

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

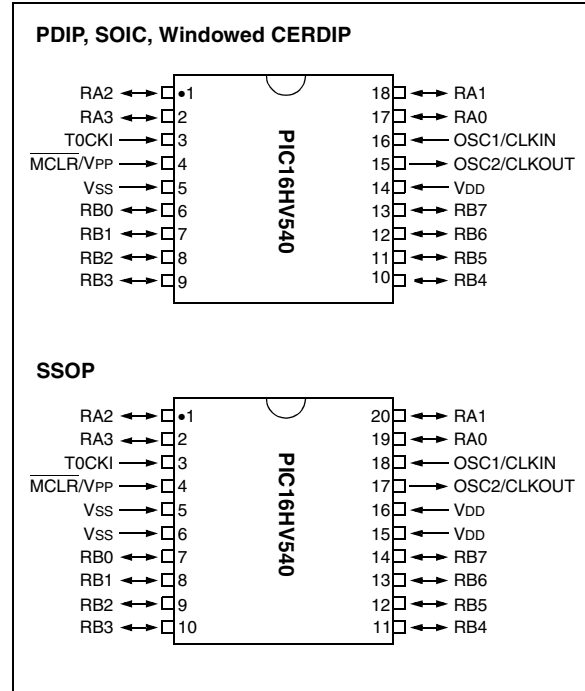
### High-Performance RISC CPU:

- 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 MCLR and pin change inputs

### Pin Configurations



### 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  $\mu$ A typical @ 3V, 32 kHz
  - < 4.5  $\mu$ A typical standby current @ 15V (with WDT disabled), 0°C to 70°C

# PIC16HV540

## Table of Contents

1.0	General Description .....	3
2.0	PIC16HV540 Device Varieties .....	5
3.0	Architectural Overview .....	7
4.0	Memory Organization .....	11
5.0	I/O Ports.....	19
6.0	Timer0 Module and TMR0 Register.....	25
7.0	Special Features of the CPU .....	31
8.0	Instruction Set Summary .....	43
9.0	Development Support .....	55
10.0	Electrical Characteristics - PIC16HV540 .....	61
11.0	DC and AC Characteristics - PIC16HV540.....	69
12.0	Packaging Information .....	73
	Index .....	79
	On-Line Support.....	81
	Reader Response .....	82
	PIC16HV540 Product Identification System.....	83

### *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](mailto: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<sup>®</sup> 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 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 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<sup>®</sup> 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  $\overline{MCLR}$  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 ( $\overline{BODEN}$ ) in OPTION2 register.

# PIC16HV540

## 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 1 $\mu$ A (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
<b>Clock</b>	Maximum Frequency (MHz)	20
<b>Memory</b>	EPROM Program Memory	512
	RAM Data Memory (bytes)	25
<b>Peripherals</b>	Timer Module(s)	TMR0
<b>Packages</b>	I/O Pins	12
	Voltage Range (Volts)	3.5V-15V
	Number of Instructions	33
	Packages	18-pin DIP SOIC 20-pin SSOP

All PICmicro<sup>®</sup> 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.)

# PIC16HV540

---

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.

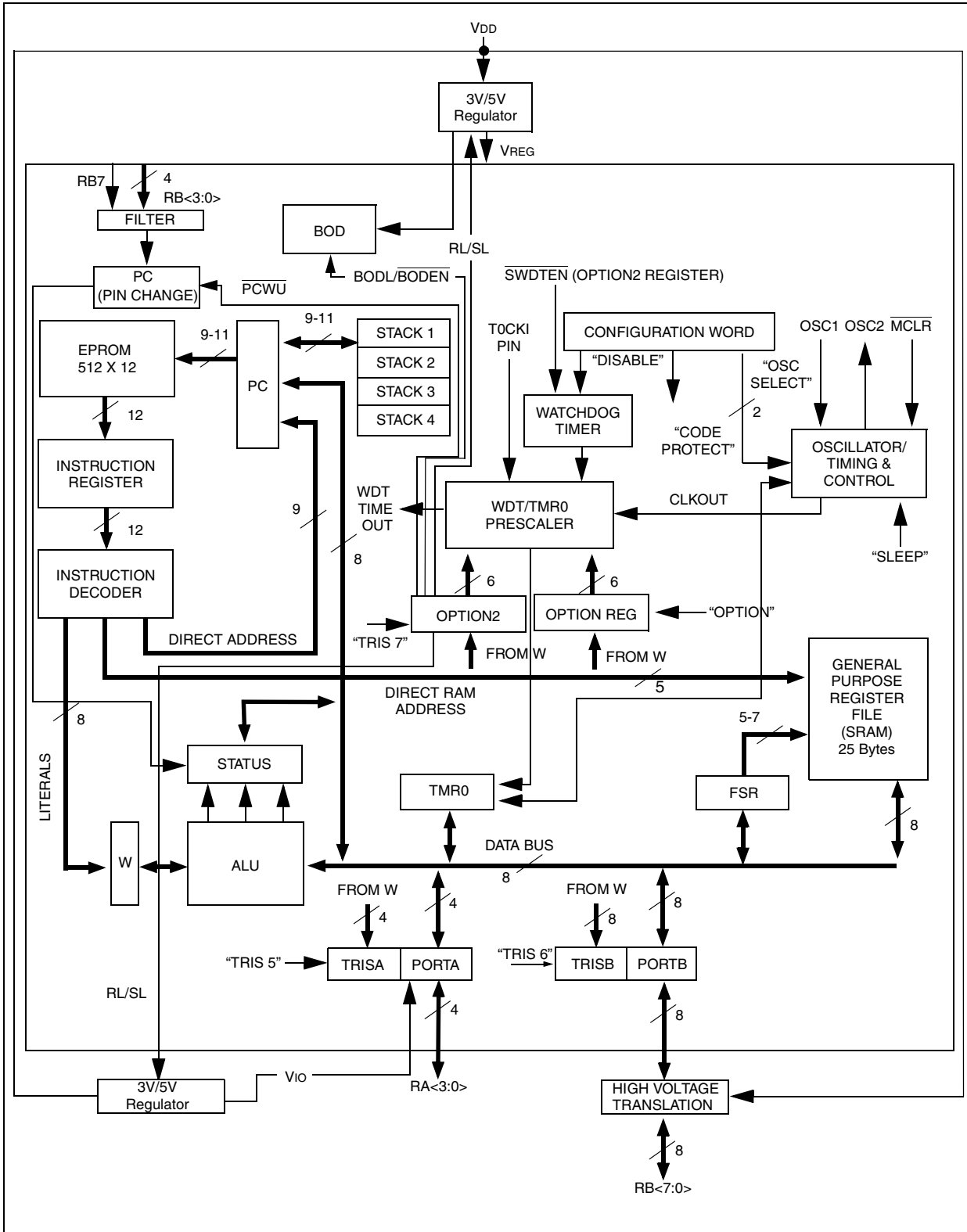
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

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.

# PIC16HV540

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 RA1 RA2 RA3	17 18 1 2	19 20 1 2	I/O I/O I/O I/O	TTL TTL TTL TTL	Independently regulated Bi-directional I/O port — V <sub>IO</sub>	
RB0 RB1 RB2 RB3	6 7 8 9	7 8 9 10	I/O I/O I/O I/O	TTL TTL TTL TTL	High-voltage Bi-directional I/O port. Sourced from V <sub>DD</sub> .	Wake-up on pin change
RB4 RB5 RB6	10 11 12	11 12 13	I/O I/O I/O	TTL TTL 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 V <sub>SS</sub> or V <sub>DD</sub> , if not in use, to reduce current consumption.	
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 MCLR/VPP pin must not exceed V <sub>DD</sub> <sup>(1)</sup> 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:** V<sub>DD</sub> during programming mode can not exceed parameter PD1 called out in the PIC16C5X Programming Specification (Literature number DS30190).

# PIC16HV540

## 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.

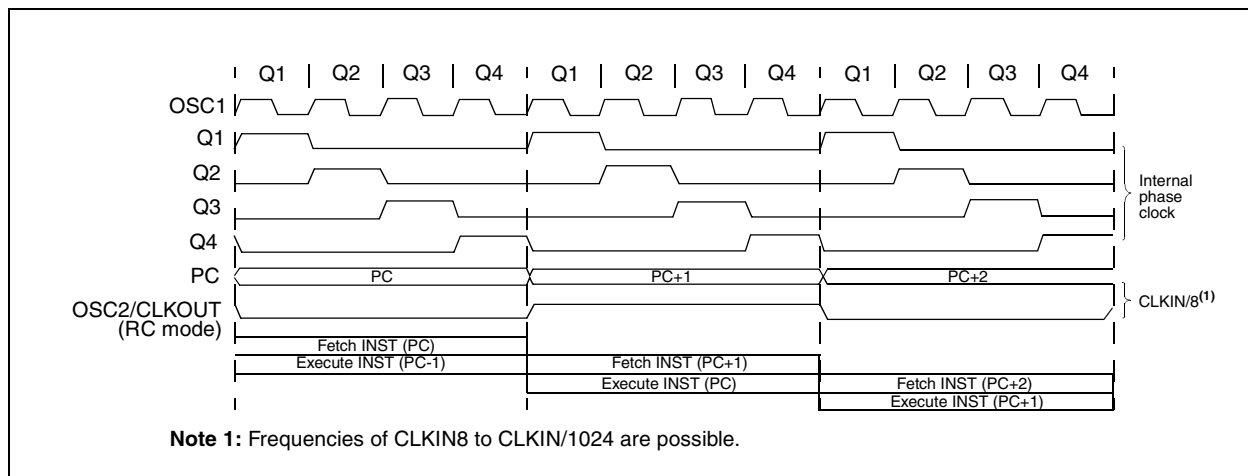
## 3.2 Instruction Flow/Pipelining

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).

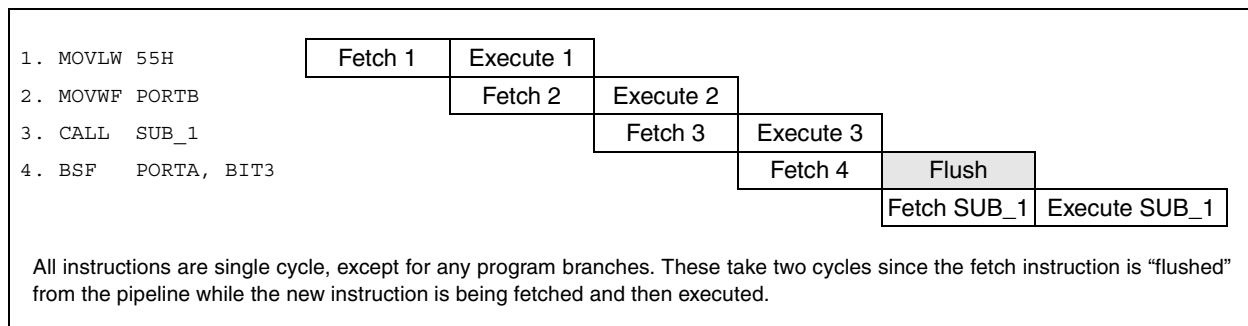
A fetch cycle begins with the program counter (PC) incrementing in Q1.

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**



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



## 4.0 MEMORY ORGANIZATION

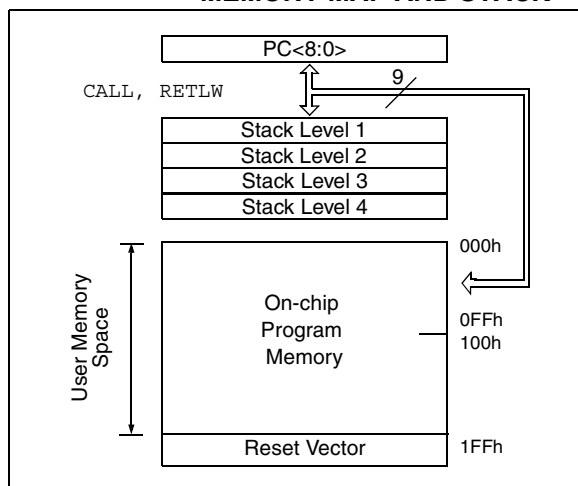
PIC16HV540 memory is organized into program memory and data memory. For devices with more than 512 bytes of program memory, a paging scheme is used. Program memory pages are accessed using one or two STATUS register bits. For devices with a data memory register file of more than 32 registers, a banking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

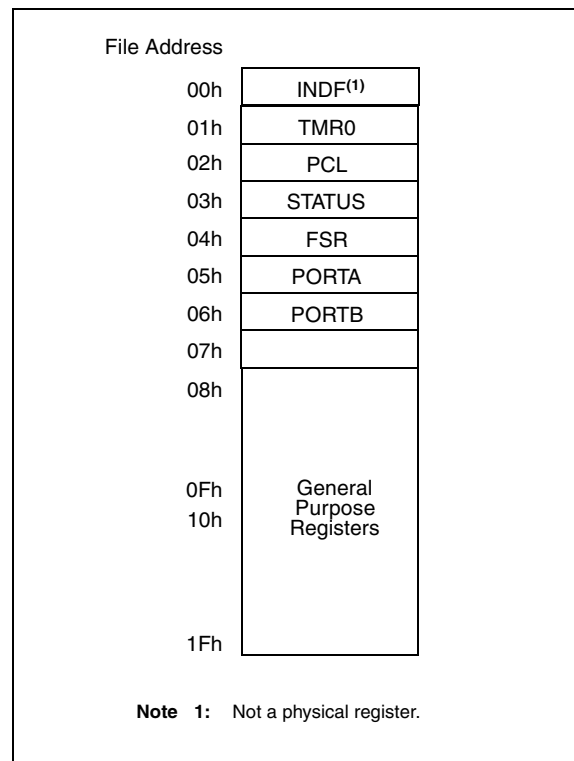
The general purpose registers are used for data and control information under command of the instructions.

