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

# PIC16HV540

## Enhanced PIC16C54 EPROM-Based 8-Bit CMOS Microcontroller With On-Chip Voltage Regulator

### High-Performance RISC CPU:

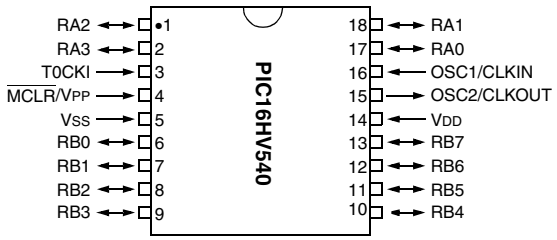
- Only 33 single word instructions to learn
- All instructions are single cycle (200 ns) except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 12-bit wide instructions
- 8-bit wide data path
- Seven special function hardware registers
- Four-level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

### Peripheral Features:

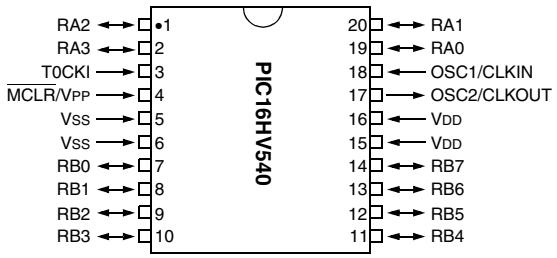
- 8-bit real time clock/counter (TMR0) with 8-bit programmable prescaler
- Power-On Reset (POR)
- Brown-Out Protection
- Device Reset Timer (DRT) with short RC oscillator start-up time
- Programmable Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Sleep Timer
- 8 High Voltage I/O
- 4 Regulated I/O
- Wake up from SLEEP on-pin change
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options:
  - RC: Low-cost RC oscillator
  - XT: Standard crystal/resonator
  - HS: High speed crystal/resonator
  - LP: Power saving, low frequency crystal
- Glitch filtering on $\overline{\text{MCLR}}$ and pin change inputs

### Pin Configurations

**PDIP, SOIC, Windowed CERDIP**




**SSOP**




### CMOS Technology:

- Selectable on-chip 3V/5V Regulator
- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range:
  - 3.5V to 15V
- Temperature range:
  - Commercial: 0°C to 70°C
  - Industrial: -40°C to 85°C
- Low-power consumption
  - < 2 mA typical @ 5V, 4 MHz
  - 15 μA typical @ 3V, 32 kHz
  - < 4.5 μA typical standby current @ 15V (with WDT disabled), 0°C to 70°C

## Table of Contents



*To Our Valued Customers*

**Most Current Data Sheet**

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

http://www.microchip.com

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

**Errata**

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

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

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

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

**Corrections to this Data Sheet**

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at webmaster@microchip.com.

We appreciate your assistance in making this a better document.

## 1.0 GENERAL DESCRIPTION

The PIC16HV540 from Microchip Technology is a low-cost, high-performance, 8-bit, fully-static, EPROM-based CMOS microcontroller. It is pin and software compatible with the PIC16C5X family of devices. It employs a RISC architecture with only 33 single word/single cycle instructions. All instructions are single cycle except for program branches, which take two cycles. The PIC16HV540 delivers performance an order of magnitude higher than its competitors in the same price category. The 12-bit wide instructions are highly orthogonal resulting in 2:1 code compression over other 8-bit microcontrollers in its class. The easy-to-use and easy-to-remember instruction set reduces development time significantly.

The PIC16HV540 is the first One-Time-Programmable (OTP) microcontroller with an on-chip 3 volt and 5 volt regulator. This eliminates the need for an external regulator in many applications powered from 9 Volt or 12 Volt batteries or unregulated 6 volt, 9 volt or 12 volt mains adapters. The PIC16HV540 is ideally suited for applications that require very low standby current at high voltages. These typically require expensive low current regulators.

The PIC16HV540 is equipped with special features that reduce system cost and power requirements. The Power-On Reset (POR) and Device Reset Timer (DRT) eliminate the need for external reset circuitry. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator, cost saving RC oscillator, and XT and HS for crystal oscillators. Power saving SLEEP mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The UV erasable CERDIP packaged versions are ideal for code development, while the cost-effective OTP versions are suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in OTP microcontrollers, while benefiting from the OTP's flexibility.

The PIC16HV540 is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer, and a full featured programmer. All the tools are supported on IBM® PC and compatible machines.

### 1.1 Applications

The PIC16HV540 fits in low-power battery applications such as CO and smoke detection, toys, games, security systems and automobile modules. The EPROM technology makes customizing of application programs (transmitter codes, receiver frequencies, etc.) extremely fast and convenient. The small footprint package, for through hole or surface mounting, make this microcontroller suitable for applications with space limitations. Low-cost, low-power, high-performance, ease of use and I/O flexibility make the PIC16HV540 very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, replacement of "glue" logic in larger systems, coprocessor applications).

### 1.2 Enhanced Features

#### 1.2.1 REGULATED I/O PORTA INDEPENDENT OF CORE REGULATOR

PORTA I/O pads and OSC2 output are powered by the regulated internal voltage $V_{IO}$. A maximum of 10mA per output is allowed, or a total of 40mA. The core itself is powered from the independently regulated supply $V_{REG}$.

#### 1.2.2 HIGH VOLTAGE I/O PORTB

All eight PORTB I/Os are high voltage I/O. The inputs will tolerate input voltages as high as the $V_{DD}$ and outputs will swing from $V_{SS}$ to the $V_{DD}$. The input threshold voltages vary with supply voltage. (See Electrical Characteristics.)

#### 1.2.3 WAKE-UP ON PIN CHANGE ON PORTB [0:3]

Four of the PORTB inputs latch the status of the pin at the onset of sleep mode. A level change on the inputs resets the device, implementing wake up on pin change (via warm reset). The $\overline{\text{PCWUF}}$ bit in the status register is reset to indicate that a pin change caused the reset condition. Any pin change (glitch insensitive) of the opposite level of the initial value wakes up the device. This option can be enabled/disabled in OPTION2 register. (See OPTION2 Register, Register 4-3.)

#### 1.2.4 WAKE-UP ON PIN CHANGE WITH A SLOWLY-RISING VOLTAGE ON PORTB [7]

PORTB [7] also implements wake up from sleep, however this input is specifically adapted so that a slowly **rising** voltage does not cause excessive power consumption. This input can be used with external RC circuits for long sleep periods without using the internal timer and prescaler. This option is also enabled/disabled in OPTION2 register. (The enable/disable bit is shared with the other 4 wake-up inputs.) The $\overline{\text{PCWUF}}$ bit in the status register is also shared with the other four wake-up inputs.

#### 1.2.5 LOW-VOLTAGE (BROWN-OUT) DETECTION

A low voltage (Brown-out) detect circuit optionally resets the device at a voltage level higher than that at which the PICmicro® device stops operating. The nominal trip voltages are 3.1 volts (for 5 volt operation) and 2.2 volt (for 3 volt operation), respectively. The core remains in the reset state as long as this condition holds (as if a $\overline{\text{MCLR}}$ external reset was given). The Brown-out trip level is user selectable, with built-in interlocks. The Brown-out detector is disabled at power-up and is activated by clearing the appropriate bit ($\overline{\text{BODEN}}$) in OPTION2 register.

### 1.2.6 INCREASED STACK DEPTH

The stack depth is 4 levels to allow modular program implementation by using functions and subroutines.

### 1.2.7 ENHANCED WATCHDOG TIMER (WDT) OPERATION

The WDT is enabled by setting FUSE 2 in the configuration word. The WDT setting is latched and the fuse disabled during SLEEP mode to reduce current consumption.

