 or RA0.

**3:** PIC16F707 only.

**FIGURE 2: 44-PIN QFN (8X8X0.9)**

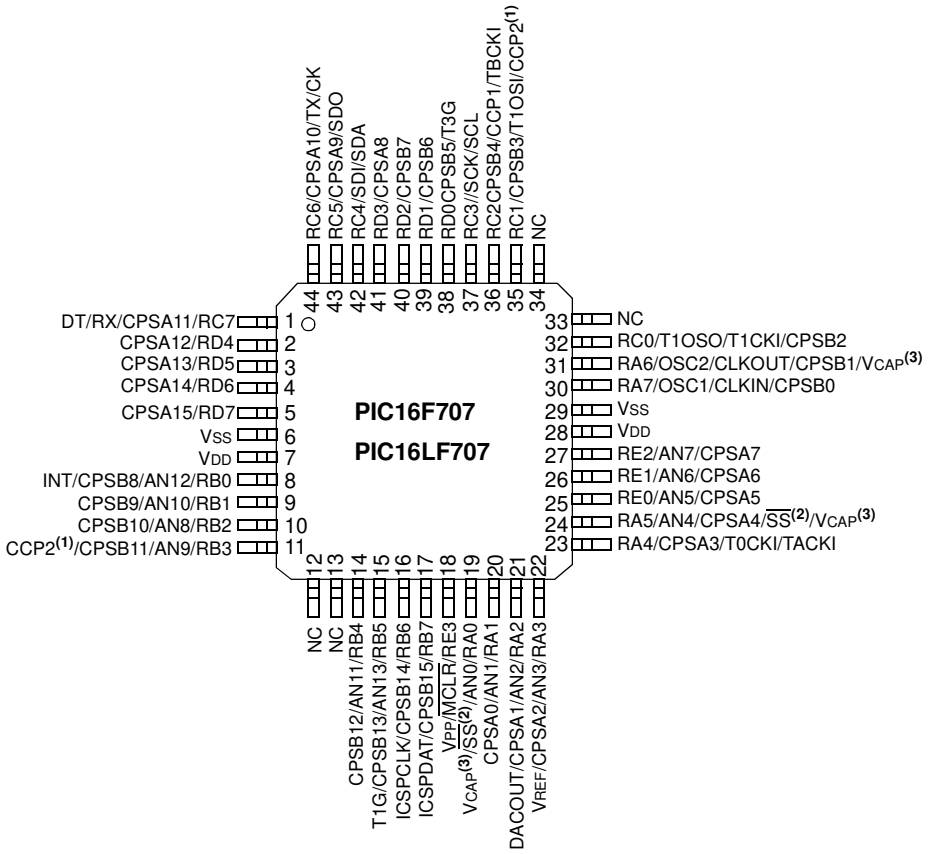



**Note 1:** CCP2 pin location may be selected as RB3 or RC1.

**2:** $\overline{SS}$ pin location may be selected as RA5 or RA0.

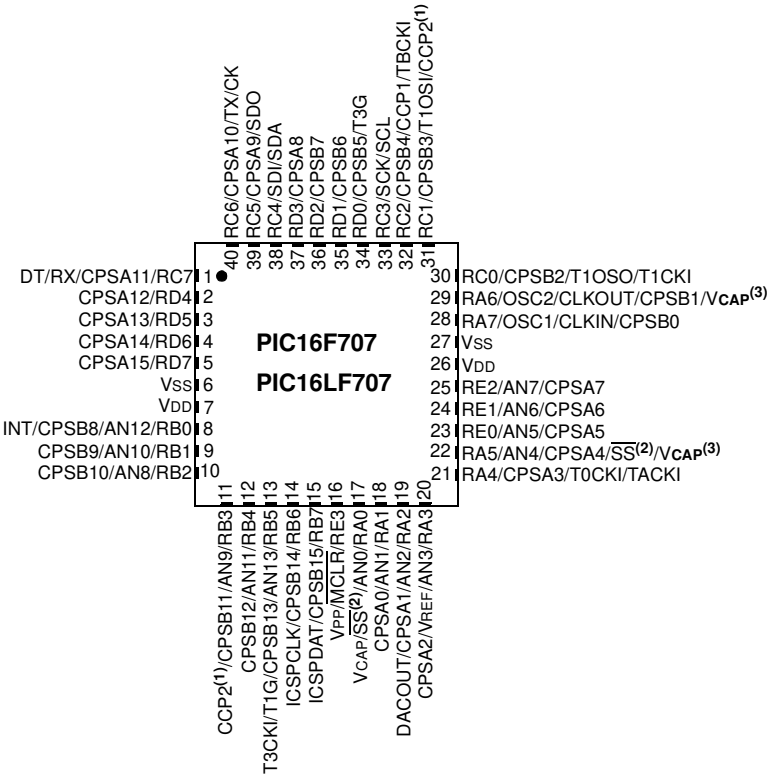
**3:** PIC16F707 only.

**FIGURE 3: 44-PIN TQFP**




**Note 1:** CCP2 pin location may be selected as RB3 or RC1.
**2:** $\overline{SS}$ pin location may be selected as RA5 or RA0.
**3:** PIC16F707 only.

**FIGURE 4: 40-PIN UQFN (5X5X0.5)**




**Note 1:** CCP2 pin location may be selected as RB3 or RC1.
**2:** SS pin location may be selected as RA5 or RA0.
**3:** PIC16F707 only.