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**



### 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.

# PIC16HV540

**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 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 <sup>(1)</sup>	PCL	Low order 8 bits of PC								1111 1111	1111 1111	1111 1111	1111 1111
03h	STATUS	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	
PCWUF	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	
bit7								bit0
<p>bit 7: <b>PCWUF</b>: Pin Change Reset bit            1 = After Power-up Reset (POR) or SLEEP command            0 = After a wake-up on pin change event</p> <p>bit 6-5: <b>Not Applicable</b></p> <p>bit 4: <b><math>\overline{TO}</math></b>: Time-out bit            1 = After power-up, <code>CLRWDT</code> instruction, or <code>SLEEP</code> instruction            0 = A WDT time-out occurred</p> <p>bit 3: <b><math>\overline{PD}</math></b>: Power-down bit            1 = After power-up or by the <code>CLRWDT</code> instruction            0 = By execution of the <code>SLEEP</code> instruction</p> <p>bit 2: <b>Z</b>: 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</p> <p>bit 1: <b>DC</b>: Digit carry/borrow bit (for <code>ADDWF</code> and <code>SUBWF</code> instructions)  <b>ADDWF</b>            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  <b>SUBWF</b>            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</p> <p>bit 0: <b>C</b>: Carry/borrow bit (for <code>ADDWF</code>, <code>SUBWF</code> and <code>RRF</code>, <code>RLF</code> instructions)  <b>ADDWF</b>            1 = A carry occurred            0 = A carry did not occur  <b>SUBWF</b>            1 = A borrow did not occur            0 = A borrow occurred</p> <p><b>RRF or RLF</b>            Load bit with LSB or MSb, respectively</p>								
<p>R = Readable bit            W = Writable bit            - n = Value at POR reset</p>								

# PIC16HV540

## 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
							0

bit7

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	W-1	W-1
—	—	PCWU	SWDTEN	RL	SL	BODL	BODEN		
bit7								0	

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

bit 7-6: **Unimplemented**

bit 5: **PCWU**: Wake-up on Pin Change  
 1 = Disabled  
 0 = Enabled

bit 4: **SWDTEN**: Software Controlled WDT Enable bit  
 1 = WDT is turned off if 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: **BODEN**: Brown-out Enabled  
 1 = Disabled  
 0 = Enabled

# PIC16HV540

## 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.

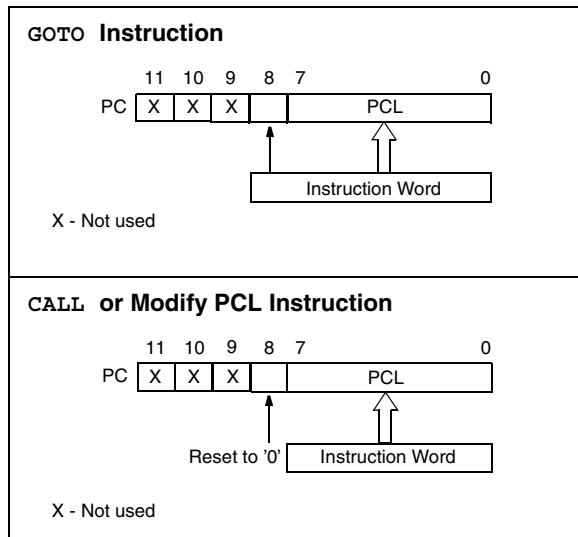
For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. (Figure 4-3).

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 pre-selected.