If the WDT is disabled by FUSE 2, it can be enabled/disabled under program control using bit 4 in OPTION2 Register (SWDTEN). The software WDT control is disabled at power-up.

The current consumption of the on-chip oscillator (used for the watchdog, oscillator startup timer and sleep timer) is less than 1μA (typical) at 3 Volt operation.

### 1.2.8 REDUCED EXTERNAL RC OSCILLATOR STARTUP TIME

If the RC oscillator option is selected in the Configuration word (FOSC1=1 and FOSCO=1), the oscillator startup time is 1.0 ms nominal instead of 18 ms nominal. This is applicable after power-up (POR), either WDT interrupt or wake-up, external reset on MCLR, PCWU (wake on pin change) and Brown-out.

### 1.2.9 LOW-VOLTAGE OPERATION OF THE ENTIRE CPU DURING SLEEP

The voltage regulator can automatically lower the voltage to the core from 5 Volt to 3 Volt during sleep, resulting in reduced current consumption. This is an option bit (SL) in the OPTION2 register.

### 1.2.10 GLITCH FILTERS ON WAKE-UP PINS AND MCLR

Glitch sensitive inputs for wake-up on pin change are filtered to reduce susceptibility to interference. A similar filter reduces false reset on MCLR.

### 1.2.11 PROGRAMMABLE CLOCK GENERATOR

When used in RC mode, the CLKOUT pin can be used as a programmable clock output. The output is connected to TMR0, bit 0 and by setting the prescaler, clock out frequencies of CLKIN/8 to CLKIN/1024 can be generated. The CLKOUT pin can also be used as a general purpose output by modifying TMR0, bit 0.

**TABLE 1-1: PIC16HV540 DEVICE**

| | | PIC16HV540 |
|---|---|---|
| **Clock** | Maximum Frequency (MHz) | 20 |
| **Memory** | EPROM Program Memory | 512 |
| | RAM Data Memory (bytes) | 25 |
| **Peripherals** | Timer Module(s) | TMR0 |
| **Packages** | I/O Pins | 12 |
| | Voltage Range (Volts) | 3.5V-15V |
| | Number of Instructions | 33 |
| | Packages | 18-pin DIP<br>SOIC<br>20-pin SSOP |

All PICmicro® devices have Power-on Reset, selectable WDT, selectable code protect and high I/O current capability.

## 2.0 PIC16HV540 DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in this section. When placing orders, please use the PIC16HV540 Product Identification System at the back of this data sheet to specify the correct part number.

For the PIC16HV540 family of devices, there is one device type, as indicated in the device number:

1. **HV**, as in PIC16HV540. These devices have EPROM program memory and operate over the standard voltage range of 3.5 to 15 volts.

### 2.1 UV Erasable Devices

The UV erasable versions, offered in CERDIP packages, are optimal for prototype development and pilot programs.

UV erasable devices can be programmed for any of the four oscillator configurations. Microchip's PICSTART® and PRO MATE® programmers both support programming of the PIC16HV540. Third party programmers also are available; refer to Literature Number DS00104 for a list of sources.

### 2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must be programmed.

### 2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration bit options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. (Please contact your Microchip Technology sales office for more details.)

### 2.4 Serialized Quick-Turnaround-Production (SQTP) Devices

Microchip offers the unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry code, password or ID number. (Please contact your Microchip Technology sales office for more details.)

**NOTES:**

## 3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16HV540 can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16HV540 uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12-bits wide making it possible to have all single word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle (200ns @ 20MHz) except for program branches.

The PIC16HV540 address 512 x 12 of program memory. All program memory is internal.

The PIC16HV540 can directly or indirectly address its register files and data memory. All special function registers including the program counter are mapped in the data memory. The PIC16HV540 has a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16HV540 simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16HV540 device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a borrow and digit borrow out bit, respectively, in subtraction. See the `SUBWF` and `ADDWF` instructions for examples.

A simplified block diagram is shown in Figure 3-1, with the corresponding device pins described in Table 3-1.