## PIN ALLOCATION TABLE

### TABLE 1: 40/44-PIN ALLOCATION TABLE FOR PIC16F707/PIC16LF707

| I/O | 40-Pin PDIP | 40-Pin UQFN | 44-Pin TQFP | 44-Pin QFN | ANSEL | A/D | DAC | Cap Sensor | Timers | CCP | AUSART | SSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 2 | 17 | 19 | 19 | Y | AN0 | — | | — | — | — | $\overline{SS}$(3) | — | — | VCAP(4) |
| RA1 | 3 | 18 | 20 | 20 | Y | AN1 | — | CPSA0 | — | — | — | — | — | — | — |
| RA2 | 4 | 19 | 21 | 21 | Y | AN2 | DACOUT | CPSA1 | — | — | — | — | — | — | — |
| RA3 | 5 | 20 | 22 | 22 | Y | AN3/ VREF | VREF | CPSA2 | — | — | — | — | — | — | — |
| RA4 | 6 | 21 | 23 | 23 | Y | — | — | CPSA3 | T0CKI/ TACKI | — | — | — | — | — | — |
| RA5 | 7 | 22 | 24 | 24 | Y | AN4 | — | CPSA4 | — | — | — | $\overline{SS}$(3) | — | — | VCAP(4) |
| RA6 | 14 | 29 | 31 | 33 | Y | — | — | CPSB1 | — | — | — | — | — | — | OSC2/ CLKOUT/ VCAP(4) |
| RA7 | 13 | 28 | 30 | 32 | Y | — | — | CPSB0 | — | — | — | — | — | — | OSC1/ CLKIN |
| RB0 | 33 | 8 | 8 | 9 | Y | AN12 | — | CPSB8 | — | — | — | — | IOC/INT | Y | — |
| RB1 | 34 | 9 | 9 | 10 | Y | AN10 | — | CPSB9 | — | — | — | — | IOC | Y | — |
| RB2 | 35 | 10 | 10 | 11 | Y | AN8 | — | CPSB10 | — | — | — | — | IOC | Y | — |
| RB3 | 36 | 11 | 11 | 12 | Y | AN9 | — | CPSB11 | — | CCP2(2) | — | — | IOC | Y | — |
| RB4 | 37 | 12 | 14 | 14 | Y | AN11 | — | CPSB12 | — | — | — | — | IOC | Y | — |
| RB5 | 38 | 13 | 15 | 15 | Y | AN13 | — | CPSB13 | T1G/ T3CKI | — | — | — | IOC | Y | — |
| RB6 | 39 | 14 | 16 | 16 | Y | — | — | CPSB14 | — | — | — | — | IOC | Y | ICSPCLK/ ICDCLK |
| RB7 | 40 | 15 | 17 | 17 | Y | — | — | CPSB15 | — | — | — | — | IOC | Y | ICSPDAT/ ICDDAT |
| RC0 | 15 | 30 | 32 | 34 | Y | — | — | CPSB2 | T1OSO/ T1CKI | — | — | — | — | — | — |
| RC1 | 16 | 31 | 35 | 35 | Y | — | — | CPSB3 | T1OSI | CCP2(2) | — | — | — | — | — |
| RC2 | 17 | 32 | 36 | 36 | Y | — | — | CPSB4 | TBCKI | CCP1 | — | — | — | — | — |
| RC3 | 18 | 33 | 37 | 37 | — | — | — | — | — | — | — | SCK/ SCL | — | — | — |
| RC4 | 23 | 38 | 42 | 42 | — | — | — | — | — | — | — | SDI/ SDA | — | — | — |
| RC5 | 24 | 39 | 43 | 43 | Y | — | — | CPSA9 | — | — | — | SDO | — | — | — |
| RC6 | 25 | 40 | 44 | 44 | Y | — | — | CPSA10 | — | — | TX/CK | — | — | — | — |
| RC7 | 26 | 1 | 1 | 1 | Y | — | — | CPSA11 | — | — | RX/DT | — | — | — | — |
| RD0 | 19 | 34 | 38 | 38 | Y | — | — | CPSB5 | T3G | — | — | — | — | — | — |
| RD1 | 20 | 35 | 39 | 39 | Y | — | — | CPSB6 | — | — | — | — | — | — | — |
| RD2 | 21 | 36 | 40 | 40 | Y | — | — | CPSB7 | — | — | — | — | — | — | — |
| RD3 | 22 | 37 | 41 | 41 | Y | — | — | CPSA8 | — | — | — | — | — | — | — |
| RD4 | 27 | 2 | 2 | 2 | Y | — | — | CPSA12 | — | — | — | — | — | — | — |
| RD5 | 28 | 3 | 3 | 3 | Y | — | — | CPSA13 | — | — | — | — | — | — | — |
| RD6 | 29 | 4 | 4 | 4 | Y | — | — | CPSA14 | — | — | — | — | — | — | — |
| RD7 | 30 | 5 | 5 | 5 | Y | — | — | CPSA15 | — | — | — | — | — | — | — |
| RE0 | 8 | 23 | 25 | 25 | Y | AN5 | — | CPSA5 | — | — | — | — | — | — | — |
| RE1 | 9 | 24 | 26 | 26 | Y | AN6 | — | CPSA6 | — | — | — | — | — | — | — |
| RE2 | 10 | 25 | 27 | 27 | Y | AN7 | — | CPSA7 | — | — | — | — | — | — | — |
| RE3 | 1 | 16 | 18 | 18 | — | — | — | — | — | — | — | — | — | Y(1) | $\overline{MCLR}$/ VPP |

**Note 1:** Pull-up activated only with external $\overline{MCLR}$ configuration.

**2:** RC1 is the default pin location for CCP2. RB3 may be selected by changing the CCP2SEL bit in the APFCON register.

**3:** RA5 is the default pin location for $\overline{SS}$. RA0 may be selected by changing the SSSEL bit in the APFCON register.