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

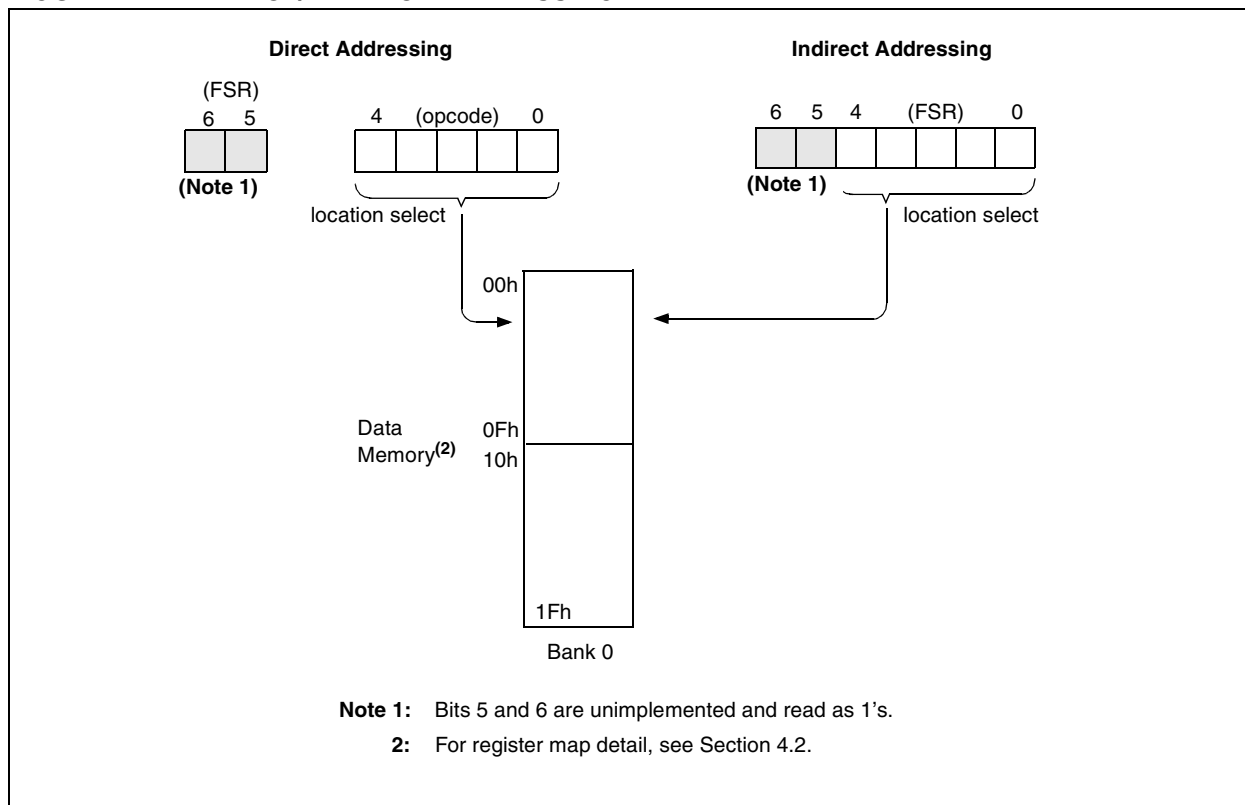
CONTINUE
       : ;YES, continue
    
```

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**



# PIC16HV540

---

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 `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. `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.

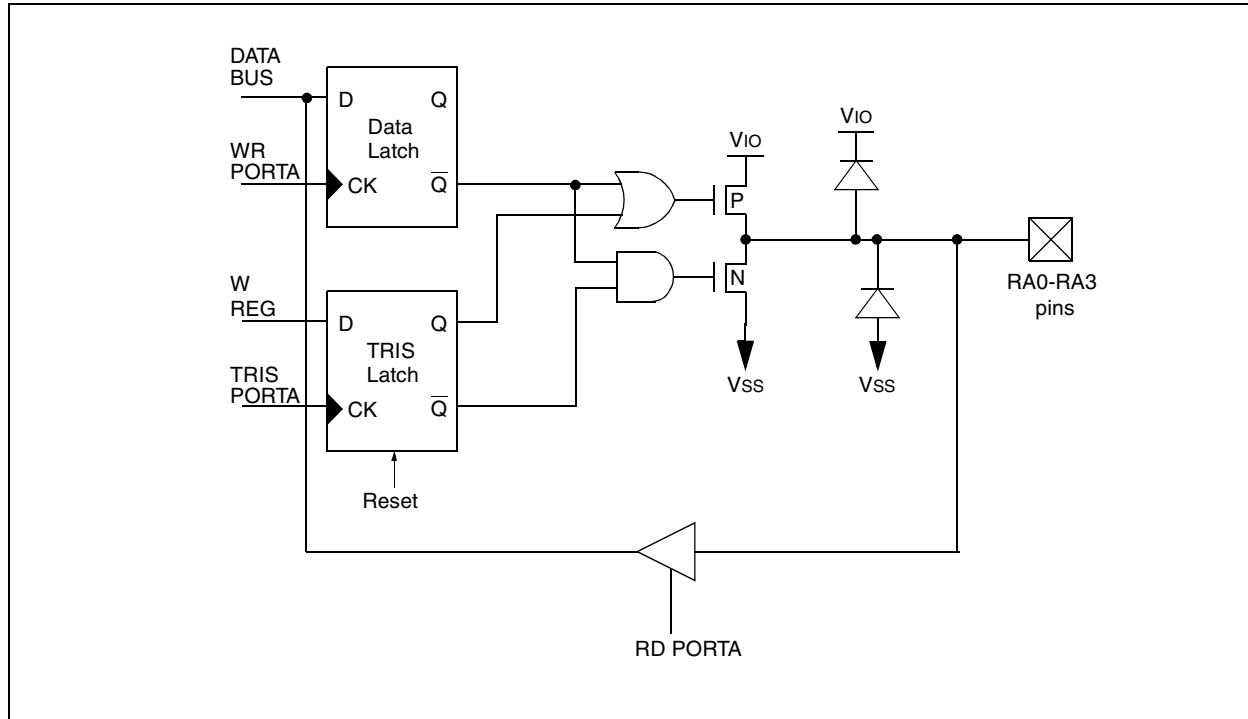
The TRIS registers are "write-only" and are set (output drivers disabled) upon RESET.

### 5.4 I/O Interfacing

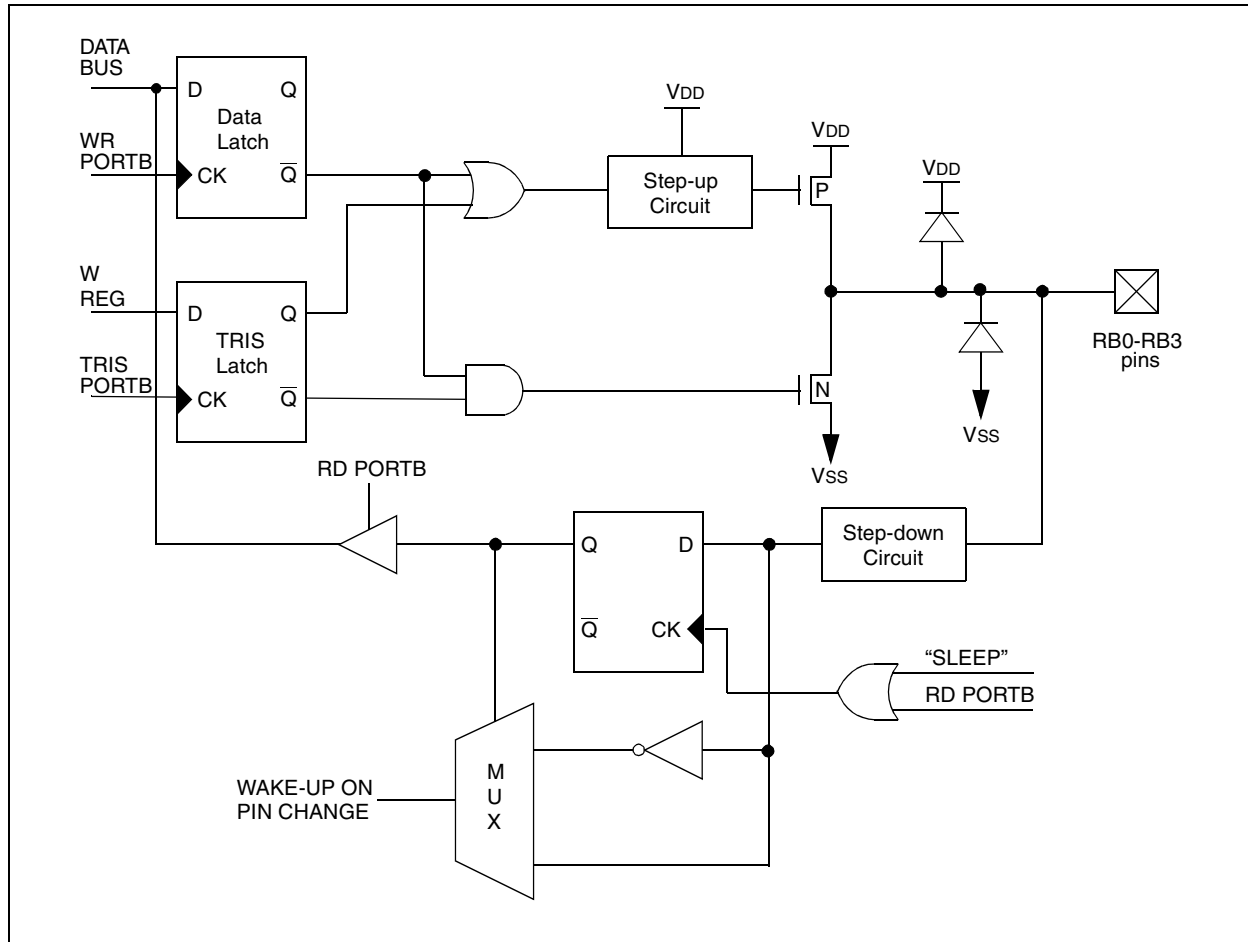
The equivalent circuit for the PORTA and PORTB I/O pins are shown in Figure 5-1 through Figure 5-4. All ports may be used for both input and output operation. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF PORTB, W`). The 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.

# PIC16HV540

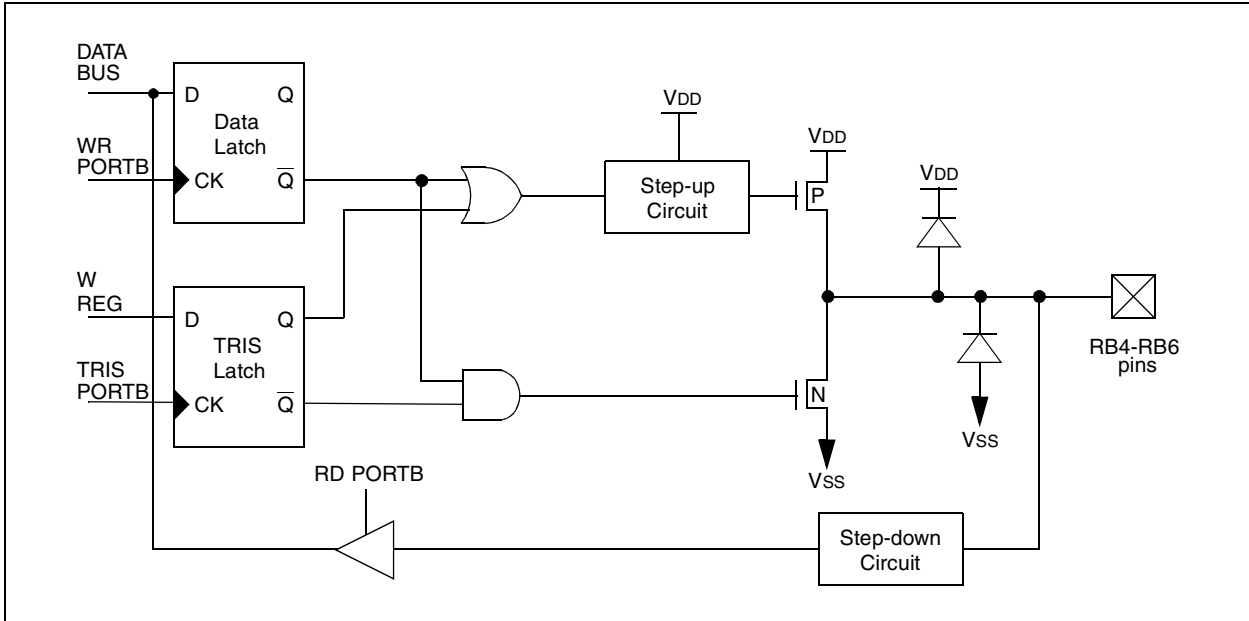
**FIGURE 5-1: BLOCK DIAGRAM OF PORTA<0:3> PINS**



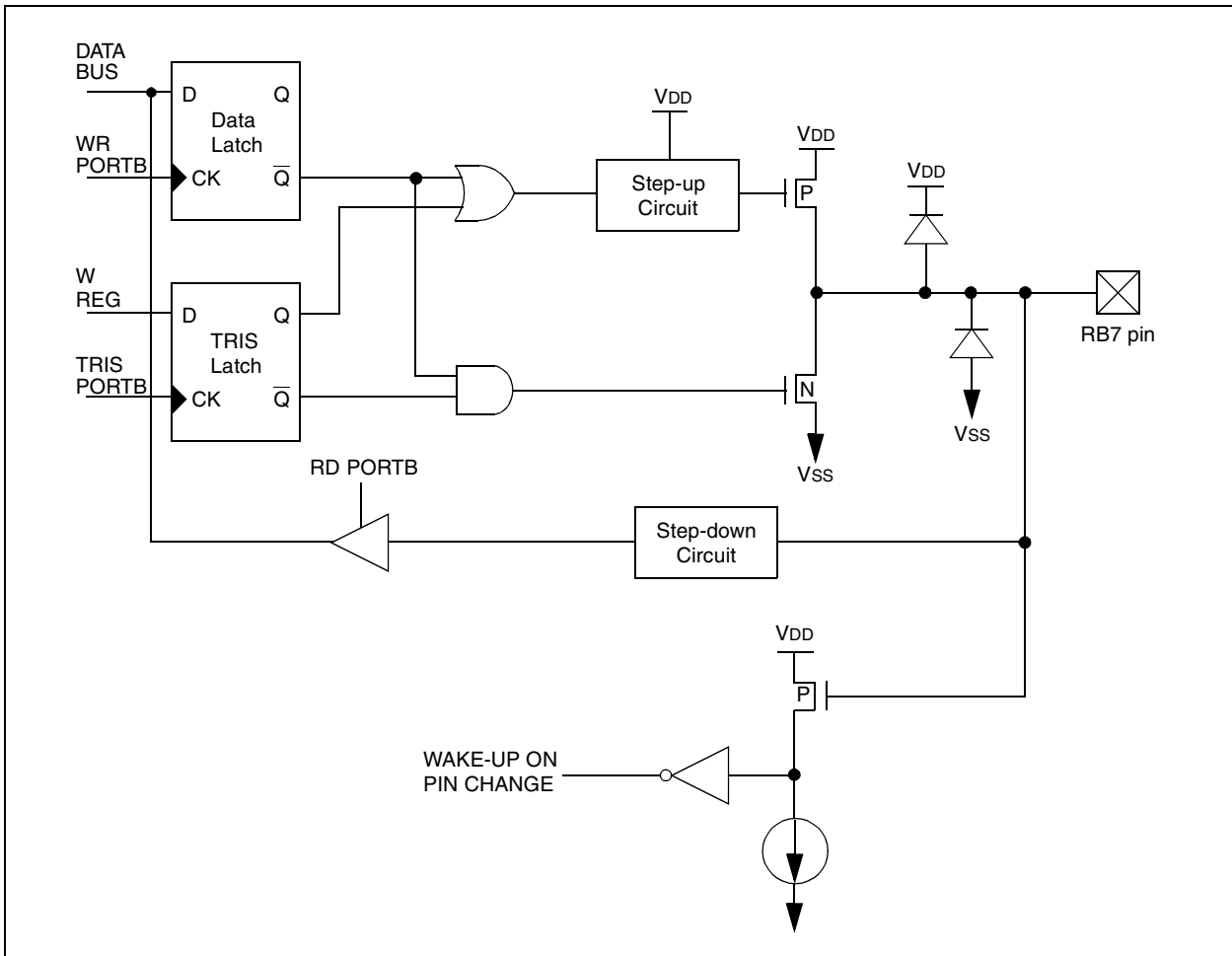
**FIGURE 5-2: BLOCK DIAGRAM OF PORTB<0:3> PINS**



**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 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	PCWUF	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	100x xxxx	100q quuu	000u uuuu	x00x xxxx
N/A	OPTION2	—	—	PCWU	SWDTEN	RL	SL	BODL	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

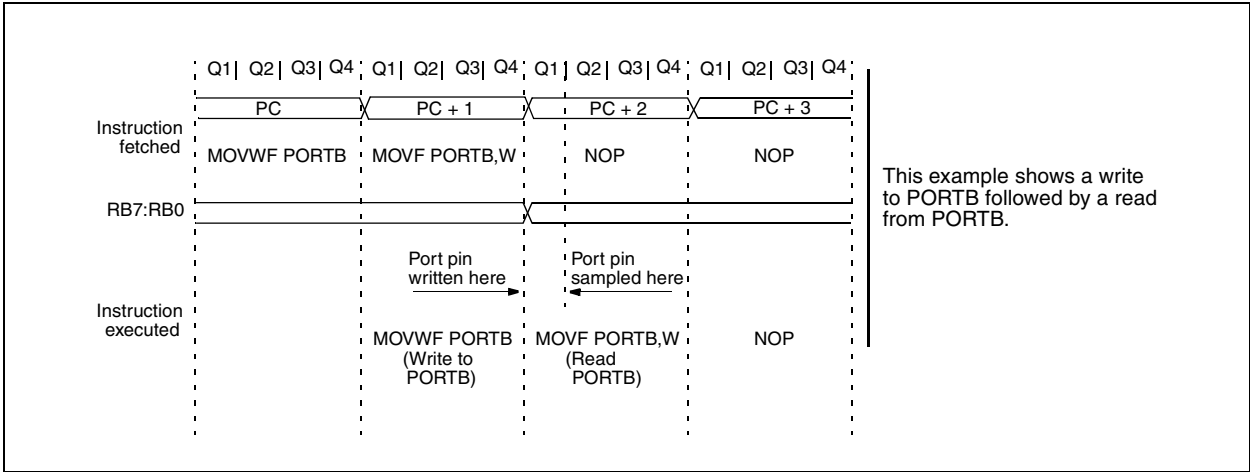
```

;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



# PIC16HV540

---

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.

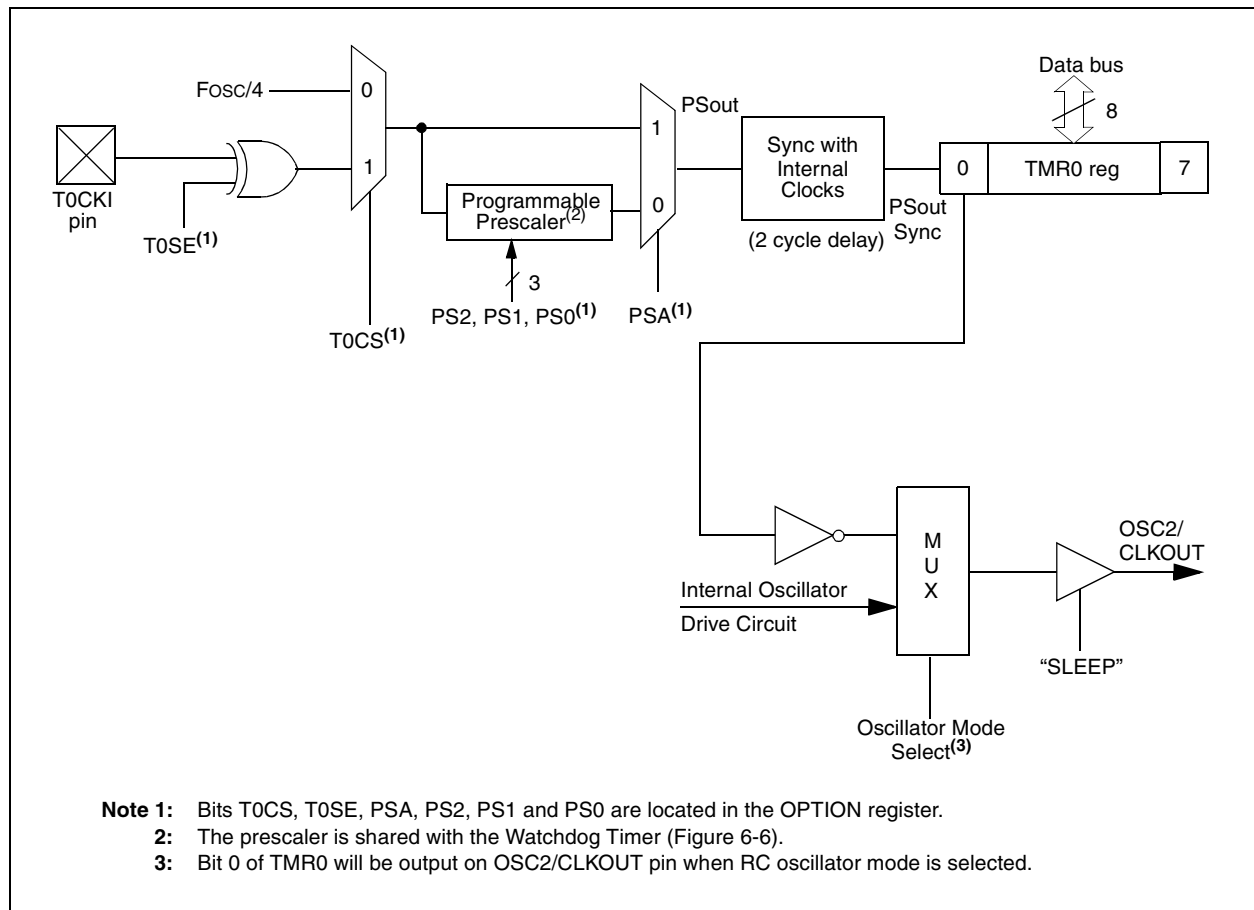
Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles (Figure 6-3 and Figure 6-4). The user can work around this by 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 TOCK1. 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**





## 6.1 Using Timer0 with an External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

### 6.1.1 EXTERNAL CLOCK SYNCHRONIZATION

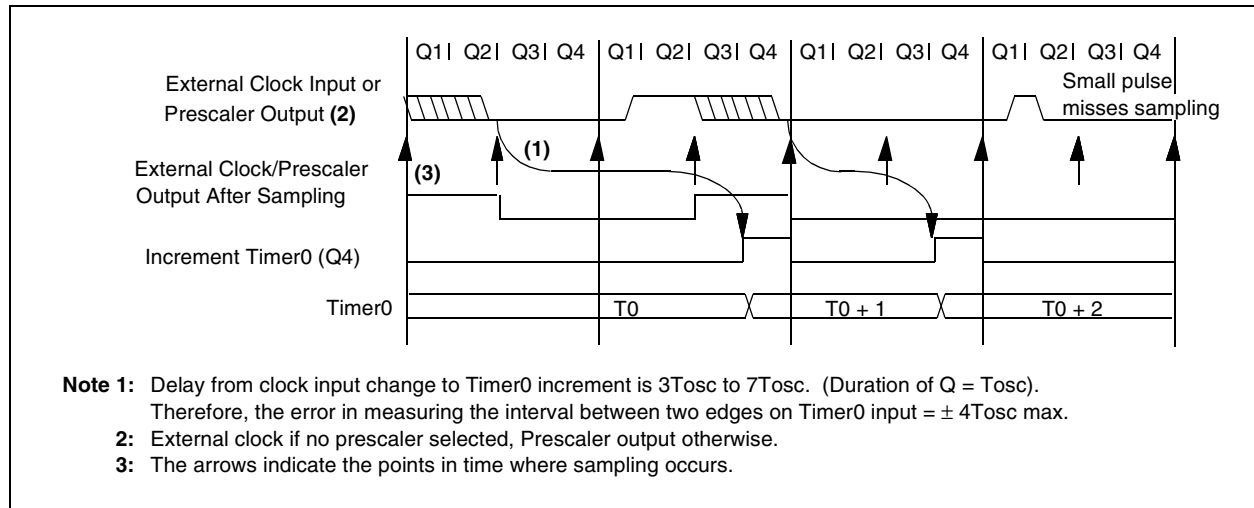
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4TOSC (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

### 6.1.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

**FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK**



## 6.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer (WDT) (WDT postscaler not implemented on PIC16C52), respectively (Section 6.1.2). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that the prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT, and vice-versa.

The PSA and PS2:PS0 bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRWF 1, MOVWF 1, BSF 1,x, etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a RESET, the prescaler contains all '0's.