**FIGURE 3-1: PIC16HV540 BLOCK DIAGRAM**

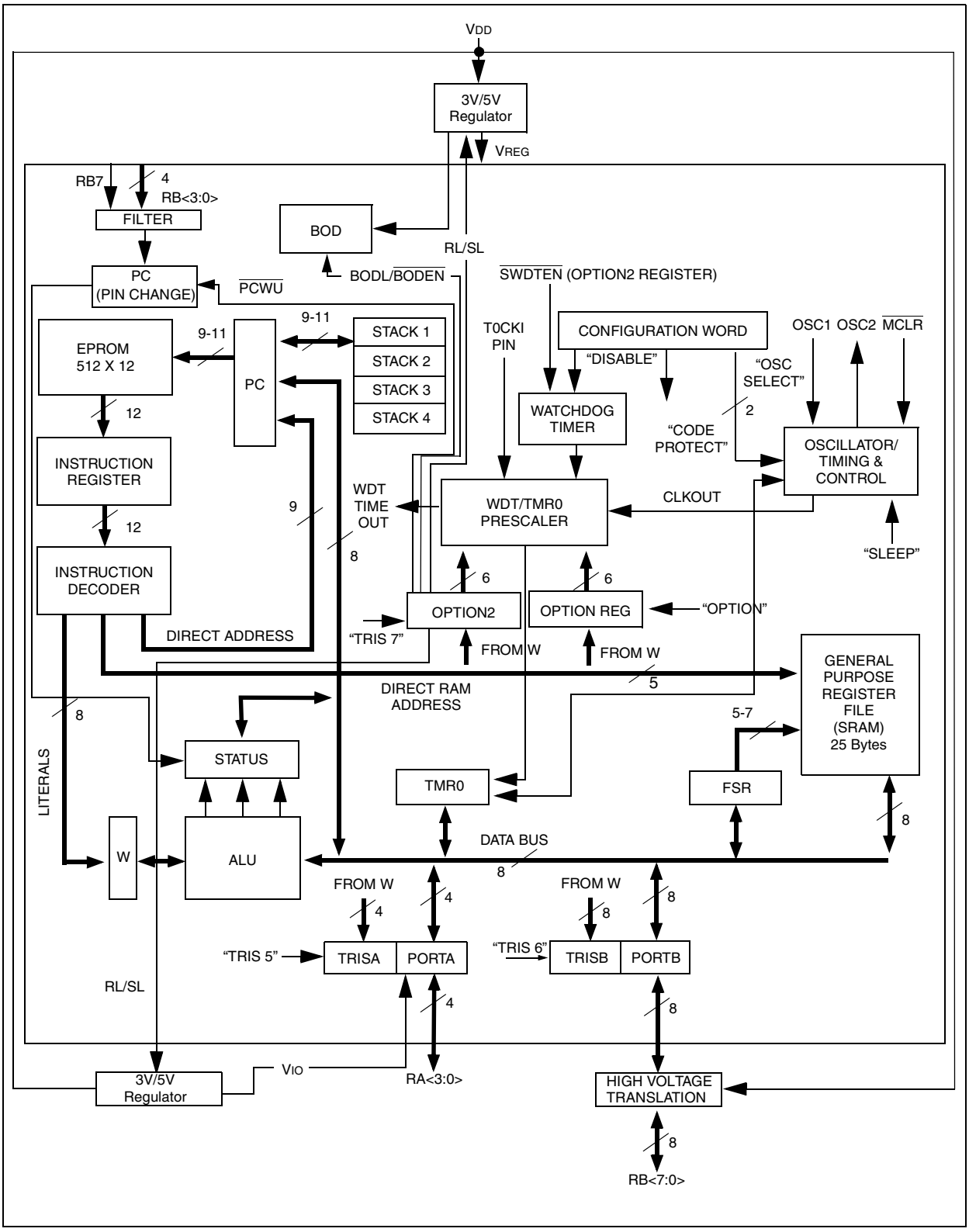


**TABLE 3-1: PINOUT DESCRIPTION - PIC16HV540**

| Name | DIP, SOIC No. | SSOP No. | I/O/P Type | Input Levels | Description | |
|---|---|---|---|---|---|---|
| RA0 | 17 | 19 | I/O | TTL | Independently regulated Bi-directional I/O port — VIO | |
| RA1 | 18 | 20 | I/O | TTL | | |
| RA2 | 1 | 1 | I/O | TTL | | |
| RA3 | 2 | 2 | I/O | TTL | | |
| RB0 | 6 | 7 | I/O | TTL | High-voltage Bi-directional I/O port. Sourced from VDD. | Wake-up on pin change |
| RB1 | 7 | 8 | I/O | TTL | | |
| RB2 | 8 | 9 | I/O | TTL | | |
| RB3 | 9 | 10 | I/O | TTL | | |
| RB4 | 10 | 11 | I/O | TTL | | |
| RB5 | 11 | 12 | I/O | TTL | | |
| RB6 | 12 | 13 | I/O | TTL | | |
| RB7 | 13 | 14 | I/O | TTL | | Wake-up on SLOW rising pin change. |
| T0CKI | 3 | 3 | I | ST | Clock input to Timer 0. Must be tied to VSS or VDD, if not in use, to reduce current consumption. | |
| $\overline{MCLR}$/VPP | 4 | 4 | I | ST | Master clear (reset) input/programming voltage input. This pin is an active low reset to the device. Voltage on the $\overline{MCLR}$/VPP pin must not exceed VDD(1) to avoid unintended entering of programming mode. | |
| OSC1/CLKIN | 16 | 18 | I | ST | Oscillator crystal input/external clock source input. | |
| OSC2/CLKOUT | 15 | 17 | O | — | Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2/CLKOUT output is connected to TMR0, bit 0. Frequencies of CLKIN/8 to CLKIN/1024 can be generated on this pin. | |
| VDD | 14 | 15,16 | P | — | Positive supply. | |
| VSS | 5 | 5,6 | P | — | Ground reference. | |

Legend: I = input, O = output, I/O = input/output, P = power, — = Not Used, TTL = TTL input, ST = Schmitt Trigger input.

**Note 1:** VDD during programming mode can not exceed parameter PD1 called out in the PIC16C5X Programming Specification (Literature number DS30190).

### 3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter is incremented every Q1, and the instruction is fetched from program memory and latched into instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2 and Example 3-1.

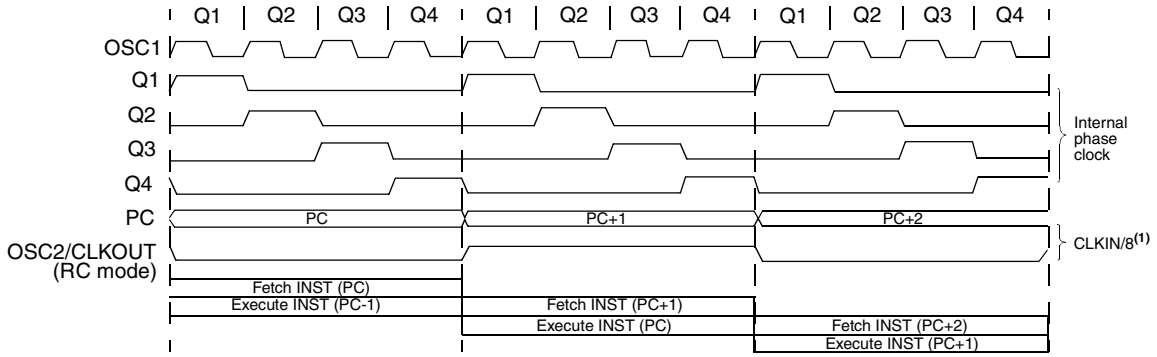
### 3.2 Instruction Flow/Pipelining

An Instruction Cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., `GOTO`) then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

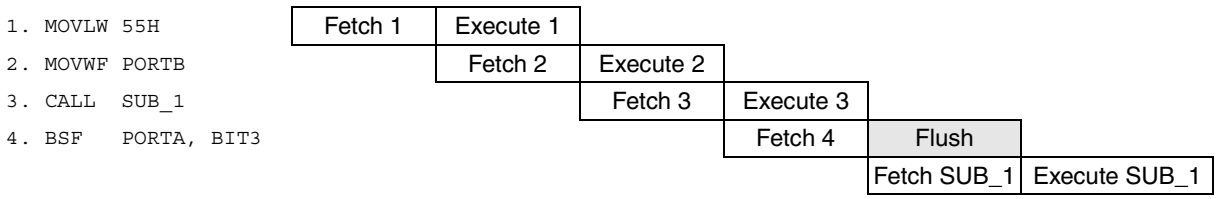
In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

**FIGURE 3-2: CLOCK/INSTRUCTION CYCLE**




**Note 1:** Frequencies of CLKIN8 to CLKIN/1024 are possible.

**EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW**




All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

## 4.0 MEMORY ORGANIZATION

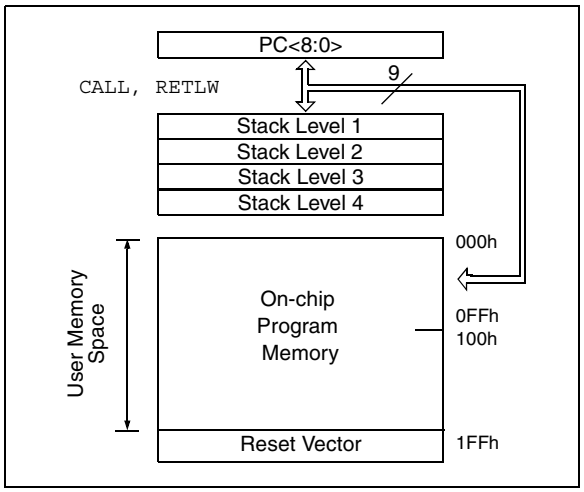
PIC16HV540 memory is organized into program memory and data memory. For devices with more than 512 bytes of program memory, a paging scheme is used. Program memory pages are accessed using one or two STATUS register bits. For devices with a data memory register file of more than 32 registers, a banking scheme is used. Data memory banks are accessed using the File Selection Register (FSR).

### 4.1 Program Memory Organization

The PIC16HV540 has a 9-bit Program Counter (PC) capable of addressing a 512 x 12 program memory space (Figure 4-1). Accessing a location above the physically implemented address will cause a wrap-around.

The reset vector for the PIC16HV540 is at 1FFh. A NOP at the reset vector location will cause a restart at location 000h.

**FIGURE 4-1: PIC16HV540 PROGRAM MEMORY MAP AND STACK**




### 4.2 Data Memory Organization

Data memory is composed of registers, or bytes of RAM. Therefore, data memory for a device is specified by its register file. The register file is divided into two functional groups: special function registers and general purpose registers.

The special function registers include the TMR0 register, the Program Counter (PC), the Status Register, the I/O registers (ports), and the File Select Register (FSR). In addition, special purpose registers are used to control the I/O port configuration and prescaler options.