**4:** PIC16F707 only. VCAP functionality is selectable by the VCAPEN bits in Configuration Word 2.

**TABLE 1: 40/44-PIN ALLOCATION TABLE FOR PIC16F707/PIC16LF707**

| I/O | 40-Pin PDIP | 40-Pin UQFN | 44-Pin TQFP | 44-Pin QFN | ANSEL | A/D | DAC | Cap Sensor | Timers | CCP | AUSART | SSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VDD | 11, 32 | 7, 26 | 7, 28 | 7, 8, 28 | | — | — | — | — | — | — | — | — | — | VDD |
| Vss | 12, 31 | 6, 27 | 6, 29 | 6, 30, 31 | | — | — | — | — | — | — | — | — | — | Vss |

**Note 1:** Pull-up activated only with external $\overline{\text{MCLR}}$ configuration.

**2:** RC1 is the default pin location for CCP2. RB3 may be selected by changing the CCP2SEL bit in the APFCON register.

**3:** RA5 is the default pin location for $\overline{\text{SS}}$. RA0 may be selected by changing the SSSEL bit in the APFCON register.

**4:** PIC16F707 only. VCAP functionality is selectable by the VCAPEN bits in Configuration Word 2.

## 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., DS30000000A is version A of document DS30000000).

### Errata

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

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

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

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

### Customer Notification System

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

## 1.0 DEVICE OVERVIEW

The PIC16(L)F707 devices are covered by this data sheet. They are available in 40/44-pin packages. Figure 1-1 shows a block diagram of the PIC16(L)F707 devices. Table 1-1 shows the pinout descriptions.