### 6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on the fly” during program execution). To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

#### EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

```

1. CLRWDT           ;Clear WDT
2. CLRF    TMR0    ;Clear TMR0 & Prescaler
3. MOVLW   '00xx1111'b ;These 3 lines (5, 6, 7)
4. OPTION          ; are required only if
                   ; desired
5. CLRWDT           ;PS<2:0> are 000 or 001
6. MOVLW   '00xx1xxx'b ;Set Postscaler to
7. OPTION          ; desired WDT rate

```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 6-2. This sequence must be used even if the WDT is disabled. A CLRWDT instruction should be executed before switching the prescaler.

#### EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)

```

CLRWDT           ;Clear WDT and
                 ;prescaler
MOVLW   'xxxx0xxx' ;Select TMR0, new
                 ;prescale value and
                 ;clock source

OPTION

```

## 6.3 Programmable Clock Generator

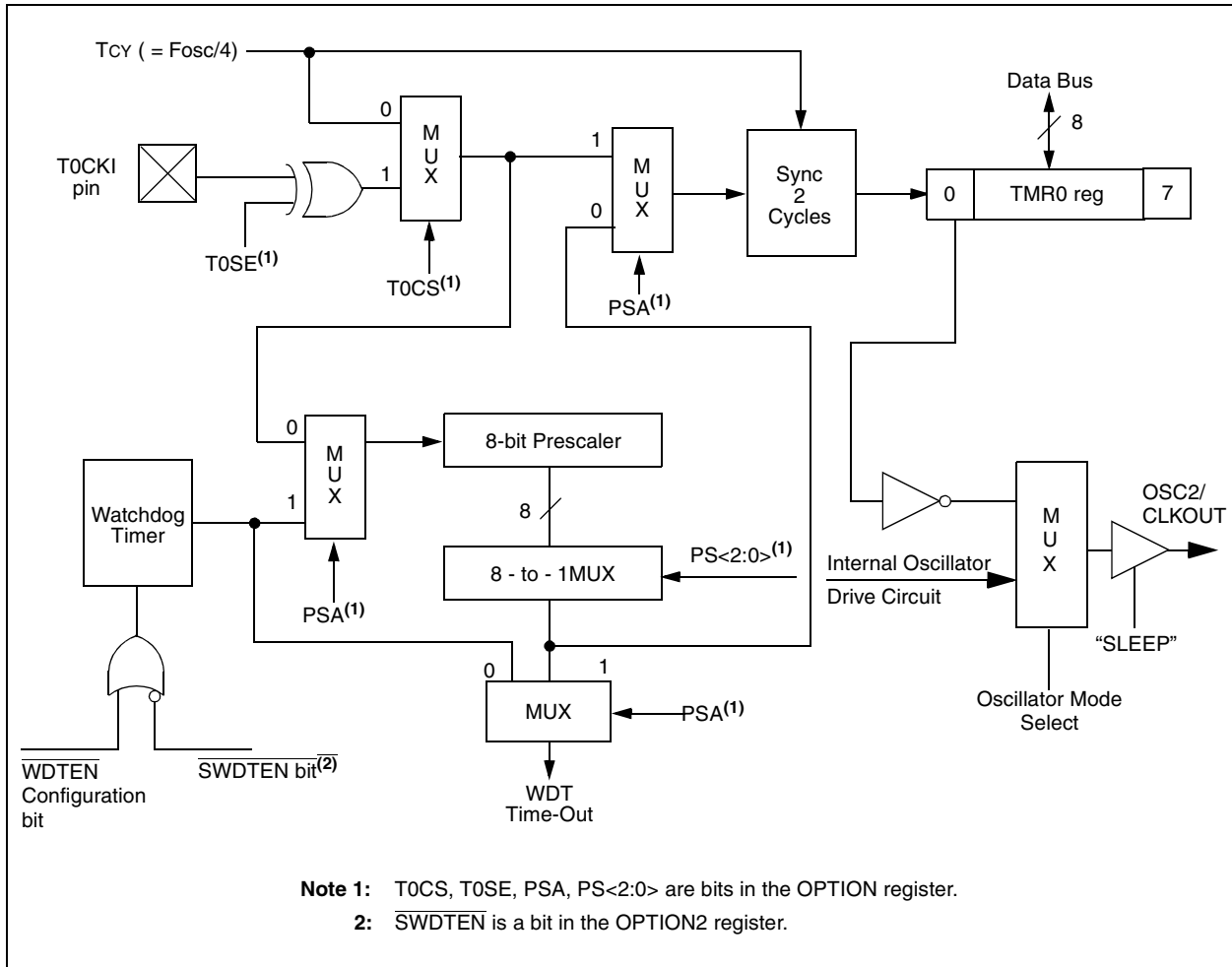
When the PIC16HV540 is programmed to operate in the RC oscillator mode, the CLKOUT pin is connected to the compliment state of TMR0<0>. Use of the prescaler rate select bits PSA:PS0 in the OPTION register will provide for frequencies of CLKIN/8 to CLKIN/1024 on the CLKOUT pin.

#### EXAMPLE 6-3:

Fosc	PRESCALER SETTING/CLKOUT FREQUENCY	
	CLKIN/1024	CLKIN/8
1Mhz	976 Hz	125 kHz
2Mhz	1953 Hz	250 kHz
3Mhz	2930 Hz	375 kHz
4Mhz	3906 Hz	500 kHz

In addition to this mode of operation, TMR0<0> can be toggled via the bcf and bsf bit type instructions. For this mode, the T0CS bit in the OPTION register must be set to 1. This setting configures TMR0 to increment on the T0CKI pin. To set the CLKOUT pin high, a bcf TMR0,0 instruction is used and to set the CLKOUT pin low, the bsf TMR0,0 instruction is used. The T0CKI pin should be pulled high or low to prevent false state changes on the CLKOUT pin.

**FIGURE 6-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



# PIC16HV540

---

NOTES:

## 7.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits that deal with the needs of real-time applications. The PIC16HV540 family of microcontrollers has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features are:

- Oscillator selection
- Reset
- Power-On Reset (POR)
- Brown-out Detect
- Device Reset Timer (DRT)
- Wake-up from SLEEP on Pin Change
- Enhanced Watchdog Timer (WDT)
- SLEEP
- Code protection

The PIC16HV540 Family has a Watchdog Timer which can be shut off only through configuration bit WDTEN. It runs off of its own RC oscillator for added reliability. There is an 18 ms delay provided by the Device Reset Timer (DRT), intended to keep the chip in reset until the crystal oscillator is stable. With this timer on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake up from SLEEP through external reset or through a Watchdog Timer time-out. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

### 7.1 Configuration Bits

Configuration bits can be programmed to select various device configurations. Two bits are for the selection of the oscillator type and one bit is the Watchdog Timer enable bit. Nine bits are code protection bits (Figure 7-1) for the PIC16HV540 devices.

### REGISTER 7-1: CONFIGURATION WORD FOR PIC16HV540

	CP	CP	CP	CP	CP	CP	CP	CP	CP	WDTEN	Fosc1	Fosc0	
											bit0	Register: CONFIG Address <sup>(1)</sup> : 0FFFh	
bit11													
bit 11-3:	<b>CP:</b> Code Protection bits												
	1 = Code protection off												
	0 = Code protection on												
bit 2:	<b>WDTEN:</b> Watchdog Timer Enable bit												
	1 = WDT enabled												
	0 = WDT disabled (control is placed on the <u>SWDTEN</u> bit)												
bit 1-0:	<b>Fosc&lt;1:0&gt;:</b> Oscillator Selection bits												
	11 = RC oscillator												
	10 = HS oscillator												
	01 = XT oscillator												
	00 = LP oscillator												
<b>Note 1:</b> Refer to the PIC16C5X Programming Specification (Literature number DS30190) to determine how to access the configuration word.													

# PIC16HV540

## 7.2 Oscillator Configurations

### 7.2.1 OSCILLATOR TYPES

The PIC16HV540 can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

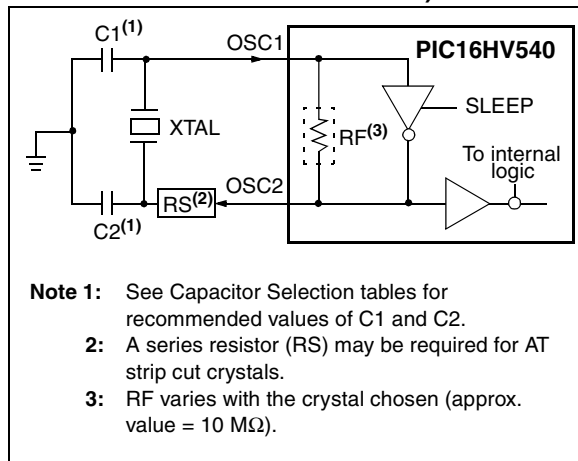
- LP: Low Power Crystal
- XT: Crystal/Resonator
- HS: High Speed Crystal/Resonator
- RC: Resistor/Capacitor

**Note:** Not all oscillator selections available for all parts. See Section 7.1.

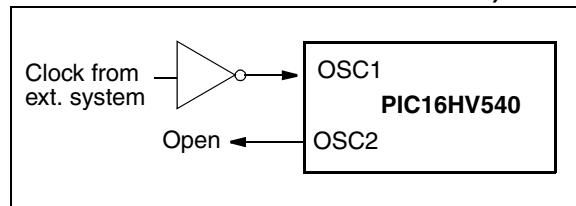
### 7.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 7-1). The PIC16HV540 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source drive the OSC1/CLKIN pin (Figure 7-2).

**FIGURE 7-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)**



**FIGURE 7-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**TABLE 7-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS - PIC16HV540**

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
XT	455 kHz	68-100 pF	68-100 pF
	2.0 MHz	15-33 pF	15-33 pF
	4.0 MHz	10-22 pF	10-22 pF
HS	8.0 MHz	10-22 pF	10-22 pF
	16.0 MHz	10 pF	10 pF

**Note:** These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

**TABLE 7-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR - PIC16HV540**

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz <sup>(1)</sup>	15 pF	15 pF
XT	100 kHz	15-30 pF	200-300 pF
	200 kHz	15-30 pF	100-200 pF
	455 kHz	15-30 pF	15-100 pF
	1 MHz	15-30 pF	15-30 pF
	2 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15 pF	15 pF
	20 MHz	15 pF	15 pF

**Note 1:** For  $V_{DD} > 4.5V$ ,  $C1 = C2 \approx 30$  pF is recommended.

**2:** These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

**Note:** If you change from this device to another device, please verify oscillator characteristics in your application.

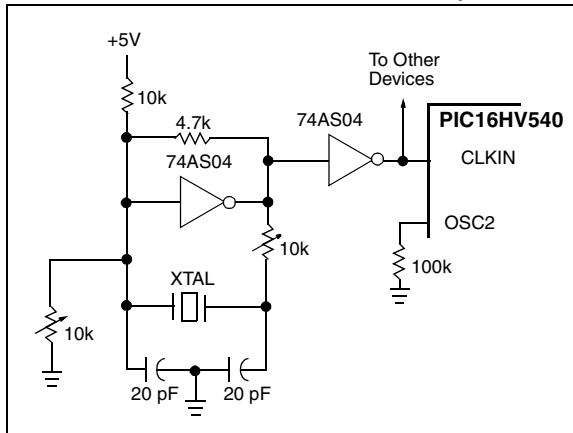


## 7.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator or a simple oscillator circuit with TTL gates can be used as an external crystal oscillator circuit. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with parallel resonance, or one with series resonance.

Figure 7-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 k $\Omega$  resistor provides the negative feedback for stability. The 10 k $\Omega$  potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

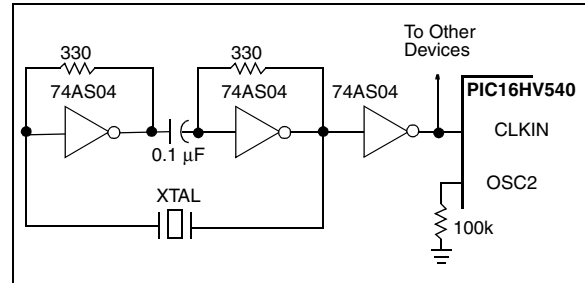
**FIGURE 7-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT (USING XT, HS OR LP OSCILLATOR MODE)**



**Note:** If you change from this device to another device, please verify oscillator characteristics in your application.

Figure 7-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330  $\Omega$  resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 7-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT (USING XT, HS OR LP OSCILLATOR MODE)**



**Note:** If you change from this device to another device, please verify oscillator characteristics in your application.

## 7.2.4 RC OSCILLATOR

For timing insensitive applications, the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{ext}$ ) and capacitor ( $C_{ext}$ ) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low  $C_{ext}$  values. The user also needs to take into account variation due to tolerance of external R and C components used.

Figure 7-5 shows how the R/C combination is connected to the PIC16HV540. For  $R_{ext}$  values below 2.2 k $\Omega$ , the oscillator operation may become unstable, or stop completely. For very high  $R_{ext}$  values (e.g., 1 M $\Omega$ ) the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping  $R_{ext}$  between 3 k $\Omega$  and 100 k $\Omega$ .

Although the oscillator will operate with no external capacitor ( $C_{ext} = 0$  pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

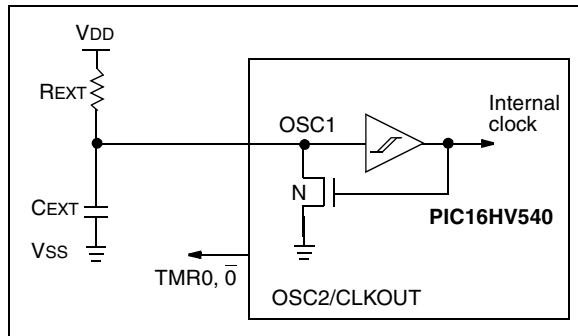
# PIC16HV540

The Electrical Specifications sections show RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

Also, see the Electrical Specifications sections for variation of oscillator frequency due to  $V_{DD}$  for given  $R_{EXT}/C_{EXT}$  values as well as frequency variation due to operating temperature for given R, C, and  $V_{DD}$  values.