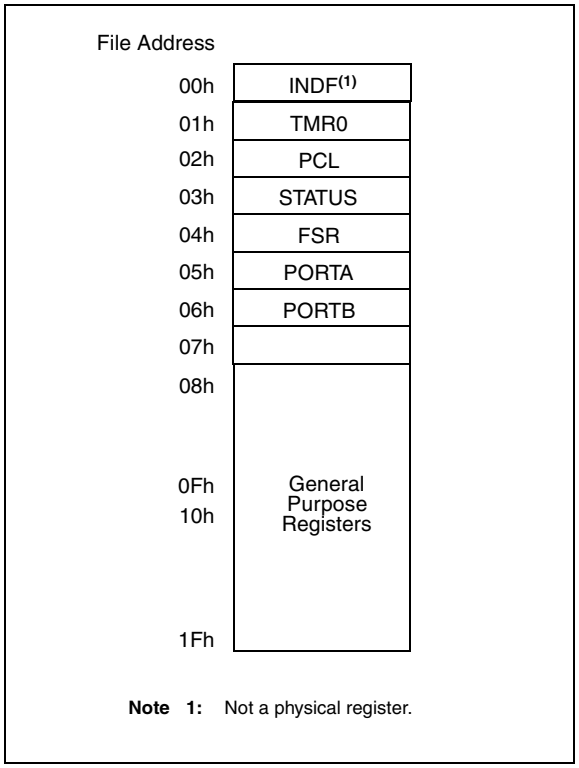
The general purpose registers are used for data and control information under command of the instructions.

For the PIC16HV540, the register file is composed of 10 special function registers and 25 general purpose registers (Figure 4-2).

#### 4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly or indirectly through the file select register FSR (Section 4.8).

**FIGURE 4-2: PIC16HV540 REGISTER FILE MAP**




**Note 1:** Not a physical register.

#### 4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions to control the operation of the device (Table 4-1).

The special registers can be classified into two sets. The special function registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section for each peripheral feature.

**TABLE 4-1: SPECIAL FUNCTION REGISTER SUMMARY**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-On Reset | Value on $\overline{MCLR}$ and WDT Reset | Value on Wake-up on Pin Change | Value on Brown-Out Reset |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N/A | TRIS | I/O control registers (TRISA, TRISB) | | | | | | | | 1111 1111 | 1111 1111 | 1111 1111 | 1111 1111 |
| N/A | OPTION | Contains control bits to configure Timer0 and Timer0/WDT prescaler | | | | | | | | --11 1111 | --11 1111 | --11 1111 | --11 1111 |
| N/A | OPTION2 | Contains control bits to configure pin changes, software enabled WDT, regulation and brown-out | | | | | | | | --11 1111 | --uu uuuu | --uu uuuu | --xx xxxx |
| 00h | INDF | Uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | uuuu uuuu | uuuu uuuu | xxxx xxxx |
| 01h | TMR0 | 8-bit real-time clock/counter | | | | | | | | xxxx xxxx | uuuu uuuu | uuuu uuuu | xxxx xxxx |
| 02h(1) | PCL | Low order 8 bits of PC | | | | | | | | 1111 1111 | 1111 1111 | 1111 1111 | 1111 1111 |
| 03h | STATUS | $\overline{PCWUF}$ | PA1 | PA0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 1001 1xxx | 100q quuu | 000u uuuu | x00x xxxx |
| 04h | FSR | Indirect data memory address pointer | | | | | | | | 111x xxxx | 111u uuuu | 111u uuuu | 111x xxxx |
| 05h | PORTA | — | — | — | — | RA3 | RA2 | RA1 | RA0 | ---- xxxx | ---- uuuu | ---- uuuu | ---- xxxx |
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu | uuuu uuuu | xxxx xxxx |

Legend: Shaded boxes = unimplemented or unused, – = unimplemented, read as '0' (if applicable)
x = unknown, u = unchanged, q = value depends on condition.

**Note 1:** The upper byte of the Program Counter is not directly accessible. See Section 4.6 of the PIC16HV540 data sheet (DS40197B) for an explanation of how to access these bits.

### 4.3 STATUS Register

This register contains the arithmetic status of the ALU, the RESET status, and the page preselect bits for program memories larger than 512 words.

The STATUS register can be the destination for any instruction, as with 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 while the $\overline{PCWUF}$ bit is a read/write bit. 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` and `MOVWF` instructions be used to alter the STATUS register because these instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions, which do affect STATUS bits, see Section 8.0, Instruction Set Summary.

**REGISTER 4-1: STATUS REGISTER (ADDRESS:03h)**

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| $\overline{PCWUF}$ | PA1 | PA0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
- n = Value at POR reset

bit 7: **$\overline{PCWUF}$**: Pin Change Reset bit
1 = After Power-up Reset (POR) or SLEEP command
0 = After a wake-up on pin change event

bit 6-5: **Not Applicable**

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/$\overline{borrow}$ bit (for `ADDWF` and `SUBWF` instructions)
**ADDWF**
1 = A carry from the 4th low order bit of the result occurred
0 = A carry from the 4th low order bit of the result did not occur
**SUBWF**
1 = A borrow from the 4th low order bit of the result did not occur
0 = A borrow from the 4th low order bit of the result occurred

bit 0: **C**: Carry/$\overline{borrow}$ bit (for `ADDWF`, `SUBWF` and `RRF`, `RLF` instructions)

| **ADDWF** | **SUBWF** | **RRF or RLF** |
|---|---|---|
| 1 = A carry occurred | 1 = A borrow did not occur | Load bit with LSb or MSb, respectively |
| 0 = A carry did not occur | 0 = A borrow occurred | |

## 4.4 OPTION Register

The OPTION register is a 6-bit wide, write-only register which contains various control bits to configure the Timer0/WDT prescaler and Timer0.

By executing the OPTION instruction, the contents of the W register will be transferred to the OPTION register. A RESET sets the OPTION<5:0> bits.

Example 4-1 illustrates how to initialize the OPTION register.

**EXAMPLE 4-1: INSTRUCTIONS FOR INITIALIZING OPTION REGISTER**

```
movlw   '0000 0111'b   ; load OPTION setup value into W
OPTION                 ; initialize OPTION register
```

**REGISTER 4-2: OPTION REGISTER**

| U-0 | U-0 | W-1 | W-1 | W-1 | W-1 | W-1 | W-1 |
|---|---|---|---|---|---|---|---|
| — | — | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit7 | | | | | | | 0 |

W = Writable bit
U = Unimplemented bit
- n = Value at POR reset

bit 7-6: **Unimplemented**

bit 5: **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

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

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 |

## 4.5 OPTION2 Register

The OPTION2 register is a 6-bit wide, write-only register which contains various control bits to configure the added features on the PIC16HV540. A Power-on Reset sets the OPTION2<5:0> bits.

Example 4-2 illustrates how to initialize the OPTION2 register.

| **Note:** | All Power-on Resets will disable the Brown-out Detect circuit. All subsequent resets will not disable the Brown-out Detect if enabled. |
|---|---|

**EXAMPLE 4-2: INSTRUCTIONS FOR INITIALIZING OPTION2 REGISTER**

```
movlw    '0001 0111'b   ; load OPTION2 setup value into W
tris     0x07           ; initialize OPTION2 register
```

**REGISTER 4-3: OPTION2 REGISTER (TRIS 07H)**

| U-0 | U-0 | W-1 | W-1 | W-1 | W-1 | W-1 | W-1 |
|---|---|---|---|---|---|---|---|
| — | — | $\overline{\text{PCWU}}$ | $\overline{\text{SWDTEN}}$ | RL | SL | BODL | $\overline{\text{BODEN}}$ |
| bit7 | | | | | | | 0 |

W = Writable bit
U = Unimplemented bit
- n = Value at POR reset

bit 7-6: **Unimplemented**

bit 5: **$\overline{\text{PCWU}}$**: Wake-up on Pin Change
1 = Disabled
0 = Enabled

bit 4: **$\overline{\text{SWDTEN}}$**: Software Controlled WDT Enable bit
1 = WDT is turned off it the WDTEN configuration bit = 0
0 = WDT is on if the WDTEN configuration bit = 0; if WDTEN bit = 1, then SWDTEN is 'don't care'

bit 3: **RL**: Regulated Voltage Level Select bit
1 = 5 volt
0 = 3 volt

bit 2: **SL:** Sleep Voltage Level Select bit
1 = **RL** bit setting
0 = 3 volt

bit 1: **BODL**: Brown-out Voltage Level Select bit
1 = **RL** bit setting, but **SL** during SLEEP
0 = 3 volt

bit 0: **$\overline{\text{BODEN}}$**: Brown-out Enabled
1 = Disabled
0 = Enabled

## 4.6 Program Counter

As a program instruction is executed, the Program Counter (PC) will contain the address of the next program instruction to be executed. The PC value is increased by one every instruction cycle, unless an instruction changes the PC.