**FIGURE 1-1: PIC16(L)F707 BLOCK DIAGRAM**

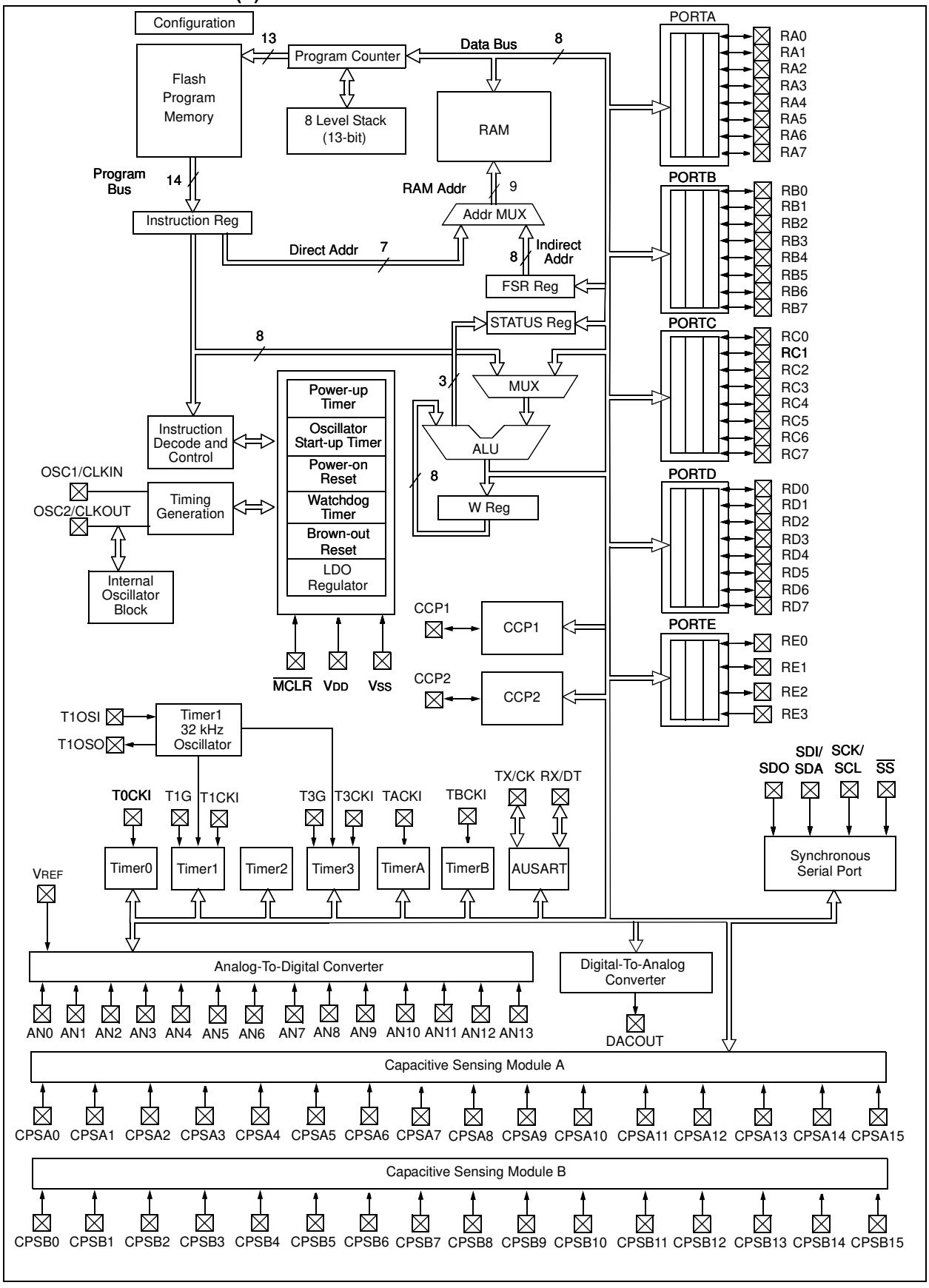


**TABLE 1-1: PIC16(L)F707 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/$\overline{SS}$/VCAP | RA0 | TTL | CMOS | General purpose I/O. |
| | AN0 | AN | — | A/D Channel 0 input. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| | VCAP | Power | Power | Filter capacitor for Voltage Regulator (PIC16F only). |
| RA1/AN1/CPSA0 | RA1 | TTL | CMOS | General purpose I/O. |
| | AN1 | AN | — | A/D Channel 1 input. |
| | CPSA0 | AN | — | Capacitive sensing A input 0. |
| RA2/AN2/CPSA1/DACOUT | RA2 | TTL | CMOS | General purpose I/O. |
| | AN2 | AN | — | A/D Channel 2 input. |
| | CPSA1 | AN | — | Capacitive sensing A input 1. |
| | DACOUT | — | AN | Voltage Reference Output. |
| RA3/AN3/VREF/CPSA2 | RA3 | TTL | CMOS | General purpose I/O. |
| | AN3 | AN | — | A/D Channel 3 input. |
| | VREF | AN | — | A/D Voltage Reference input. |
| | CPSA2 | AN | — | Capacitive sensing A input 2. |
| RA4/CPSA3/T0CKI/TACKI | RA4 | TTL | CMOS | General purpose I/O. |
| | CPSA3 | AN | — | Capacitive sensing A input 3. |
| | T0CKI | ST | — | Timer0 clock input. |
| | TACKI | ST | — | TimerA clock input. |
| RA5/AN4/CPSA4/$\overline{SS}$/VCAP | RA5 | TTL | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel 4 input. |
| | CPSA4 | AN | — | Capacitive sensing A input 4. |
| | $\overline{SS}$ | ST | — | Slave Select input. |
| | VCAP | Power | Power | Filter capacitor for Voltage Regulator (PIC16F only). |
| RA6/OSC2/CLKOUT/VCAP/ CPSB1 | RA6 | TTL | CMOS | General purpose I/O. |
| | OSC2 | — | XTAL | Crystal/Resonator (LP, XT, HS modes). |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| | VCAP | Power | Power | Filter capacitor for Voltage Regulator (PIC16F only). |
| | CPSB1 | AN | — | Capacitive sensing B input 1. |
| RA7/OSC1/CLKIN/CPSB0 | RA7 | TTL | CMOS | General purpose I/O. |
| | OSC1 | XTAL | — | Crystal/Resonator (LP, XT, HS modes). |
| | CLKIN | CMOS | — | External clock input (EC mode). |
| | CLKIN | ST | — | RC oscillator connection (RC mode). |
| | CPSB0 | AN | — | Capacitive sensing B input 0. |
| RB0/AN12/CPSB8/INT | RB0 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN12 | AN | — | A/D Channel 12 input. |
| | CPSB8 | AN | — | Capacitive sensing B input 8. |
| | INT | ST | — | External interrupt. |
| RB1/AN10/CPSB9 | RB1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10 input. |
| | CPSB9 | AN | — | Capacitive sensing B input 9. |

**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

**TABLE 1-1: PIC16(L)F707 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RB2/AN8/CPSB10 | RB2 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN8 | AN | — | A/D Channel 8 input. |
| | CPSB10 | AN | — | Capacitive sensing B input 10. |
| RB3/AN9/CPSB11/CCP2 | RB3 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN9 | AN | — | A/D Channel 9 input. |
| | CPSB11 | AN | — | Capacitive sensing B input 11. |
| | CCP2 | ST | CMOS | Capture/Compare/PWM2. |
| RB4/AN11/CPSB12 | RB4 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN11 | AN | — | A/D Channel 11 input. |
| | CPSB12 | AN | — | Capacitive sensing B input 12. |
| RB5/AN13/CPSB13/T1G/T3CKI | RB5 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN13 | AN | — | A/D Channel 13 input. |
| | CPSB13 | AN | — | Capacitive sensing B input 13. |
| | T1G | ST | — | Timer1 gate input. |
| | T3CKI | ST | — | Timer3 clock input. |
| RB6/ICSPCLK/ICDCLK/CPSB14 | RB6 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| | ICDCLK | ST | — | In-Circuit Debug Clock. |
| | CPSB14 | AN | — | Capacitive sensing B input 14. |
| RB7/ICSPDAT/ICDDAT/CPSB15 | RB7 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | ICSPDAT | ST | CMOS | ICSP™ Data I/O. |
| | ICDDAT | ST | — | In-Circuit Data I/O. |
| | CPSB15 | AN | — | Capacitive sensing B input 15. |
| RC0/T1OSO/T1CKI/CPSB2 | RC0 | ST | CMOS | General purpose I/O. |
| | T1OSO | XTAL | XTAL | Timer1 oscillator connection. |
| | T1CKI | ST | — | Timer1 clock input. |
| | CPSB2 | AN | — | Capacitive sensing B input 2. |
| RC1/T1OSI/CCP2/CPSB3 | RC1 | ST | CMOS | General purpose I/O. |
| | T1OSI | XTAL | XTAL | Timer1 oscillator connection. |
| | CCP2 | ST | CMOS | Capture/Compare/PWM2. |
| | CPSB3 | AN | — | Capacitive sensing B input 3. |
| RC2/CCP1/CPSB4/TBCKI | RC2 | ST | CMOS | General purpose I/O. |
| | CCP1 | ST | CMOS | Capture/Compare/PWM1. |
| | CPSB4 | AN | — | Capacitive sensing B input 4. |
| | TBCKI | ST | — | TimerB clock input. |
| RC3/SCK/SCL | RC3 | ST | CMOS | General purpose I/O. |
| | SCK | ST | CMOS | SPI clock. |
| | SCL | $I^2C$ | OD | $I^2C$ clock. |