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 rate select bits, clock out frequencies of  $CLKIN/8$  to  $CLKIN/1024$  can be generated.

**FIGURE 7-5: RC OSCILLATOR MODE**



**Note:** If you change from this device to another device, please verify oscillator characteristics in your application.

## 7.3 Reset

PIC16HV540 devices may be reset in one of the following ways:

- Power-On Reset (POR)
- $\overline{MCLR}$  reset (normal operation)
- $\overline{MCLR}$  wake-up reset (from SLEEP)
- WDT reset (normal operation)
- WDT wake-up reset (from SLEEP)
- Wake-up from SLEEP on Pin Change
- Brown-out Detect

Table 7-3 shows these reset conditions for the PCL and STATUS registers.

Some registers are not affected in any reset condition. Their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-On Reset (POR),  $\overline{MCLR}$  or WDT Reset. A  $\overline{MCLR}$ , WDT Wake-up from SLEEP or Wake-up from SLEEP on Pin Change also results in a device RESET, and not a continuation of operation before SLEEP.

The  $\overline{TO}$  and  $\overline{PD}$  bits (STATUS <4:3>) and  $\overline{PCWUF}$  (STATUS<7>) are set or cleared depending on the different reset conditions (Section 7.9). These bits may be used to determine the nature of the reset.

Table 7-4 lists a full description of reset states of all registers. Figure 7-6 shows a simplified block diagram of the on-chip reset circuit.

**TABLE 7-3: RESET CONDITIONS FOR SPECIAL REGISTERS**

Condition	PCL Addr: 02h	STATUS Addr: 03h
Power-on Reset	1111 1111	1001 1xxx
MCLR Reset (normal operation)	1111 1111	u00u uuuu <sup>(1)</sup>
MCLR Wake-up (from SLEEP)	1111 1111	1001 0uuu
WDT Reset (normal operation)	1111 1111	u000 1uuu <sup>(2)</sup>
WDT Wake-up (from SLEEP)	1111 1111	1000 0uuu
Wake-up from SLEEP on Pin Change	1111 1111	000u uuuu
Brown-out Reset	1111 1111	x00x xxxx

Legend: u = unchanged, x = unknown, - = unimplemented read as '0'.

**Note 1:**  $\overline{TO}$  and  $\overline{PD}$  bits retain their last value until one of the other reset conditions occur.

**2:** The CLRWDT instruction will set the  $\overline{TO}$  and  $\overline{PD}$  bits.

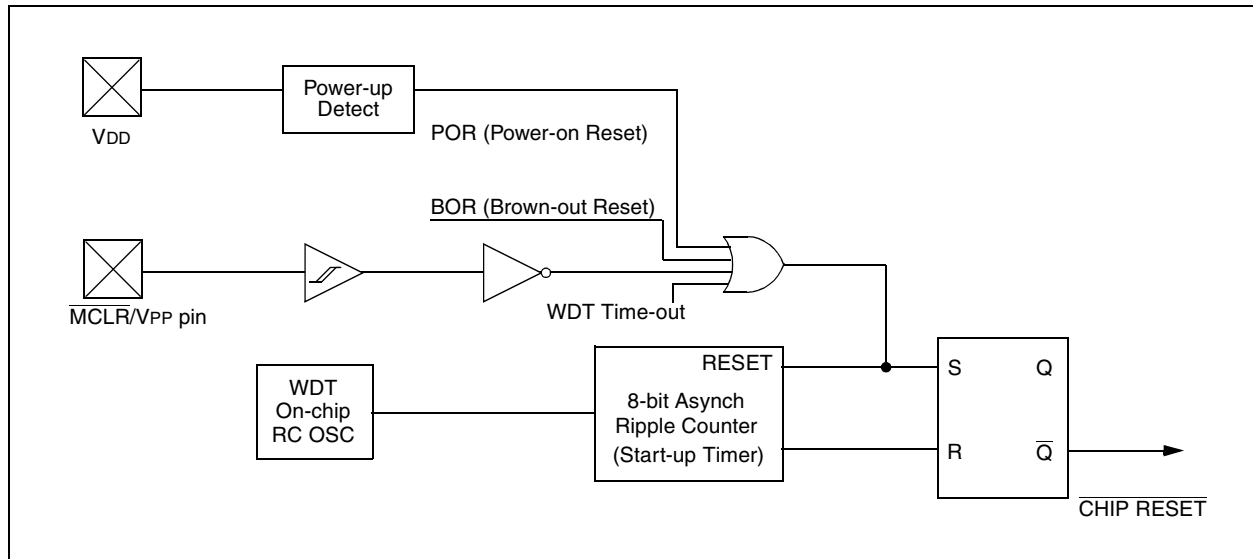
**TABLE 7-4: RESET CONDITIONS FOR ALL REGISTERS**

Register	Address	Power-On Reset	MCLR or WDT Reset	Wake-up on Pin Change	Brown-out Reset
W	N/A	xxxx xxxx	uuuu uuuu	uuuu uuuu	xxxx xxxx
TRIS	N/A	1111 1111	1111 1111	1111 1111	1111 1111
OPTION	N/A	--11 1111	--11 1111	--11 1111	--11 1111
OPTION2	N/A	--11 1111	--uu uuuu	--uu uuuu	--xx xxxx
INDF	00h	xxxx xxxx	uuuu uuuu	uuuu uuuu	xxxx xxxx
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu	xxxx xxxx
PCL <sup>(1)</sup>	02h	1111 1111	1111 1111	1111 1111	1111 1111
STATUS <sup>(1)</sup>	03h	1001 1xxx	100? ?uuu	000u uuuu	x00x xxxx
FSR	04h	111x xxxx	111u uuuu	111u uuuu	111x xxxx
PORTA	05h	---- xxxx	---- uuuu	---- uuuu	---- xxxx
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu	xxxx xxxx
General Purpose Register Files	07-1Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu	xxxx xxxx

Legend: u = unchanged, x = unknown, - = unimplemented, read as '0', q = see tables in Section 7.10 for possible values.  
? = value depends on condition.

**Note 1:** See Table 7-3 for reset value for specific conditions.

**FIGURE 7-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



# PIC16HV540

## 7.4 Power-On Reset (POR)

The PIC16HV540 incorporates on-chip Power-on Reset (POR) circuitry which provides an internal chip reset for most power-up situations. To use this feature, the user merely ties the  $\overline{\text{MCLR}}/\text{VPP}$  pin to  $\text{VDD}$ . A simplified block diagram of the on-chip Power-on Reset circuit is shown in Figure 7-7.

The Power-on Reset circuit and the Device Reset Timer (Section 7.5) circuit are closely related. On power-up, the reset latch is set and the DRT is reset. The DRT timer begins counting once it detects  $\overline{\text{MCLR}}$  to be high. After the time-out period, which is typically 18 ms, it will reset the reset latch and thus end the on-chip reset signal.

A power-up example where  $\overline{\text{MCLR}}$  is not tied to  $\text{VDD}$  is shown in Figure 7-8.  $\text{VDD}$  is allowed to rise and stabilize before bringing  $\overline{\text{MCLR}}$  high. The chip will actually come out of reset  $\text{T}_{\text{DRT}}$  msec after  $\overline{\text{MCLR}}$  goes high.

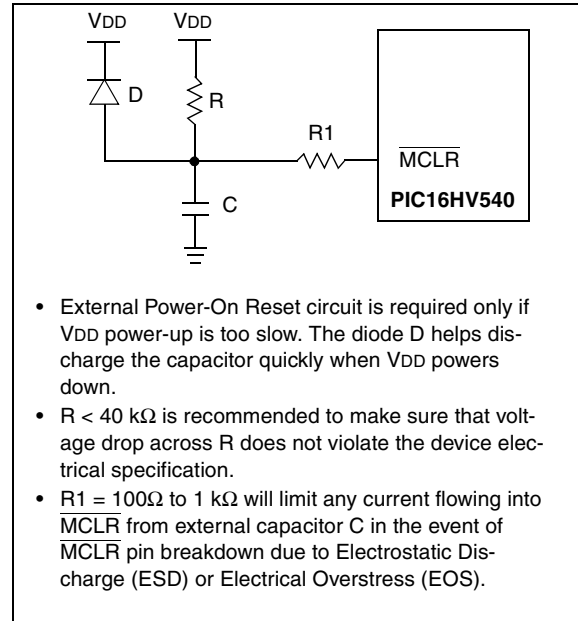
In Figure 7-9, the on-chip Power-on Reset feature is being used ( $\overline{\text{MCLR}}$  and  $\text{VDD}$  are tied together). The  $\text{VDD}$  is stable before the start-up timer times out and there is no problem in getting a proper reset. However, Figure 7-10 depicts a problem situation where  $\text{VDD}$  rises too slowly. The time between when the DRT senses a high on the  $\overline{\text{MCLR}}/\text{VPP}$  pin, and when the  $\overline{\text{MCLR}}/\text{VPP}$  pin (and  $\text{VDD}$ ) actually reach their full value, is too long. In this situation, when the start-up timer times out,  $\text{VDD}$  has not reached the  $\text{VDD}(\text{min})$  value and the chip is, therefore, not guaranteed to function correctly. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times (Figure 7-7).

**Note:** When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met.

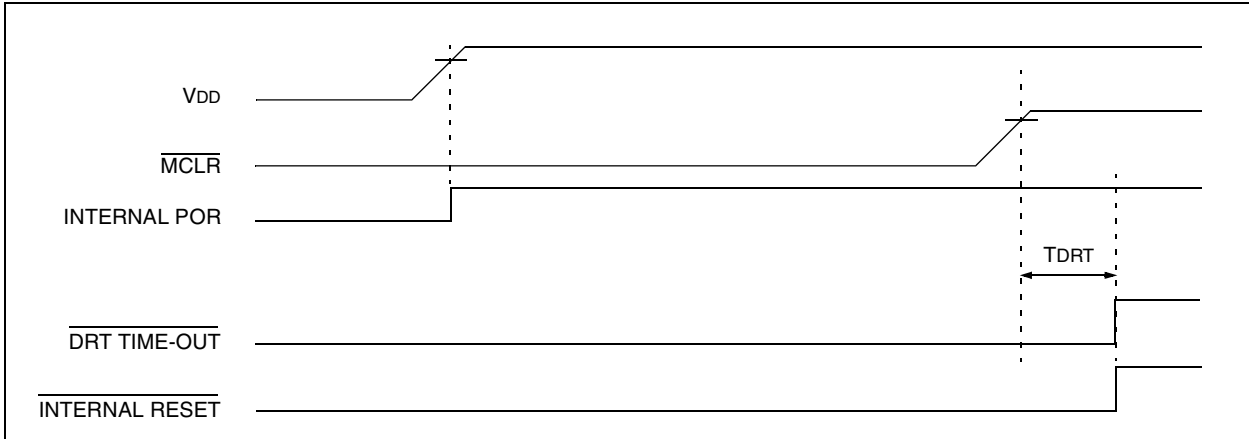
For more information on PIC16HV540 POR, see *Power-Up Considerations - AN522* in the [Embedded Control Handbook](#).

The POR circuit does not produce an internal reset when  $\text{VDD}$  declines.

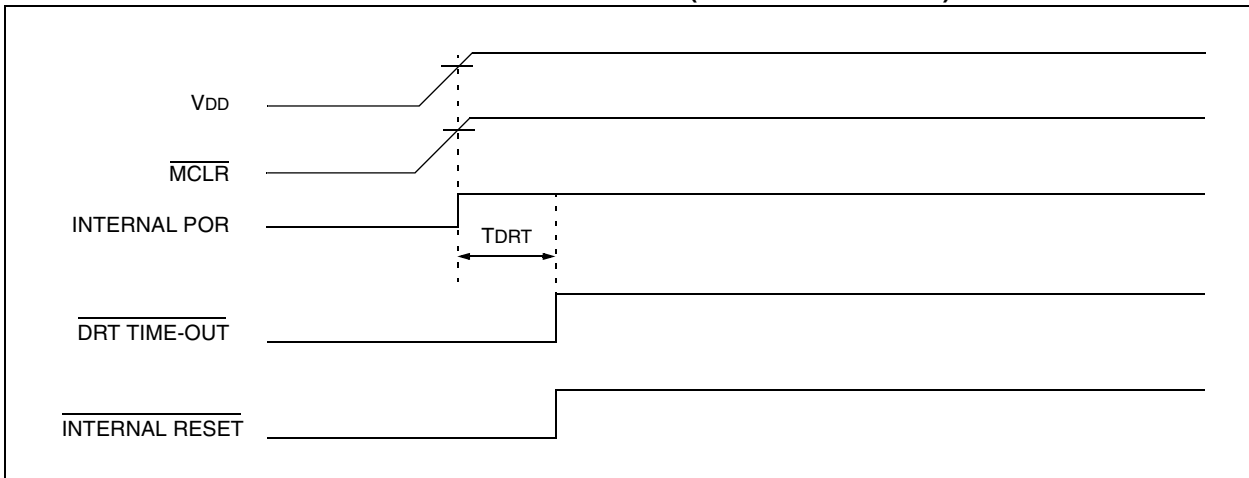
**FIGURE 7-7: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW  $\text{VDD}$  POWER-UP)**



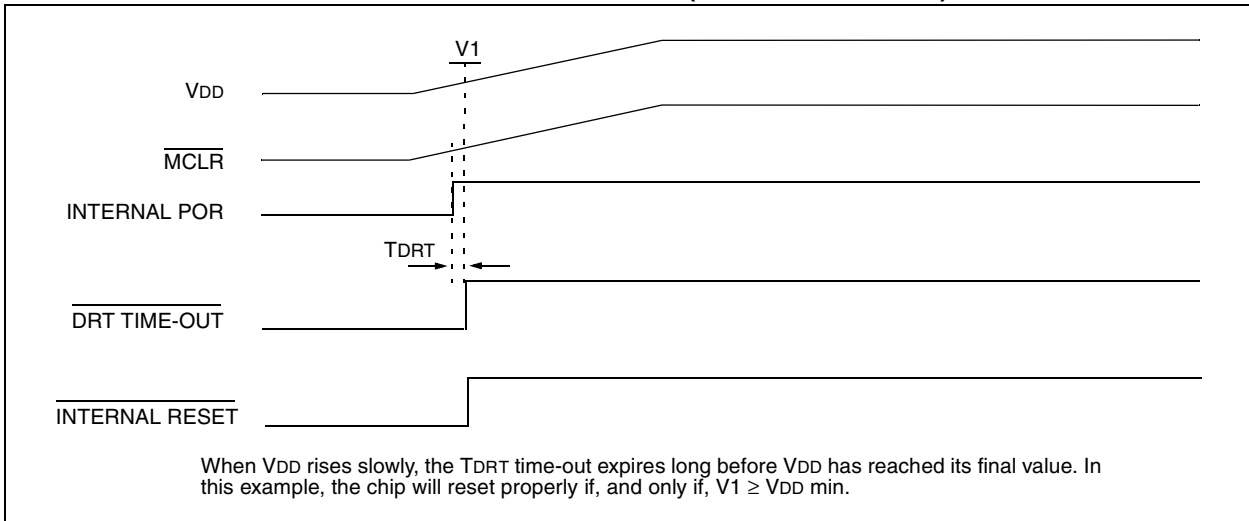
**FIGURE 7-8: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ )**



**FIGURE 7-9: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ ): FAST  $V_{\text{DD}}$  RISE TIME**



**FIGURE 7-10: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ ): SLOW  $V_{\text{DD}}$  RISE TIME**



## 7.5 Device Reset Timer (DRT)

In the PIC16HV540, the Device Reset Timer (DRT) runs any time the device is powered up. DRT runs from reset and varies based on oscillator selection (see Table 7-5).