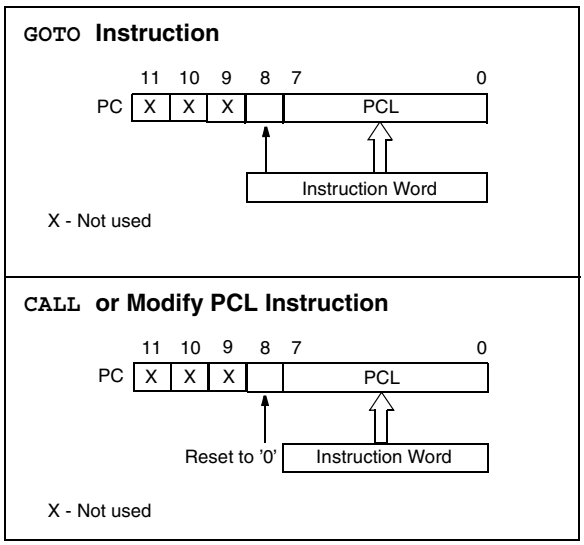
For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. (Figure 4-3).

For a CALL instruction, or any instruction where the PCL is the destination, bits 7:0 of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 4-3).

Instructions where the PCL is the destination, or Modify PCL instructions, include MOVWF PC, ADDWF PC, and BSF PC, 5. .

| | |
|---|---|
| **Note:** | Because PC<8> is cleared in the CALL instruction, or any Modify PCL instruction, all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long). |

**FIGURE 4-3: LOADING OF PC BRANCH INSTRUCTIONS - PIC16HV540**




### 4.6.1 EFFECTS OF RESET

The Program Counter is set upon a RESET, which means that the PC addresses the last location in the last page i.e., the reset vector.

The STATUS register page preselect bits are cleared upon a RESET, which means that page 0 is preselected.

Therefore, upon a RESET, a GOTO instruction at the reset vector location will automatically cause the program to jump to page 0.

## 4.7 Stack

PIC16HV540 device has a 12-bit wide L.I.F.O. (last in, first out) hardware 4 level stack.

A CALL instruction will *push* the current value of stack 1 into stack 2 and then push the current program counter value, incremented by one, into stack level 1. If more than four sequential CALL's are executed, only the most recent four return addresses are stored.

A RETLW instruction will *pop* the contents of stack level 1 into the program counter and then copy stack level 2 contents into level 1. If more than four sequential RETLW's are executed, the stack will be filled with the address previously stored in level 4. Note that the W register will be loaded with the literal value specified in the instruction. This is particularly useful for the implementation of data look-up tables within the program memory.

Upon any reset, the contents of the stack remain unchanged, however the program counter (PCL) will also be reset to 0.

| | |
|---|---|
| **Note 1:** | There are no STATUS bits to indicate stack overflows or stack underflow conditions. |
| **Note 2:** | There are no instructions mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL and RETLW instructions. |

## 4.8 Indirect Data Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

### EXAMPLE 4-3: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected).

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in Example 4-4.

### EXAMPLE 4-4: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

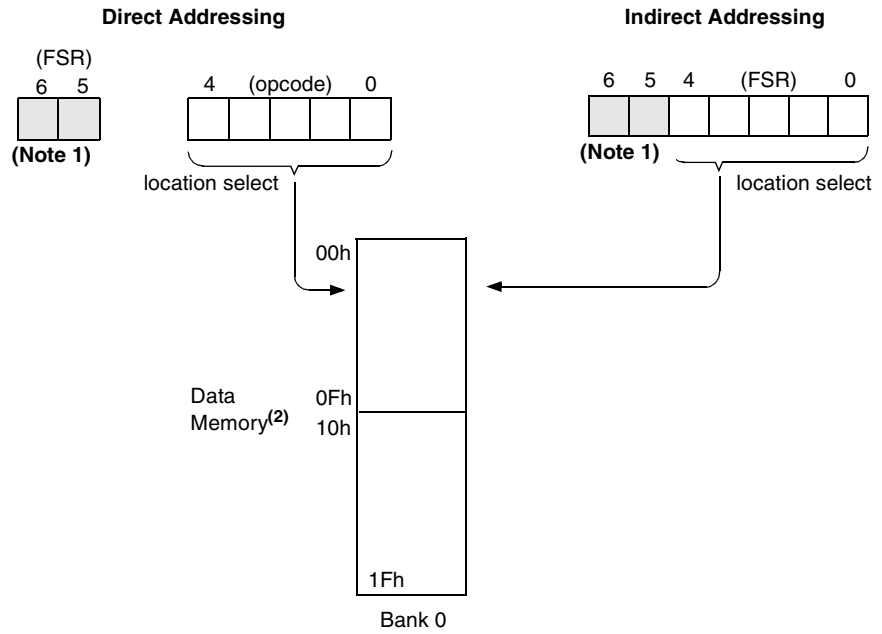
```
            movlw   0x10    ;initialize pointer
            movwf   FSR     ; to RAM
NEXT        clrf    INDF    ;clear INDF register
            incf    FSR,F   ;inc pointer
            btfsc   FSR,4   ;all done?
            goto    NEXT    ;NO, clear next
CONTINUE
            :               ;YES, continue
```

The FSR is a 5-bit (PIC16HV540) wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

**PIC16HV540:** Do not use banking. FSR<6:5> are unimplemented and read as '1's.

### FIGURE 4-4: DIRECT/INDIRECT ADDRESSING




**Note 1:** Bits 5 and 6 are unimplemented and read as 1's.

**2:** For register map detail, see Section 4.2.

**NOTES:**

## 5.0 I/O PORTS

As with any other register, the I/O registers can be written and read under program control. However, read instructions (e.g., `MOVF PORTB,W`) always read the I/O pins independent of the pin's input/output modes. On RESET, all I/O ports are defined as input (inputs are at hi-impedance) since the I/O control registers (TRISA, TRISB) are all set.

### 5.1 PORTA

PORTA is a 4-bit I/O register. Only the low order 4 bits are used (RA3:RA0). Bits 7-4 are unimplemented and read as '0's. The inputs will tolerate input voltages as high as $V_{IO}$ and outputs will swing from $V_{SS}$ to $V_{IO}$. The internal voltage regulator $V_{IO}$ powers PORTA I/O pads. The internal regulator output, $V_{IO}$, is switchable between 3Vdc and 5Vdc, via the (RL) bit in the OPTION2 register.