**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

**TABLE 1-1: PIC16(L)F707 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC4/SDI/SDA | RC4 | ST | CMOS | General purpose I/O. |
| | SDI | ST | — | SPI data input. |
| | SDA | $I^2C$ | OD | $I^2C$ data input/output. |
| RC5/SDO/CPSA9 | RC5 | ST | CMOS | General purpose I/O. |
| | SDO | — | CMOS | SPI data output. |
| | CPSA9 | AN | — | Capacitive sensing A input 9. |
| RC6/TX/CK/CPSA10 | RC6 | ST | CMOS | General purpose I/O. |
| | TX | — | CMOS | USART asynchronous transmit. |
| | CK | ST | CMOS | USART synchronous clock. |
| | CPSA10 | AN | — | Capacitive sensing A input 10. |
| RC7/RX/DT/CPSA11 | RC7 | ST | CMOS | General purpose I/O. |
| | RX | ST | — | USART asynchronous input. |
| | DT | ST | CMOS | USART synchronous data. |
| | CPSA11 | AN | — | Capacitive sensing A input 11. |
| RD0/CPSB5/T3G | RD0 | ST | CMOS | General purpose I/O. |
| | CPSB5 | AN | — | Capacitive sensing B input 5. |
| | T3G | ST | — | Timer3 Gate input. |
| RD1/CPSB6 | RD1 | ST | CMOS | General purpose I/O. |
| | CPSB6 | AN | — | Capacitive sensing B input 6. |
| RD2/CPSB7 | RD2 | ST | CMOS | General purpose I/O. |
| | CPSB7 | AN | — | Capacitive sensing B input 7. |
| RD3/CPSA8 | RD3 | ST | CMOS | General purpose I/O. |
| | CPSA8 | AN | — | Capacitive sensing A input 8. |
| RD4/CPSA12 | RD4 | ST | CMOS | General purpose I/O. |
| | CPSA12 | AN | — | Capacitive sensing A input 12. |
| RD5/CPSA13 | RD5 | ST | CMOS | General purpose I/O. |
| | CPSA13 | AN | — | Capacitive sensing A input 13. |
| RD6/CPSA14 | RD6 | ST | CMOS | General purpose I/O. |
| | CPSA14 | AN | — | Capacitive sensing A input 14. |
| RD7/CPSA15 | RD7 | ST | CMOS | General purpose I/O. |
| | CPSA15 | AN | — | Capacitive sensing A input 15. |
| RE0/AN5/CPSA5 | RE0 | ST | CMOS | General purpose I/O. |
| | AN5 | AN | — | A/D Channel 5 input. |
| | CPSA5 | AN | — | Capacitive sensing A input 5. |
| RE1/AN6/CPSA6 | RE1 | ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | A/D Channel 6 input. |
| | CPSA6 | AN | — | Capacitive sensing A input 6. |
| RE2/AN7/CPSA7 | RE2 | ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | A/D Channel 7 input. |
| | CPSA7 | AN | — | Capacitive sensing A input 7. |
| RE3/$\overline{MCLR}$/$V_{PP}$ | RE3 | TTL | — | General purpose input. |
| | $\overline{MCLR}$ | ST | — | Master Clear with internal pull-up. |
| | $V_{PP}$ | HV | — | Programming voltage. |
| $V_{DD}$ | $V_{DD}$ | Power | — | Positive supply. |

**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

**TABLE 1-1: PIC16(L)F707 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| 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

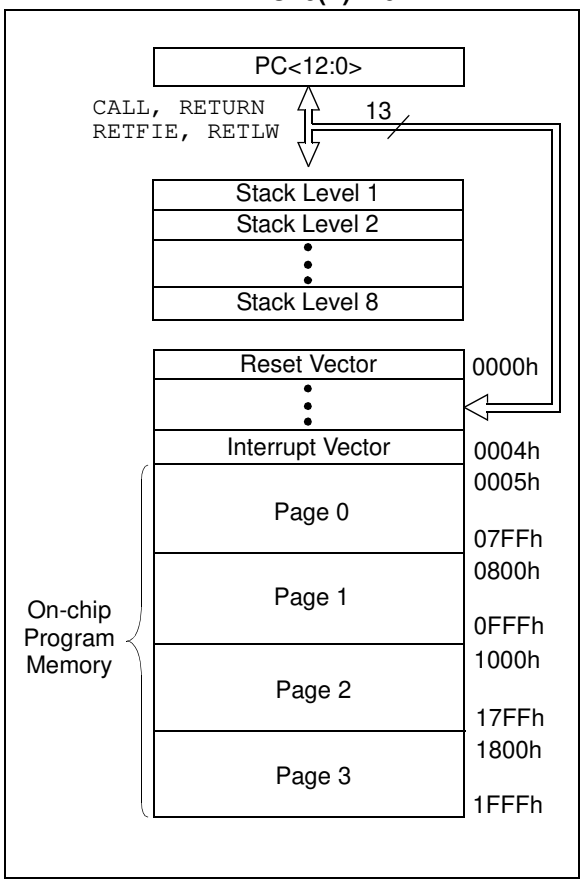
**Note:** The PIC16F707 devices have an internal low dropout voltage regulator. An external capacitor must be connected to one of the available VCAP pins to stabilize the regulator. For more information, see **Section 5.0 "Low Dropout (LDO) Voltage Regulator"**. The PIC16LF707 devices do not have the voltage regulator and therefore no external capacitor is required.