The DRT provides a fixed 18 ms nominal time-out on reset. The DRT operates on an internal RC oscillator. The processor is kept in RESET as long as the DRT is active. The DRT delay allows V<sub>DD</sub> to rise above V<sub>DD</sub> min., and for the oscillator to stabilize.

Oscillator circuits based on crystals or ceramic resonators require a certain time after power-up to establish a stable oscillation. The on-chip DRT keeps the device in a RESET condition for approximately 18 ms after the voltage on the  $\overline{\text{MCLR}}/\text{VPP}$  pin has reached a logic high (V<sub>IH</sub>) level. Thus, external RC networks connected to the  $\overline{\text{MCLR}}$  input are not required in most cases, allowing for savings in cost-sensitive and/or space restricted applications.

The Device Reset time delay will vary from chip to chip due to V<sub>DD</sub>, temperature, and process variation. See AC parameters for details.

The DRT will also be triggered upon a Watchdog Timer time-out,  $\overline{\text{MCLR}}$  Reset, Wake-up from SLEEP on Pin Change and Brown-out Reset. When the external RC oscillator mode is selected, all DRT periods, after the initial POR, are 1 ms (typical).

**TABLE 7-5: DRT (DEVICE RESET TIMER PERIOD)**

Oscillator Configuration	POR Reset	Subsequent Resets
EXTRC	18 ms (typical)	1 ms (typical)
LP, XT & HS	18 ms (typical)	18 ms (typical)

## 7.6 Brown-Out Detect (BOD)

The PIC16HV540 has on-chip Brown-out Detect circuitry. If enabled and if the internal power, V<sub>REG</sub>, falls below parameter B<sub>VDD</sub> (See Section 10.1), for greater time than parameter T<sub>BOD</sub> (See Table 10-3) the brown-out condition will reset the chip. A reset is not guaranteed if V<sub>REG</sub> falls below B<sub>VDD</sub> for less time than parameter (T<sub>BOD</sub>).

On resets (Brown-out, Watchdog,  $\overline{\text{MCLR}}$  and Wake-up on Pin Change), the chip will remain in reset until V<sub>REG</sub> rises above B<sub>VDD</sub>. Once the B<sub>VDD</sub> threshold has been met the DRT will now be invoked and will keep the chip in reset an additional 18mS (LP, XT and HS oscillator modes) or 1mS for EXTRC.

If V<sub>REG</sub> drops below B<sub>VDD</sub> while the DRT is running, the chip will go back into a Brown-out Reset and the DRT will be re-initialized. Once V<sub>REG</sub> rises above the B<sub>VDD</sub>, the DRT will execute the specified time period. Figure 7-11 shows typical Brown-out situations.

The Brown-out Detect circuit can be disabled or enabled by setting the BODEN bit in the OPTION2 SFR. The Brown-out Detect is disabled upon all Power-on Resets (POR).

### 7.6.1 IMPLEMENTING THE ON-CHIP BOD CIRCUIT

The PIC16HV540 BOD circuitry differs from “conventional” brown-out detect circuitry in that the BOD circuitry on the PIC16HV540 does not directly detect “dips” in the external V<sub>DD</sub> supply voltage but rather the internal V<sub>REG</sub>. The functionality of the BOD circuitry ensures that program execution will halt and a reset state will be entered into prior to the internal logic becoming corrupted. The BOD circuit has two selectable voltage settings, nominally 5V and 3V. Each regulation voltage setting with its associated minimum and maximum B<sub>VDD</sub> parameters has an intended operational mode that must be carefully considered.

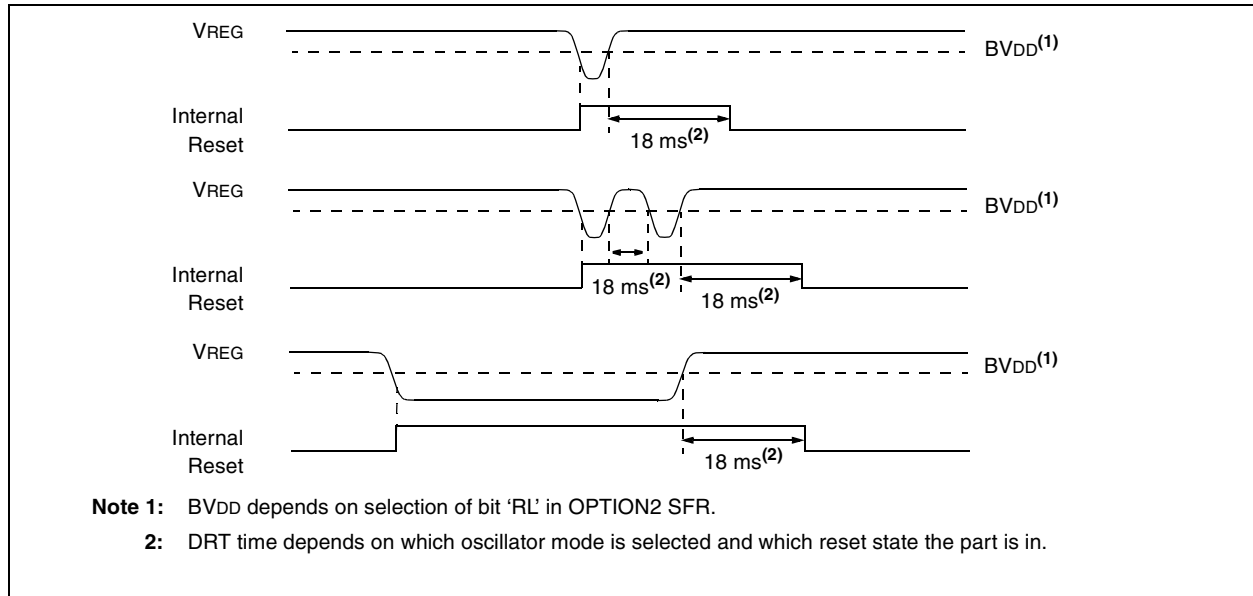
For the 5V V<sub>REG</sub> setting, the minimum B<sub>VDD</sub> parameter is 2.7V. This minimum B<sub>VDD</sub> voltage is below the part V<sub>DD</sub> minimum requirements. This operational setting is primarily intended for use when the PIC16HV540 is operating at 4Mhz and V<sub>DD</sub> > 5.5V.

For the 3V V<sub>REG</sub> setting, the minimum B<sub>VDD</sub> parameter is 1.8V. This minimum B<sub>VDD</sub> voltage is below the part V<sub>DD</sub> minimum requirements. This operational setting is primarily intended for use when the PIC16HV540 is in SLEEP. RAM retention is protected by the 1.8V trip level.

For the regulation and Brown-out circuits to function as intended the applied V<sub>DD</sub> is nominally 0.5V greater than the regulation voltage setting.

Finally, if the internal brown-out circuit is deemed not to meet system design requirements then an external brown-out protection circuit may be required. Microchip offers a complete family of voltage supervisor products which can meet most design requirements.

**FIGURE 7-11: BROWN-OUT SITUATIONS**



## 7.7 Watchdog Timer (WDT)

The Watchdog Timer (WDT) is a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins have been stopped, for example, by execution of a SLEEP instruction. During normal operation or SLEEP, a WDT Reset or Wake-up Reset generates a device RESET.

The  $\overline{\text{TO}}$  bit (STATUS<4>) will be cleared upon a Watchdog Timer Reset.

The Watchdog Timer is enabled/disabled by a device configuration bit (see Figure 7-1). If the WDT is enabled, software execution may not disable this function. When the WDTEN configuration bit is cleared, the SWDTEN bit, OPTION2<4>, enables/disables the operation of the WDT.

### 7.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). If a longer time-out period is desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, time-out a period of a nominal 2.3 seconds can be realized. These periods vary with temperature, VDD and part-to-part process variations (see DC specs).

Under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

### 7.7.2 WDT PROGRAMMING CONSIDERATIONS

The CLRWDT instruction clears the WDT and the postscaler, if assigned to the WDT, and prevents it from timing out and generating a device RESET.

The SLEEP instruction resets the WDT and the postscaler, if assigned to the WDT. This gives the maximum SLEEP time before a WDT Wake-up Reset.

## 7.8 Internal Voltage Regulators

The PIC16HV540 has 2 internal voltage regulators. The PORTA I/O pads and OSC2 are powered by one internal voltage regulator V<sub>IO</sub>, while the second internal voltage regulator V<sub>REG</sub>, powers the PICmicro® device core. Both regulated voltage levels can be synchronously switched in the active modes between 3V and 5V through bit "RL" in the OPTION2 register. In addition, the "SL" bit in the OPTION2 register can be used to control the core's regulated voltage level during SLEEP mode. V<sub>REG</sub> regulates the 15V power applied to the V<sub>DD</sub> pin.

The on-chip Brown-out Detect circuitry monitors the CPU regulated voltage V<sub>REG</sub>, for determining if a brown-out reset is generated (see Section 7.6 for more details on the BOD).

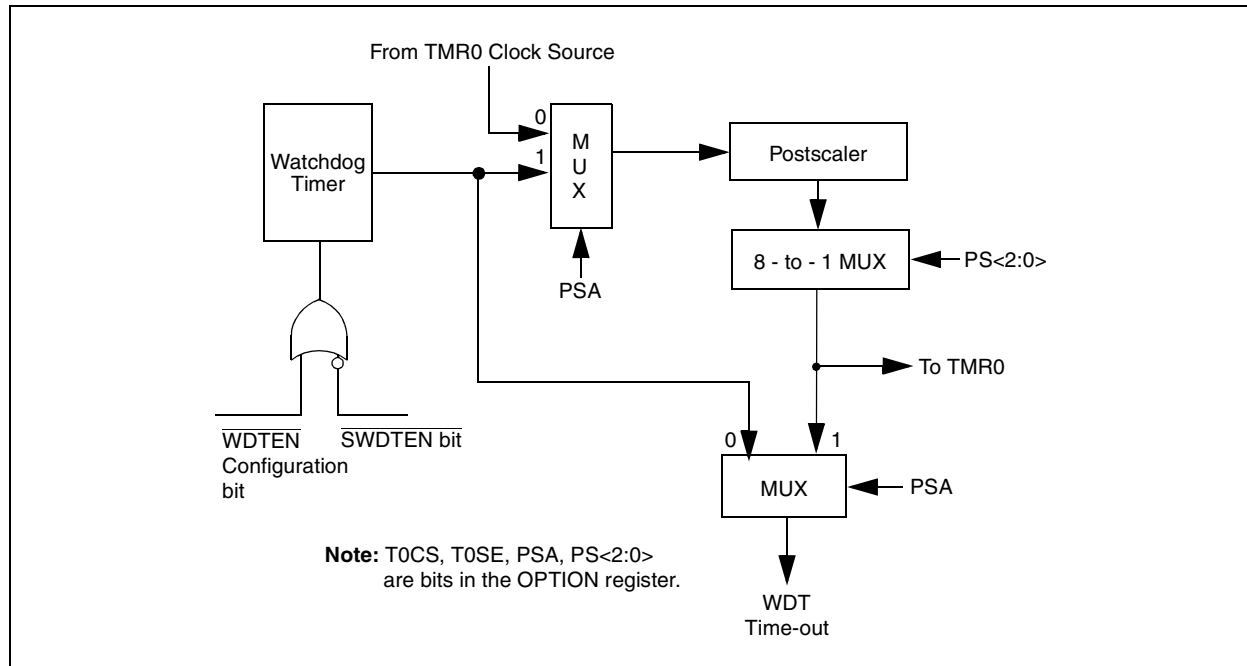
The regulator circuits are identical in functional nature but only the V<sub>IO</sub> regulator voltage can be measured, externally (See Section 10.1 for V<sub>IO</sub> parameters). The operational voltage range and pin loading requirements must be considered to ensure proper system operation. For example, if 3V regulation is implemented during the SLEEP mode and 40mA is being sourced from PORTA, the V<sub>IO</sub> regulation voltage may approach the specified minimum voltage. This may be an issue to consider for connections to external circuitry. Likewise, if zero current is sourced from the PORTA pins, the regulation

# PIC16HV540

voltage may approach the maximum value. Again this condition should be considered when interfacing to external circuitry.

In addition, the voltage level applied to the external  $V_{DD}$  pin and operational temperature affects the internal regulation voltage.