### 5.2 PORTB

PORTB is an 8-bit I/O register (PORTB<7:0>). All 8 PORTB I/Os are high voltage I/O. The inputs will tolerate input voltages as high as $V_{DD}$ and outputs will swing from $V_{SS}$ to $V_{DD}$. In addition, 5 of the PORTB pins can be configured for the wake-up on change feature. Pins RB0, RB1, RB2 and RB3 latch the state of the pin at the onset of sleep mode. (No "dummy" read of the PORTB pins is required prior to executing the `SLEEP` instruction.) A level change on the input resets the device, implementing wake-up on pin change. The $\overline{\text{PCWUF}}$ bit in the status register is cleared to indicate that a pin change caused the reset. This feature can be enabled/disabled in the OPTION2 register.

PORTB pin RB7 also exhibits this wake-up on pin high feature but is specially adapted for a slow-rising input signal. This special feature prevents excessive power consumption when desiring long sleep periods without using the watchdog timer and prescaler. $\overline{\text{PCWUF}}$ bit in the status register is cleared to indicate that a pin change caused the reset. This feature can be enabled/disabled in the OPTION2 register.

Only pins configured as inputs can cause this wake-up on pin change to occur.

To prevent false wake-up on pin change events on pins RB<0:3>, the pin state must be driven to a logic 1 or logic 0 and not left floating during the "`SLEEP`" state. For pin RB7, the pin state must be driven to logic 0 and allowed to ramp to a logic 1 for correct operation.

### 5.3 TRIS Registers

The output driver control registers are loaded with the contents of the W register by executing the `TRIS f` instruction. A '1' from a TRIS register bit puts the corresponding output driver in a hi-impedance mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer.

| **Note:** | A read of the ports reads the pins, not the output data latches. That is, if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low. |
|---|---|

The TRIS registers are "write-only" and are set (output drivers disabled) upon RESET.

### 5.4 I/O Interfacing

The equivalent circuit for the PORTA and PORTB I/O pins are shown in Figure 5-1 through Figure 5-4. All ports may be used for both input and output operation. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF PORTB, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit (in TRISA, TRISB) must be cleared (= 0). For use as an input, the corresponding TRIS bit must be set. Any I/O pin can be programmed individually as input or output.