# 2.0 MEMORY ORGANIZATION

## 2.1 Program Memory Organization

The PIC16(L)F707 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h.

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F707**




## 2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPRs) and the Special Function Registers (SFRs). Bits RP0 and RP1 are bank select bits.

| RP1 | RP0 | | |
|---|---|---|---|
| 0 | 0 | → | Bank 0 is selected |
| 0 | 1 | → | Bank 1 is selected |
| 1 | 0 | → | Bank 2 is selected |
| 1 | 1 | → | Bank 3 is selected |

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank are mirrored in another bank for code reduction and quicker access.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 363 x 8 bits. Each register is accessed either directly or indirectly through the File Select Register (FSR), (Refer to **Section 2.5 "Indirect Addressing, INDF and FSR Registers"**).

### 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (refer to Table 2-2). These registers are static RAM.

The Special Function Registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the "core" are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

**TABLE 2-1: DATA MEMORY MAP FOR PIC16(L)F707**

| | BANK 0 | BANK 1 | BANK 2 | BANK 3 |
|---|---|---|---|---|
| File Address | | | | |
| | Indirect addr.(*) | Indirect addr.(*) | Indirect addr.(*) | Indirect addr.(*) |
| 00h / 80h / 100h / 180h | | | | |
| 01h / 81h / 101h / 181h | TMR0 | OPTION | TMR0 | OPTION |
| 02h / 82h / 102h / 182h | PCL | PCL | PCL | PCL |
| 03h / 83h / 103h / 183h | STATUS | STATUS | STATUS | STATUS |
| 04h / 84h / 104h / 184h | FSR | FSR | FSR | FSR |
| 05h / 85h / 105h / 185h | PORTA | TRISA | TACON | ANSELA |
| 06h / 86h / 106h / 186h | PORTB | TRISB | CPSBCON0 | ANSELB |
| 07h / 87h / 107h / 187h | PORTC | TRISC | CPSBCON1 | ANSELC |
| 08h / 88h / 108h / 188h | PORTD | TRISD | CPSACON0 | ANSELD |
| 09h / 89h / 109h / 189h | PORTE | TRISE | CPSACON1 | ANSELE |
| 0Ah / 8Ah / 10Ah / 18Ah | PCLATH | PCLATH | PCLATH | PCLATH |
| 0Bh / 8Bh / 10Bh / 18Bh | INTCON | INTCON | INTCON | INTCON |
| 0Ch / 8Ch / 10Ch / 18Ch | PIR1 | PIE1 | PMDATL | PMCON1 |
| 0Dh / 8Dh / 10Dh / 18Dh | PIR2 | PIE2 | PMADRL | Reserved |
| 0Eh / 8Eh / 10Eh / 18Eh | TMR1L | PCON | PMDATH | Reserved |
| 0Fh / 8Fh / 10Fh / 18Fh | TMR1H | T1GCON | PMADRH | Reserved |
| 10h / 90h / 110h / 190h | T1CON | OSCCON | TMRA | General Purpose Register 16 Bytes |
| 11h / 91h / 111h / 191h | TMR2 | OSCTUNE | TBCON | |
| 12h / 92h / 112h / 192h | T2CON | PR2 | TMRB | |
| 13h / 93h / 113h / 193h | SSPBUF | SSPADD/SSPMSK | DACCON0 | |
| 14h / 94h / 114h / 194h | SSPCON | SSPSTAT | DACCON1 | |
| 15h / 95h / 115h / 195h | CCPR1L | WPUB | General Purpose Register 11 Bytes | |
| 16h / 96h / 116h / 196h | CCPR1H | IOCB | | |
| 17h / 97h / 117h / 197h | CCP1CON | T3CON | | |
| 18h / 98h / 118h / 198h | RCSTA | TXSTA | | |
| 19h / 99h / 119h / 199h | TXREG | SPBRG | | |
| 1Ah / 9Ah / 11Ah / 19Ah | RCREG | TMR3L | | |
| 1Bh / 9Bh / 11Bh / 19Bh | CCPR2L | TMR3H | | |
| 1Ch / 9Ch / 11Ch / 19Ch | CCPR2H | APFCON | | |
| 1Dh / 9Dh / 11Dh / 19Dh | CCP2CON | FVRCON | | |
| 1Eh / 9Eh / 11Eh / 19Eh | ADRES | T3GCON | | |
| 1Fh / 9Fh / 11Fh / 19Fh | ADCON0 | ADCON1 | | |
| 20h / A0h / 120h / 1A0h | General Purpose Register 96 Bytes | General Purpose Register 80 Bytes | General Purpose Register 80 Bytes | General Purpose Register 80 Bytes |
| EFh / 16Fh / 1EFh | | | | |
| F0h / 170h / 1F0h | Accesses 70h – 7Fh | Accesses 70h – 7Fh | Accesses 70h – 7Fh | Accesses 70h – 7Fh |
| FFh / 17Fh / 1FFh | | | | |
| 7Fh | | | | |