**FIGURE 7-12: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 7-6: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER**

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 MCLR and WDT Reset	Value on Wake-up on Pin Change	Value on Brown-out Reset
N/A	OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0	--11 1111	--11 1111	--11 1111	--11 1111
N/A	OPTION2	—	—	PCWU	SWDTEN	RL	SL	BODL	BODEN	--00 000	--uu uuuu	--uu uuuu	--xx xxxx

Legend: Shaded boxes = Not used by Watchdog Timer, — = unimplemented, read as '0', u = unchanged, x = unknown.



## 7.9 Time-out Sequence and Power-down Status Bits ( $\overline{\text{TO}}$ / $\overline{\text{PD}}$ / $\overline{\text{PCWUF}}$ )

The  $\overline{\text{TO}}$ ,  $\overline{\text{PD}}$  and  $\overline{\text{PCWUF}}$  bits in the STATUS register can be tested to determine if a RESET condition has been caused by a power-up condition, a MCLR, Watchdog Timer (WDT) Reset, WDT Wake-up Reset, or Wake-up from SLEEP on Pin Change.

**TABLE 7-7:  $\overline{\text{TO}}$ / $\overline{\text{PD}}$ / $\overline{\text{PCWUF}}$  STATUS AFTER RESET**

PCWUF	$\overline{\text{TO}}$	$\overline{\text{PD}}$	RESET was caused by
1	1	1	Power-up (POR)
u	u	u	MCLR Reset (normal operation) <sup>(1)</sup>
u	1	0	MCLR Wake-up Reset (from SLEEP)
u	0	1	WDT Reset (normal operation)
u	0	0	WDT Wake-up Reset (from SLEEP)
0	u	u	Wake-up from SLEEP on Pin Change
x	x	x	Brown-out Reset

Legend: u = unchanged, x = unknown

**Note 1:** The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  and  $\overline{\text{PCWUF}}$  bits maintain their status (u) until a reset occurs. A low-pulse on the MCLR input does not change the  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  and  $\overline{\text{PCWUF}}$  status bits.

These STATUS bits are only affected by events listed in Table 7-8.

**TABLE 7-8: EVENTS AFFECTING  $\overline{\text{TO}}$ / $\overline{\text{PD}}$  STATUS BITS**

Event	$\overline{\text{PCWUF}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Remarks
Power-up	1	1	1	
WDT Time-out	u	0	u	No effect on $\overline{\text{PD}}$
SLEEP instruction	1	1	0	
CLRWDT instruction	u	1	1	
Wake-up from SLEEP on Pin Change	0	u	u	

Legend: u = unchanged

**Note:** A WDT time-out will occur regardless of the status of the  $\overline{\text{TO}}$  bit. A SLEEP instruction will be executed, regardless of the status of the  $\overline{\text{PD}}$  bit. Table 7-7 reflects the status of  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  after the corresponding event.

Table 7-3 lists the reset conditions for the special function registers, while Table 7-4 lists the reset conditions for all the registers.

## 7.10 Power-down Mode (SLEEP)

A device may be powered down (SLEEP) and later powered up (Wake-up from SLEEP).

### 7.10.1 SLEEP

The Power-down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the  $\overline{\text{TO}}$  bit (STATUS<4>) is set, the  $\overline{\text{PD}}$  bit (STATUS<3>) is cleared, the  $\overline{\text{PCWUF}}$  bit (STATUS<7>) is set and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low, or hi-impedance).

It should be noted that a RESET generated by a WDT time-out does not drive the MCLR/VPP pin low.

For lowest current consumption while powered down, the TOCKI input should be at VDD or VSS and the MCLR/VPP pin must be at a logic high level (VIH MCLR).

### 7.10.2 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. An external reset input on MCLR/VPP pin.
2. A Watchdog Timer Time-out Reset (if WDT was enabled).
3. A change on input pins PORTB:<0-3,7> when Wake-up on Pin Change is enabled.
4. Brown-out Reset.

These events cause a device RESET. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  and  $\overline{\text{PCWUF}}$  bits can be used to determine the cause of device RESET. The  $\overline{\text{TO}}$  bit is cleared if a WDT time-out occurred (and caused wake-up). The  $\overline{\text{PD}}$  bit, which is set on power-up, is cleared when SLEEP is invoked.

The  $\overline{\text{PCWUF}}$  bit indicates a change in state while in SLEEP at pins PORTB:<0-3,7> (since the SLEEP state was entered).

The WDT is cleared when the device wakes from SLEEP, regardless of the wake-up source.

## 7.11 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

**Note:** Microchip does not recommend code protecting windowed devices.

## 7.12 ID Locations

Four memory locations are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify.

Use only the lower 4 bits of the ID locations and always program the upper 8 bits as '1's.

**Note:** Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

## 8.0 INSTRUCTION SET SUMMARY

Each PIC16HV540 instruction is a 12-bit word divided into an OPCODE, which specifies the instruction type, and one or more operands which further specify the operation of the instruction. The PIC16HV540 instruction set summary in Table 8-2 groups the instructions into byte-oriented, bit-oriented, and literal and control operations. Table 8-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the 32 file registers is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8 or 9-bit constant or literal value.

**TABLE 8-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0 (store result in W) d = 1 (store result in file register 'f') Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
WDT	Watchdog Timer Counter
TO	Time-Out bit
PD	Power-Down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
<i>italics</i>	User defined term (font is courier)

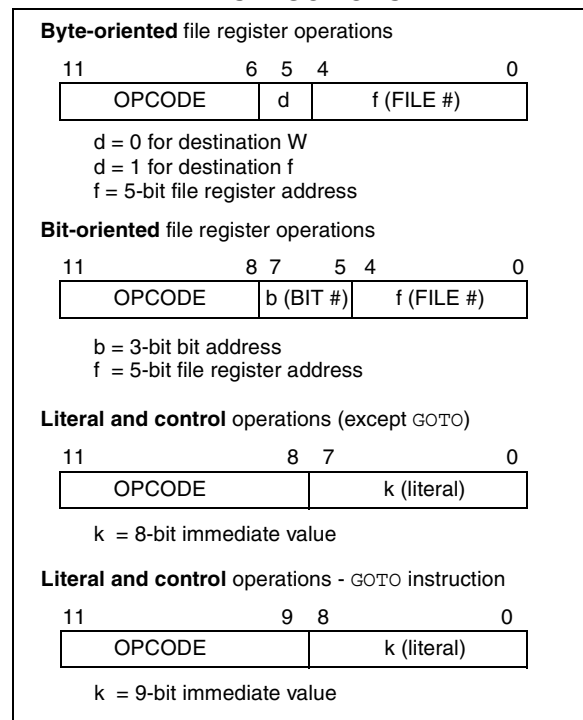
All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Figure 8-1 shows the three general formats that the instructions can have. All examples in the figure use the following format to represent a hexadecimal number:

0xhhh

where 'h' signifies a hexadecimal digit.

**FIGURE 8-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC16HV540

**TABLE 8-2: INSTRUCTION SET SUMMARY**

Mnemonic, Operands	Description	Cycles	12-Bit Opcode			Status Affected	Notes
			MSb	LSb			
ADDWF f,d	Add W and f	1	0001	11df	ffff	C,DC,Z	1,2,4
ANDWF f,d	AND W with f	1	0001	01df	ffff	Z	2,4
CLRF f	Clear f	1	0000	011f	ffff	Z	4
CLRW –	Clear W	1	0000	0100	0000	Z	
COMF f, d	Complement f	1	0010	01df	ffff	Z	
DECf f, d	Decrement f	1	0000	11df	ffff	Z	2,4
DECFSZ f, d	Decrement f, Skip if 0	1(2)	0010	11df	ffff	None	2,4
INCF f, d	Increment f	1	0010	10df	ffff	Z	2,4
INCFSZ f, d	Increment f, Skip if 0	1(2)	0011	11df	ffff	None	2,4
IORWF f, d	Inclusive OR W with f	1	0001	00df	ffff	Z	2,4
MOVF f, d	Move f	1	0010	00df	ffff	Z	2,4
MOVWF f	Move W to f	1	0000	001f	ffff	None	1,4
NOP –	No Operation	1	0000	0000	0000	None	
RLF f, d	Rotate left f through Carry	1	0011	01df	ffff	C	2,4
RRF f, d	Rotate right f through Carry	1	0011	00df	ffff	C	2,4
SUBWF f, d	Subtract W from f	1	0000	10df	ffff	C,DC,Z	1,2,4
SWAPF f, d	Swap f	1	0011	10df	ffff	None	2,4
XORWF f, d	Exclusive OR W with f	1	0001	10df	ffff	Z	2,4
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>							
BCF f, b	Bit Clear f	1	0100	bbbb	ffff	None	2,4
BSF f, b	Bit Set f	1	0101	bbbb	ffff	None	2,4
BTFSC f, b	Bit Test f, Skip if Clear	1 (2)	0110	bbbb	ffff	None	
BTFSS f, b	Bit Test f, Skip if Set	1 (2)	0111	bbbb	ffff	None	
<b>LITERAL AND CONTROL OPERATIONS</b>							
ANDLW k	AND literal with W	1	1110	kkkk	kkkk	Z	
CALL k	Call subroutine	2	1001	kkkk	kkkk	None	1
CLRWDT k	Clear Watchdog Timer	1	0000	0000	0100	<u>TO, PD</u>	
GOTO k	Unconditional branch	2	101k	kkkk	kkkk	None	
IORLW k	Inclusive OR Literal with W	1	1101	kkkk	kkkk	Z	
MOVLW k	Move Literal to W	1	1100	kkkk	kkkk	None	
OPTION k	Load OPTION register	1	0000	0000	0010	None	
RETLW k	Return, place Literal in W	2	1000	kkkk	kkkk	None	
SLEEP –	Go into standby mode	1	0000	0000	0011	<u>TO, PD,PCWUF</u>	
TRIS f	Load TRIS register	1	0000	0000	0fff	None	3
XORLW k	Exclusive OR Literal to W	1	1111	kkkk	kkkk	Z	

**Note 1:** The 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO. (See individual device data sheets, Memory Section/Indirect Data Addressing, INDF and FSR Registers)

- When an I/O register is modified as a function of itself (e.g. MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- The instruction TRIS f, where f = 5 or 6 causes the contents of the W register to be written to the tristate latches of PORTA or B respectively. A '1' forces the pin to a hi-impedance state and disables the output buffers.
- If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared (if assigned to TMR0).

## **ADDWF**      **Add W and f**

**Syntax:**            `[ label ] ADDWF f,d`

**Operands:**         $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**         $(W) + (f) \rightarrow (\text{dest})$

**Status Affected:** C, DC, Z

**Encoding:**

0001	11df	ffff
------	------	------

**Description:**     Add the contents of the W register and register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is '1' the result is stored back in register 'f'.

**Words:**            1

**Cycles:**           1

**Example:**         `ADDWF FSR, 0`

Before Instruction

W = 0x17  
 FSR = 0xC2

After Instruction

W = 0xD9  
 FSR = 0xC2

## **ANDLW**      **And literal with W**

**Syntax:**            `[ label ] ANDLW k`

**Operands:**         $0 \leq k \leq 255$

**Operation:**         $(W).AND. (k) \rightarrow (W)$

**Status Affected:** Z

**Encoding:**

1110	kkkk	kkkk
------	------	------

**Description:**     The contents of the W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

**Words:**            1

**Cycles:**           1

**Example:**         `ANDLW 0x5F`

Before Instruction

W = 0xA3

After Instruction

W = 0x03

## **ANDWF**      **AND W with f**

**Syntax:**            `[ label ] ANDWF f,d`

**Operands:**         $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**         $(W).AND. (f) \rightarrow (\text{dest})$

**Status Affected:** Z

**Encoding:**

0001	01df	ffff
------	------	------

**Description:**     The contents of the W register are AND'ed with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is '1' the result is stored back in register 'f'.

**Words:**            1

**Cycles:**           1

**Example:**         `ANDWF FSR, 1`

Before Instruction

W = 0x17  
 FSR = 0xC2

After Instruction

W = 0x17  
 FSR = 0x02

## **BCF**          **Bit Clear f**

**Syntax:**            `[ label ] BCF f,b`

**Operands:**         $0 \leq f \leq 31$   
 $0 \leq b \leq 7$

**Operation:**         $0 \rightarrow (f<b>)$

**Status Affected:** None

**Encoding:**

0100	bbbf	ffff
------	------	------

**Description:**     Bit 'b' in register 'f' is cleared.

**Words:**            1

**Cycles:**           1

**Example:**         `BCF FLAG_REG, 7`

Before Instruction

FLAG\_REG = 0xC7

After Instruction

FLAG\_REG = 0x47

# PIC16HV540

## **BSF**      **Bit Set f**

Syntax:      [ *label* ] BSF f,b  
 Operands:     $0 \leq f \leq 31$   
                $0 \leq b \leq 7$   
 Operation:     $1 \rightarrow (f<b>)$   
 Status Affected: None  
 Encoding:    

0101	bbbf	ffff
------	------	------

  
 Description:    Bit 'b' in register 'f' is set.  
 Words:        1  
 Cycles:       1  
 Example:      BSF      FLAG\_REG,    7

Before Instruction  
     FLAG\_REG = 0x0A  
 After Instruction  
     FLAG\_REG = 0x8A

## **BTFSS**      **Bit Test f, Skip if Set**

## **BTFSS**      **Bit Test f, Skip if Set**

Syntax:      [ *label* ] BTFSS f,b  
 Operands:     $0 \leq f \leq 31$   
                $0 \leq b < 7$   
 Operation:    skip if (f<b>) = 1  
 Status Affected: None  
 Encoding:    

0111	bbbf	ffff
------	------	------

  
 Description:    If bit 'b' in register 'f' is '1' then the next instruction is skipped.  
                   If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and an NOP is executed instead, making this a 2 cycle instruction.  
 Words:        1  
 Cycles:       1(2)  
 Example:      HERE    BTFSS    FLAG, 1  
                   FALSE    GOTO    PROCESS\_CODE  
                   TRUE     •  
                           •  
                           •

Before Instruction  
     PC            =    address (HERE)  
 After Instruction  
     if FLAG<1>    =    0,  
     PC            =    address (FALSE);  
     if FLAG<1>    =    1,  
     PC            =    address (TRUE)

## **BTFSC**      **Bit Test f, Skip if Clear**

Syntax:      [ *label* ] BTFSC f,b  
 Operands:     $0 \leq f \leq 31$   
                $0 \leq b \leq 7$   
 Operation:    skip if (f<b>) = 0  
 Status Affected: None  
 Encoding:    

0110	bbbf	ffff
------	------	------

  
 Description:    If bit 'b' in register 'f' is 0 then the next instruction is skipped.  
                   If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and an NOP is executed instead, making this a 2 cycle instruction.  
 Words:        1  
 Cycles:       1(2)  
 Example:      HERE    BTFSC    FLAG, 1  
                   FALSE    GOTO    PROCESS\_CODE  
                   TRUE     •  
                           •  
                           •

Before Instruction  
     PC            =    address (HERE)  
 After Instruction  
     if FLAG<1>    =    0,  
     PC            =    address (TRUE);  
     if FLAG<1>    =    1,  
     PC            =    address (FALSE)

## CALL Subroutine Call

**Syntax:** [ *label* ] CALL k

**Operands:**  $0 \leq k \leq 255$

**Operation:** (PC) + 1 → Top of Stack;  
 k → PC<7:0>;  
 (STATUS<6:5>) → PC<10:9>;  
 0 → PC<8>

**Status Affected:** None

**Encoding:**

1001	kkkk	kkkk
------	------	------

**Description:** Subroutine call. First, return address (PC+1) is pushed onto the stack. The eight bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. CALL is a two cycle instruction.