**FIGURE 5-1: BLOCK DIAGRAM OF PORTA<0:3> PINS**

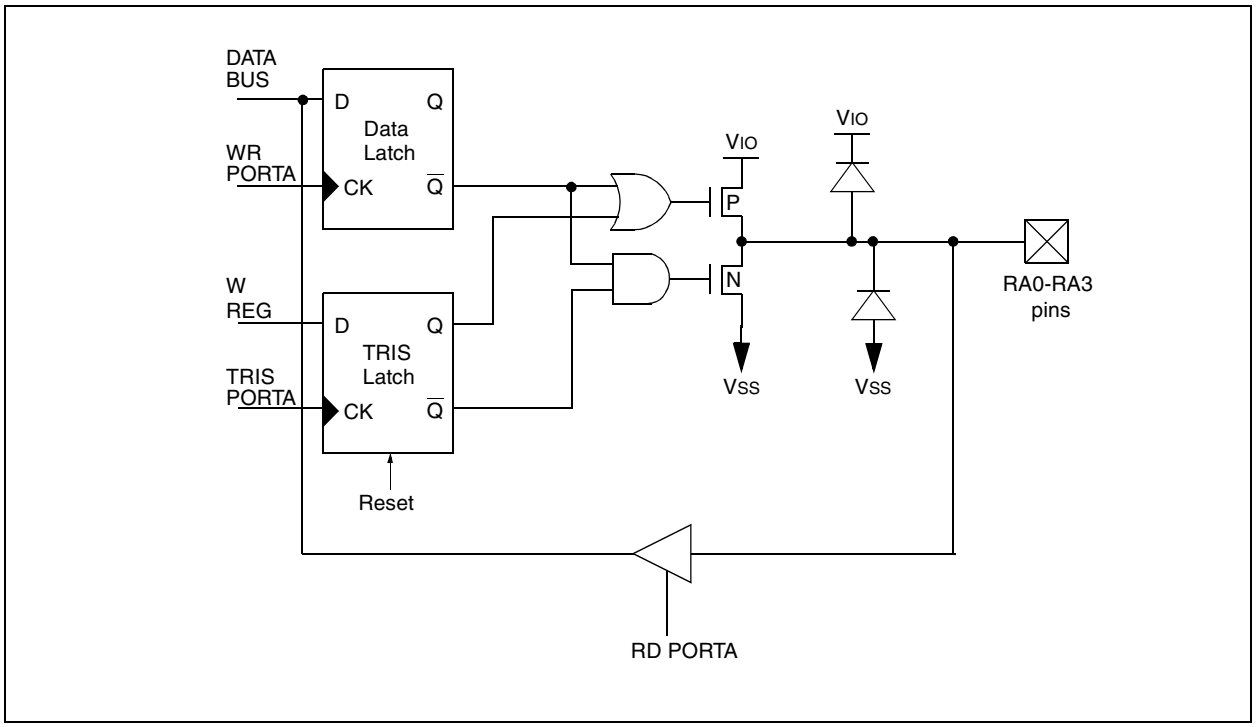



**FIGURE 5-2: BLOCK DIAGRAM OF PORTB<0:3> PINS**

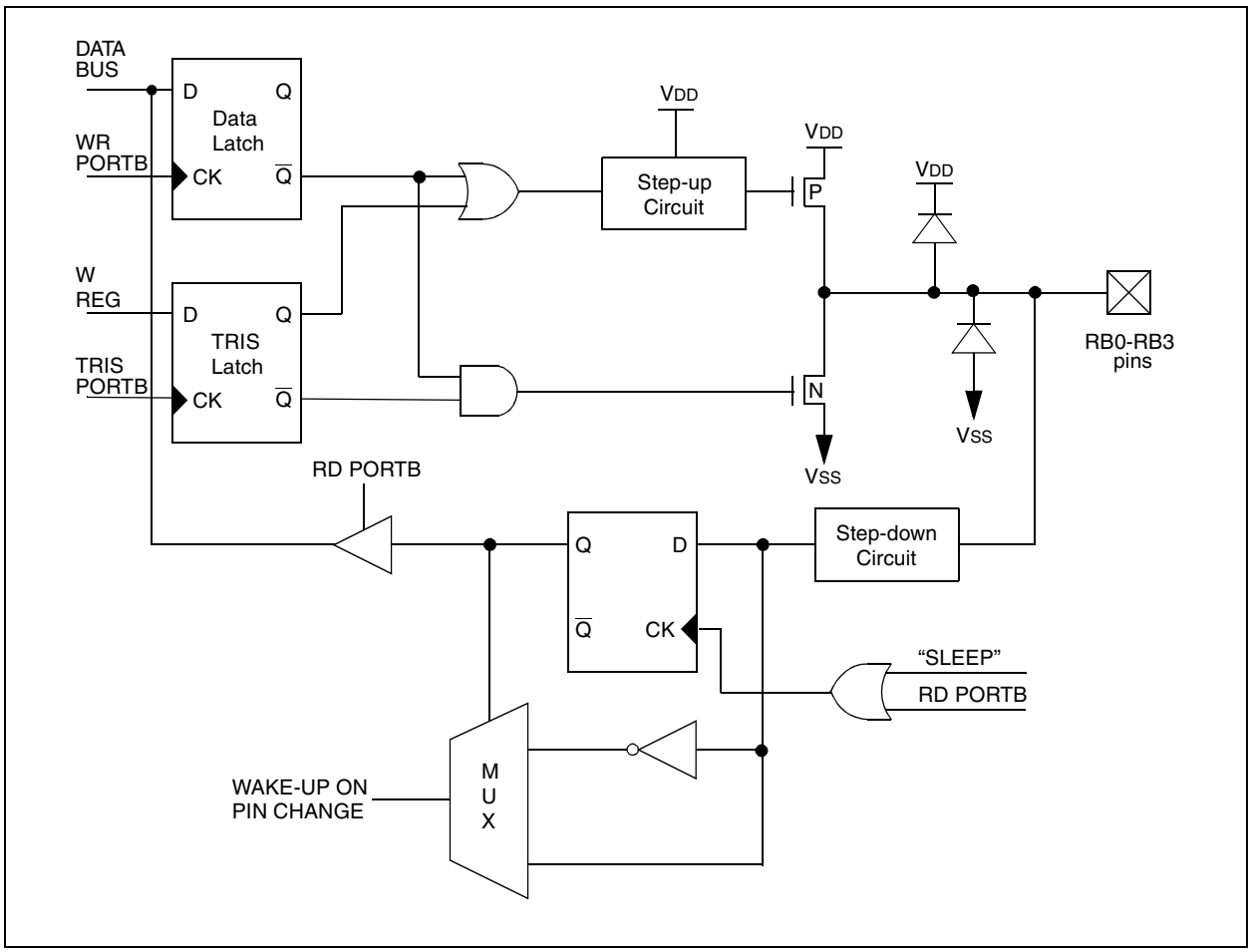


**FIGURE 5-3: BLOCK DIAGRAM OF PORTB<4:6> PINS**

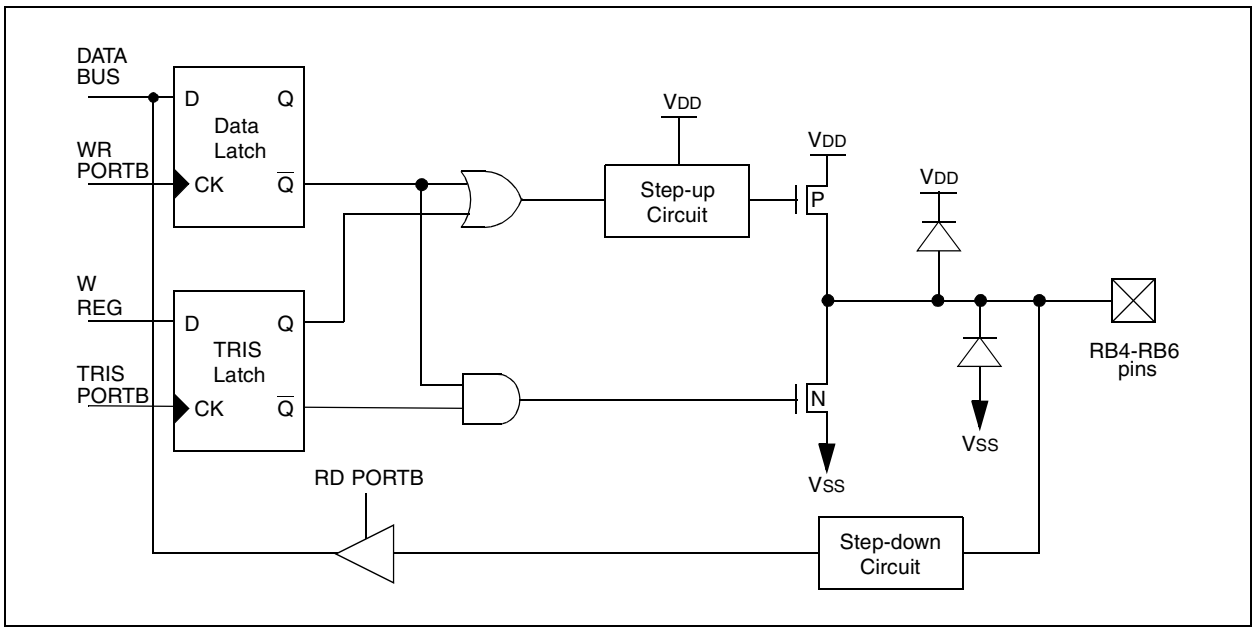



**FIGURE 5-4: BLOCK DIAGRAM OF PORTB<7> PIN**

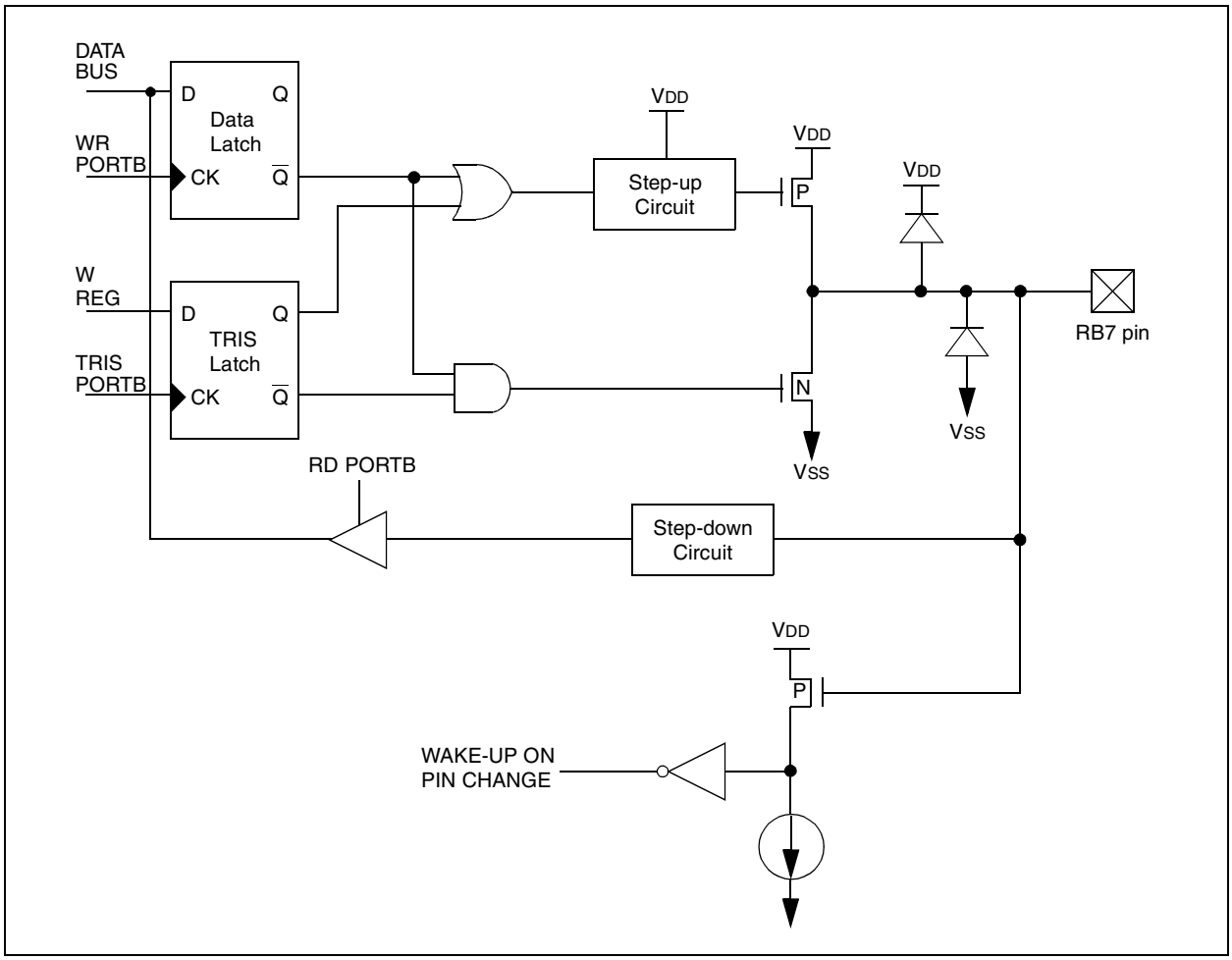


**TABLE 5-1: SUMMARY OF PORT REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on Power-On Reset | Value on $\overline{MCLR}$ and WDT Reset | Value on Wake-up on Pin Change | Value on Brown-Out Reset |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N/A | TRIS | I/O control registers (TRISA, TRISB) | | | | | | | | 1111 1111 | 1111 1111 | 1111 1111 | 1111 1111 |
| 05h | PORTA | — | — | — | — | RA3 | RA2 | RA1 | RA0 | ---- xxxx | ---- uuuu | ---- uuuu | ---- xxxx |
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu | uuuu uuuu | xxxx xxxx |
| 03h | STATUS | $\overline{PCWUF}$ | PA1 | PA0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 100x xxxx | 100q quuu | 000u uuuu | x00x xxxx |
| N/A | OPTION2 | — | — | $\overline{PCWU}$ | $\overline{SWDTEN}$ | RL | SL | BODL | $\overline{BODEN}$ | --11 1111 | --uu uuuu | --uu uuuu | --xx xxxx |