**Legend:** = Unimplemented data memory locations, read as '0',
* = Not a physical register

**TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h(2) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 01h | TMR0 | Timer0 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 02h(2) | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h(2) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h(2) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | PORTA Data Latch when written: PORTA pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 06h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 07h | PORTC | PORTC Data Latch when written: PORTC pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 08h | PORTD | PORTD Data Latch when written: PORTD pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h | PORTE | — | — | — | — | RE3 | RE2 | RE1 | RE0 | ---- xxxx | ---- uuuu |
| 0Ah(1),(2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh(2) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | TMR3GIF | TMR3IF | TMRBIF | TMRAIF | — | — | — | CCP2IF | 0000 ---0 | 0000 ---0 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | TMR1CS1 | TMR1CS0 | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | — | TMR1ON | 0000 00-0 | uuuu uu-u |
| 11h | TMR2 | Timer2 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 15h | CCPR1L | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | USART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | USART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | CCPR2L | Capture/Compare/PWM Register 2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch | CCPR2H | Capture/Compare/PWM Register 2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh | CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |
| 1Eh | ADRES | A/D Result Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | — | — | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | ADON | --00 0000 | --00 0000 |

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

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** These registers can be addressed from any bank.

**3:** Accessible only when SSPM<3:0> = 1001.

**TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h(2) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 81h | OPTION_REG | $\overline{RBPU}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h(2) | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h(2) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h(2) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | PORTA Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 88h | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 89h | TRISE | — | — | — | — | '1' | TRISE2 | TRISE1 | TRISE0 | ---- 1111 | ---- 1111 |
| 8Ah(1),(2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 8Bh(2) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 8Ch | PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh | PIE2 | TMR3GIE | TMR3IE | TMRBIE | TMRAIE | — | — | — | CCP2IE | 0000 ---0 | 0000 ---0 |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ | ---- --qq | ---- --uu |
| 8Fh | T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{DONE}$ | T1GVAL | T1GSS1 | T1GSS0 | 0000 0x00 | uuuu uxuu |
| 90h | OSCCON | — | — | IRCF1 | IRCF0 | ICSL | ICSS | — | — | --10 00-- | --10 uu-- |
| 91h | OSCTUNE | — | — | TUN5 | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | --00 0000 | --00 0000 |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 93h(3) | SSPMSK | Synchronous Serial Port ($I^2C$ mode) Address Mask Register | | | | | | | | 1111 1111 | 1111 1111 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 95h | WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 | 1111 1111 | 1111 1111 |
| 96h | IOCB | IOCB7 | IOCB6 | IOCB5 | IOCB4 | IOCB3 | IOCB2 | IOCB1 | IOCB0 | 0000 0000 | 0000 0000 |
| 97h | T3CON | TMR3CS1 | TMR3CS0 | T3CKPS1 | T3CKPS0 | — | T3SYNC | — | TMR3ON | 0000 -0-0 | uuuu -u-u |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| 9Ah | TMR3L | Holding Register for the Least Significant Byte of the 16-bit TMR3 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Bh | TMR3H | Holding Register for the Most Significant Byte of the 16-bit TMR3 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Ch | APFCON | — | — | — | — | — | — | SSSEL | CCP2SEL | ---- --00 | ---- --00 |
| 9Dh | FVRCON | FVRRDY | FVREN | — | — | CDAFVR1 | CDAFVR0 | ADFVR1 | ADFVR0 | x000 0000 | x000 0000 |
| 9Eh | T3GCON | TMR3GE | T3GPOL | T3GTM | T3GSPM | T3GGO/$\overline{DONE}$ | T3GVAL | T3GSS1 | T3GSS0 | 0000 0x00 | uuuu uxuu |
| 9Fh | ADCON1 | — | ADCS2 | ADCS1 | ADCS0 | — | — | ADREF1 | ADREF0 | -000 --00 | -000 --00 |

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

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.
**2:** These registers can be addressed from any bank.
**3:** Accessible only when SSPM<3:0> = 1001.

**TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 100h(2) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 101h | TMR0 | Timer0 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 102h(2) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 103h(2) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 104h() | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | TACON | TMRAON | — | TACS | TASE | TAPSA | TAPS2 | TAPS1 | TAPS0 | 0-00 0000 | 0-00 0000 |
| 106h | CPSBCON0 | CPSBON | CPSBRM | — | — | CPSBRNG1 | CPSBRNG0 | CPSBOUT | TBXCS | 00-- 0000 | 00-- 0000 |
| 107h | CPSBCON1 | — | — | — | — | CPSBCH3 | CPSBCH2 | CPSBCH1 | CPSBCH0 | ---- 0000 | ---- 0000 |
| 108h | CPSACON0 | CPSAON | CPSARM | — | — | CPSARNG1 | CPSARNG0 | CPSAOUT | TAXCS | 0--- 0000 | 0--- 0000 |
| 109h | CPSACON1 | — | — | — | — | CPSACH3 | CPSACH2 | CPSACH1 | CPSACH0 | ---- 0000 | ---- 0000 |
| 10Ah(1),(2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 10Bh(2) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 10Ch | PMDATL | Program Memory Read Data Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Dh | PMADRL | Program Memory Read Address Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Eh | PMDATH | — | — | Program Memory Read Data Register High Byte | | | | | | --xx xxxx | --uu uuuu |
| 10Fh | PMADRH | — | — | — | Program Memory Read Address Register High Byte | | | | | ---x xxxx | ---u uuuu |
| 110h | TMRA | TimerA Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 111h | TBCON | TMRBON | — | TBCS | TBSE | TBPSA | TBPS2 | TBPS1 | TBPS0 | 0-00 0000 | 0-00 0000 |
| 112h | TMRB | TimerB Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 113h | DACCON0 | DACEN | DACLPS | DACOE | — | DACPSS1 | DACPSS0 | — | — | 000- 00-- | 000- 00-- |
| 114h | DACCON1 | — | — | — | DACR4 | DACR3 | DACR2 | DACR1 | DACR0 | ---0 0000 | ---0 0000 |

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

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** These registers can be addressed from any bank.