Legend: Shaded boxes = unimplemented, read as '0', —= unimplemented, read as '0', x = unknown, u = unchanged.

## 5.5 I/O Programming Considerations

### 5.5.1 BI-DIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and re-write the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU, bit5 to be set and the PORTB value to be written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (say bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF, etc.) on an I/O port.

A pin actively outputting a high or a low should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

**EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT**

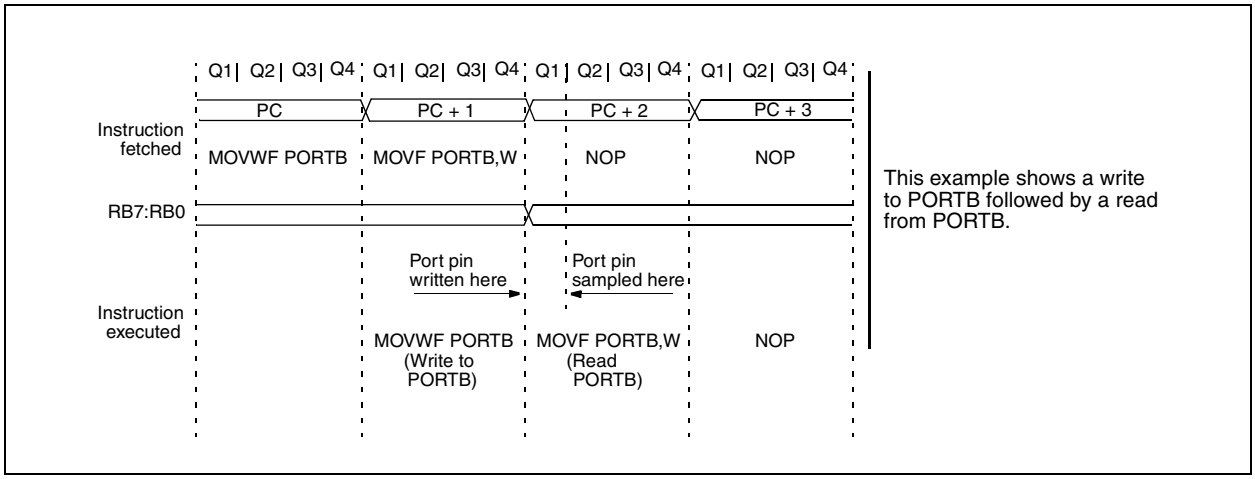
```
;Initial PORT Settings
; PORTB<7:4> Inputs
; PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                   PORT latch  PORT pins
;                    ----------  ----------
  BCF   PORTB, 7    ;01pp pppp   11pp pppp
  BCF   PORTB, 6    ;10pp pppp   11pp pppp
  MOVLW 03Fh        ;
  TRIS  PORTB       ;10pp pppp   10pp pppp
;
;Note that the user may have expected the pin
;values to be 00pp pppp. The 2nd BCF caused
;RB7 to be latched as the pin value (High).
```

### 5.5.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-5). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

**FIGURE 5-5: SUCCESSIVE I/O OPERATION**



**NOTES:**

## 6.0 TIMER0 MODULE AND TMR0 REGISTER

The Timer0 module has the following features:

- 8-bit timer/counter register, TMR0
  - Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
  - Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module, while Figure 6-2 shows the electrical structure of the Timer0 input.

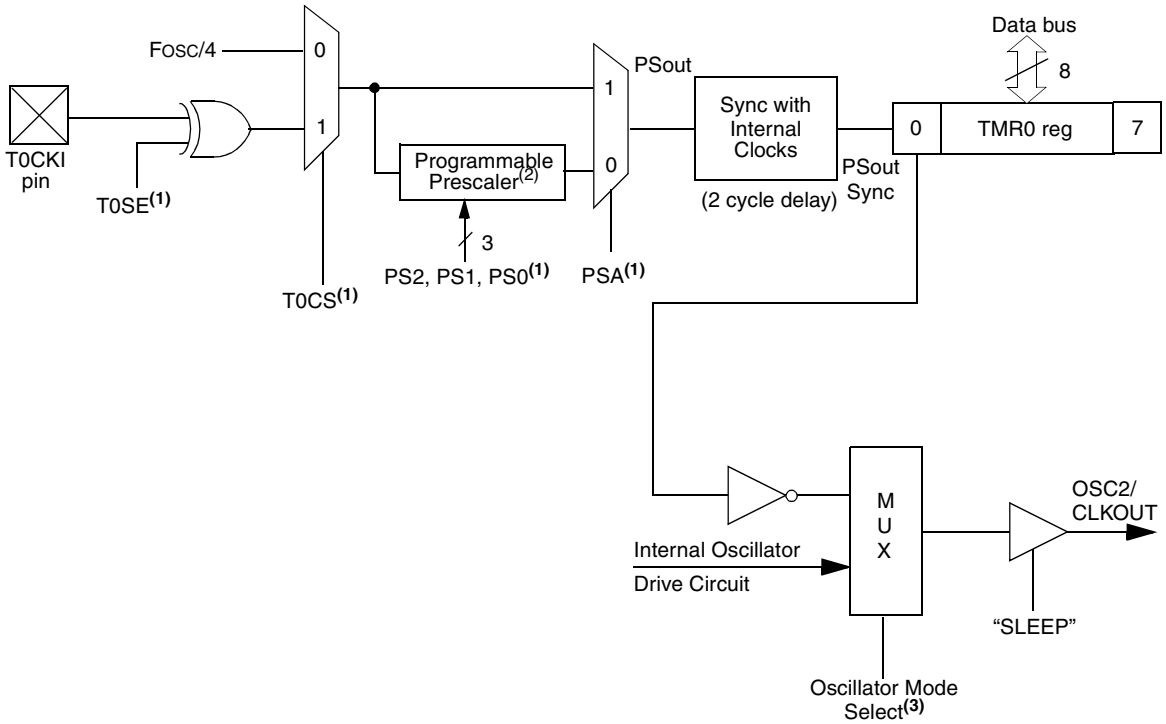
Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles (Figure 6-3 and Figure 6-4). The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The incrementing edge is determined by the source edge select bit T0SE (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.1.

The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4,..., 1:256 are selectable. Section 6.2 details the operation of the prescaler.

A summary of registers associated with the Timer0 module is found in Table 6-1.

**FIGURE 6-1: TIMER0 BLOCK DIAGRAM**




**Note 1:** Bits T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in the OPTION register.
**2:** The prescaler is shared with the Watchdog Timer (Figure 6-6).
**3:** Bit 0 of TMR0 will be output on OSC2/CLKOUT pin when RC oscillator mode is selected.