**3:** Accessible only when SSPM<3:0> = 1001.

**TABLE 2-2: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 3** | | | | | | | | | | | |
| 180h(2) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| 181h | OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 182h(2) | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 183h(2) | STATUS | IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 184h(2) | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | ANSELA | ANSA7 | ANSA6 | ANSA5 | ANSA4 | ANSA3 | ANSA2 | ANSA1 | ANSA0 | 1111 1111 | 1111 1111 |
| 186h | ANSELB | ANSB7 | ANSB6 | ANSB5 | ANSB4 | ANSB3 | ANSB2 | ANSB1 | ANSB0 | 1111 1111 | 1111 1111 |
| 187h | ANSELC | ANSC7 | ANSC6 | ANSC5 | — | — | ANSC2 | ANSC1 | ANSC0 | 111- -111 | 111- -111 |
| 188h | ANSELD | ANSD7 | ANSD6 | ANSD5 | ANSD4 | ANSD3 | ANSD2 | ANSD1 | ANSD0 | 1111 1111 | 1111 1111 |
| 189h | ANSELE | — | — | — | — | — | ANSE2 | ANSE1 | ANSE0 | ---- -111 | ---- -111 |
| 18Ah(1),(2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 18Bh(2) | INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 18Ch | PMCON1 | — | — | — | — | — | — | — | RD | 1--- ---0 | 1--- ---0 |
| 18Dh | — | Reserved | | | | | | | | — | — |
| 18Eh | — | Reserved | | | | | | | | — | — |
| 18Fh | — | Reserved | | | | | | | | — | — |

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

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** These registers can be addressed from any bank.

**3:** Accessible only when SSPM<3:0> = 1001.

#### 2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

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 23.0 "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 2-1: STATUS: STATUS REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC(1) | C(1) |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h-1FFh)
`0` = Bank 0, 1 (00h-FFh)

bit 6-5 **RP<1:0>:** Register Bank Select bits (used for direct addressing)
`00` = Bank 0 (00h-7Fh)
`01` = Bank 1 (80h-FFh)
`10` = Bank 2 (100h-17Fh)
`11` = Bank 3 (180h-1FFh)

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/$\overline{\text{Digit Borrow}}$ 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.

### 2.2.2.2 OPTION Register

The OPTION register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

- Timer0/WDT prescaler
- External RB0/INT interrupt
- Timer0
- Weak pull-ups on PORTB

**Note:** To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit of the OPTION register to '1'. Refer to **Section 13.3 "Timer1/3 Prescaler"**.

**REGISTER 2-2: OPTION_REG: OPTION REGISTER**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual bits in the WPUB register

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **TMR0CS:** Timer0 Clock Source Select bit
1 = Transition on RA4/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 RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

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

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

### 2.2.2.3 PCON Register

The Power Control (PCON) register contains flag bits (refer to Table 3-4) to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)

The PCON register bits are shown in Register 2-3.

**REGISTER 2-3: PCON: POWER CONTROL REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-q | R/W-q |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ |
| bit 7 | | | | | | | bit 0 |

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

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

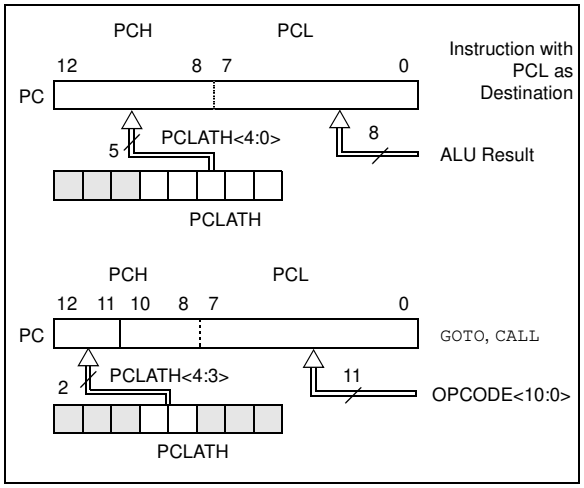
bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** 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)

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-2 shows the two situations for the loading of the PC. The upper example in Figure 2-2 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-2 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-2: LOADING OF PC IN DIFFERENT SITUATIONS**




### 2.3.1 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).

### 2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (refer to Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is 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. This means that after the stack has been PUSHed eight times, the ninth PUSH overwrites the value that was stored from the first PUSH. The tenth PUSH overwrites the second PUSH (and so on).

**Note 1:** There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.

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

## 2.4 Program Memory Paging

All devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper two bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is POPed off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

**Note:** The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

**EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0**

```
        ORG 500h
        PAGESEL SUB_P1  ;Select page 1
                        ;(800h-FFFh)
        CALL    SUB1_P1 ;Call subroutine in
        :               ;page 1 (800h-FFFh)
        :
        ORG     900h    ;page 1 (800h-FFFh)
SUB1_P1
        :               ;called subroutine
                        ;page 1 (800h-FFFh)
        :
        RETURN          ;return to
                        ;Call subroutine
                        ;in page 0
                        ;(000h-7FFh)
```