**Words:** 1

**Cycles:** 2

**Example:**     HERE     CALL     THERE

Before Instruction  
 PC = address (HERE)

After Instruction  
 PC = address (THERE)  
 TOS = address (HERE + 1)

## CLRF Clear f

**Syntax:** [ *label* ] CLRF f

**Operands:**  $0 \leq f \leq 31$

**Operation:** 00h → (f);  
 1 → Z

**Status Affected:** Z

**Encoding:**

0000	011f	ffff
------	------	------

**Description:** The contents of register 'f' are cleared and the Z bit is set.

**Words:** 1

**Cycles:** 1

**Example:**     CLRF     FLAG\_REG

Before Instruction  
 FLAG\_REG = 0x5A

After Instruction  
 FLAG\_REG = 0x00  
 Z = 1

## CLRW Clear W

**Syntax:** [ *label* ] CLRW

**Operands:** None

**Operation:** 00h → (W);  
 1 → Z

**Status Affected:** Z

**Encoding:**

0000	0100	0000
------	------	------

**Description:** The W register is cleared. Zero bit (Z) is set.

**Words:** 1

**Cycles:** 1

**Example:**     CLRW

Before Instruction  
 W = 0x5A

After Instruction  
 W = 0x00  
 Z = 1

## CLRWDW Clear Watchdog Timer

**Syntax:** [ *label* ] CLRWDW

**Operands:** None

**Operation:** 00h → WDT;  
 0 → WDT prescaler (if assigned);  
 1 →  $\overline{TO}$ ;  
 1 →  $\overline{PD}$

**Status Affected:**  $\overline{TO}$ ,  $\overline{PD}$

**Encoding:**

0000	0000	0100
------	------	------

**Description:** The CLRWDW instruction resets the WDT. It also resets the prescaler, if the prescaler is assigned to the WDT and not Timer0. Status bits  $\overline{TO}$  and  $\overline{PD}$  are set.

**Words:** 1

**Cycles:** 1

**Example:**     CLRWDW

Before Instruction  
 WDT counter = ?

After Instruction  
 WDT counter = 0x00  
 WDT prescale = 0  
 $\overline{TO}$  = 1  
 $\overline{PD}$  = 1

# PIC16HV540

## COMF Complement f

Syntax: [ *label* ] COMF f,d

Operands:  $0 \leq f \leq 31$   
 $d \in [0,1]$

Operation:  $(\bar{f}) \rightarrow (\text{dest})$

Status Affected: Z

Encoding: 

0010	01df	ffff
------	------	------

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example: COMF REG1, 0

Before Instruction

REG1 = 0x13

After Instruction

REG1 = 0x13

W = 0xEC

## DECF Decrement f

Syntax: [ *label* ] DECF f,d

Operands:  $0 \leq f \leq 31$   
 $d \in [0,1]$

Operation:  $(f) - 1 \rightarrow (\text{dest})$

Status Affected: Z

Encoding: 

0000	11df	ffff
------	------	------

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example: DECF CNT, 1

Before Instruction

CNT = 0x01

Z = 0

After Instruction

CNT = 0x00

Z = 1

## DECFSZ Decrement f, Skip if 0

Syntax: [ *label* ] DECFSZ f,d

Operands:  $0 \leq f \leq 31$   
 $d \in [0,1]$

Operation:  $(f) - 1 \rightarrow d$ ; skip if result = 0

Status Affected: None

Encoding: 

0010	11df	ffff
------	------	------

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

If the result is 0, the next instruction, which is already fetched, is discarded and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example: HERE DECFSZ CNT, 1

GOTO LOOP

CONTINUE •  
•  
•

Before Instruction

PC = address (HERE)

After Instruction

CNT = CNT - 1;

if CNT = 0,

PC = address (CONTINUE);

if CNT  $\neq$  0,

PC = address (HERE+1)

## GOTO Unconditional Branch

Syntax: [ *label* ] GOTO k

Operands:  $0 \leq k \leq 511$

Operation:  $k \rightarrow PC\langle 8:0 \rangle$ ;  
 $STATUS\langle 6:5 \rangle \rightarrow PC\langle 10:9 \rangle$

Status Affected: None

Encoding: 

101k	kkkk	kkkk
------	------	------

Description: GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits  $\langle 8:0 \rangle$ . The upper bits of PC are loaded from  $STATUS\langle 6:5 \rangle$ . GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example: GOTO THERE

After Instruction

PC = address (THERE)



## **INCF**                      **Increment f**

**Syntax:**                    `[ label ] INCF f,d`

**Operands:**                 $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**                 $(f) + 1 \rightarrow (\text{dest})$

**Status Affected:**        Z

**Encoding:**

0010	10df	ffff
------	------	------

**Description:**             The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

**Words:**                    1

**Cycles:**                   1

**Example:**                 `INCF    CNT,    1`

**Before Instruction**

CNT = 0xFF  
Z = 0

**After Instruction**

CNT = 0x00  
Z = 1

## **INCFSZ**                    **Increment f, Skip if 0**

**Syntax:**                    `[ label ] INCFSZ f,d`

**Operands:**                 $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**                 $(f) + 1 \rightarrow (\text{dest})$ , skip if result = 0

**Status Affected:**        None

**Encoding:**

0011	11df	ffff
------	------	------

**Description:**             The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.  
 If the result is 0, then the next instruction, which is already fetched, is discarded and an NOP is executed instead making it a two cycle instruction.

**Words:**                    1

**Cycles:**                   1(2)

**Example:**                 `HERE        INCFSZ    CNT,    1`  
`GOTO        LOOP`  
`CONTINUE    •`  
`•`  
`•`

**Before Instruction**

PC = address (HERE)

**After Instruction**

CNT = CNT + 1;  
 if CNT = 0,  
 PC = address (CONTINUE);  
 if CNT ≠ 0,  
 PC = address (HERE +1)

## **IORLW**                      **Inclusive OR literal with W**

**Syntax:**                    `[ label ] IORLW k`

**Operands:**                 $0 \leq k \leq 255$

**Operation:**                 $(W) .OR. (k) \rightarrow (W)$

**Status Affected:**        Z

**Encoding:**

1101	kkkk	kkkk
------	------	------

**Description:**             The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

**Words:**                    1

**Cycles:**                   1

**Example:**                 `IORLW    0x35`

**Before Instruction**

W = 0x9A

**After Instruction**

W = 0xBF  
Z = 0

## **IORWF**                      **Inclusive OR W with f**

**Syntax:**                    `[ label ] IORWF f,d`

**Operands:**                 $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**                 $(W).OR. (f) \rightarrow (\text{dest})$

**Status Affected:**        Z

**Encoding:**

0001	00df	ffff
------	------	------

**Description:**             Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

**Words:**                    1

**Cycles:**                   1

**Example:**                 `IORWF                    RESULT, 0`

**Before Instruction**

RESULT = 0x13  
W = 0x91

**After Instruction**

RESULT = 0x13  
W = 0x93  
Z = 0

# PIC16HV540

## MOVF Move f

Syntax: [ *label* ] MOVF f,d

Operands:  $0 \leq f \leq 31$   
 $d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding: 

0010	00df	ffff
------	------	------

Description: The contents of register 'f' is moved to destination 'd'. If 'd' is 0, destination is the W register. If 'd' is 1, the destination is file register 'f'. 'd' is 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example: MOVF FSR, 0

After Instruction

W = value in FSR register

## MOVLW Move Literal to W

Syntax: [ *label* ] MOVLW k

Operands:  $0 \leq k \leq 255$

Operation: k → (W)

Status Affected: None

Encoding: 

1100	kkkk	kkkk
------	------	------

Description: The eight bit literal 'k' is loaded into the W register. The don't cares will assemble as 0s.

Words: 1

Cycles: 1

Example: MOVLW 0x5A

After Instruction

W = 0x5A

## MOVWF Move W to f

Syntax: [ *label* ] MOVWF f

Operands:  $0 \leq f \leq 31$

Operation: (W) → (f)

Status Affected: None

Encoding: 

0000	001f	ffff
------	------	------

Description: Move data from the W register to register 'f'.

Words: 1

Cycles: 1

Example: MOVWF TEMP\_REG

Before Instruction

TEMP\_REG = 0xFF

W = 0x4F

After Instruction

TEMP\_REG = 0x4F

W = 0x4F

## NOP No Operation

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding: 

0000	0000	0000
------	------	------

Description: No operation.

Words: 1

Cycles: 1

Example: NOP

## OPTION Load OPTION Register

**Syntax:** `[label] OPTION`

**Operands:** None

**Operation:** (W) → OPTION

**Status Affected:** None

**Encoding:**

0000	0000	0010
------	------	------

**Description:** The content of the W register is loaded into the OPTION register.

**Words:** 1

**Cycles:** 1

**Example**            `OPTION`  
                              `N`

**Before Instruction**  
W = 0x07

**After Instruction**  
OPTION = 0x07

## RETLW Return with Literal in W

**Syntax:** `[label] RETLW k`

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $k \rightarrow (W)$ ;  
TOS → PC

**Status Affected:** None

**Encoding:**

1000	kkkk	kkkk
------	------	------

**Description:** The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

**Words:** 1

**Cycles:** 2

**Example:** `CALL TABLE ;W contains`  
                              `;table offset`  
                              `;value.`  
                              •           `;W now has table`  
                              •           `;value.`  
                              •

`TABLE           ADDWF PC ;W = offset`  
                  `RETLW k1 ;Begin table`  
                  `RETLW k2 ;`  
                  •  
                  •  
                  •  
                  `RETLW kn ; End of table`

**Before Instruction**  
W = 0x07

**After Instruction**  
W = value of k8

## RLF Rotate Left f through Carry

**Syntax:** `[label] RLF f,d`

**Operands:**  $0 \leq f \leq 31$   
 $d \in [0,1]$

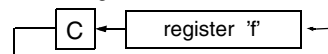
**Operation:** See description below

**Status Affected:** C

**Encoding:**

0011	01df	ffff
------	------	------

**Description:** The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



**Words:** 1

**Cycles:** 1

**Example:** `RLF REG1,0`

**Before Instruction**  
REG1 = 1110 0110  
C = 0

**After Instruction**  
REG1 = 1110 0110  
W = 1100 1100  
C = 1

## RRF Rotate Right f through Carry

**Syntax:** `[label] RRF f,d`

**Operands:**  $0 \leq f \leq 31$   
 $d \in [0,1]$

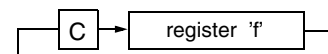
**Operation:** See description below

**Status Affected:** C

**Encoding:**

0011	00df	ffff
------	------	------

**Description:** The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



**Words:** 1

**Cycles:** 1

**Example:** `RRF REG1,0`

**Before Instruction**  
REG1 = 1110 0110  
C = 0

**After Instruction**  
REG1 = 1110 0110  
W = 0111 0011  
C = 0

# PIC16HV540

<b>SLEEP</b>	<b>Enter SLEEP Mode</b>			
Syntax:	[label] SLEEP			
Operands:	None			
Operation:	00h → WDT; 0 → $\overline{\text{WDT}}$ prescaler; 1 → $\overline{\text{TO}}$ ; 0 → $\overline{\text{PD}}$ 1 → $\overline{\text{PCWUF}}$			
Status Affected:	$\overline{\text{TO}}$ , $\overline{\text{PD}}$ , $\overline{\text{PCWUF}}$			
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0011</td></tr></table>	0000	0000	0011
0000	0000	0011		
Description:	Time-out status bit ( $\overline{\text{TO}}$ ) is set. The power down status bit ( $\overline{\text{PD}}$ ) is cleared. The WDT and its prescaler are cleared.  The processor is put into SLEEP mode with the oscillator stopped. See section on SLEEP for more details.			
Words:	1			
Cycles:	1			
Example:	SLEEP			

<b>SUBWF</b>	<b>Subtract W from f</b>			
Syntax:	[label] SUBWF f,d			
Operands:	$0 \leq f \leq 31$ $d \in [0,1]$			
Operation:	$(f) - (W) \rightarrow (\text{dest})$			
Status Affected:	C, DC, Z			
Encoding:	<table border="1"><tr><td>0000</td><td>10df</td><td>ffff</td></tr></table>	0000	10df	ffff
0000	10df	ffff		
Description:	Subtract (2's complement method) the W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.			
Words:	1			
Cycles:	1			
<u>Example 1:</u>	SUBWF REG1, 1			
Before Instruction	REG1 = 3 W = 2 C = ?			
After Instruction	REG1 = 1 W = 2 C = 1 ; result is positive			
<u>Example 2:</u>				
Before Instruction	REG1 = 2 W = 2 C = ?			
After Instruction	REG1 = 0 W = 2 C = 1 ; result is zero			
<u>Example 3:</u>				
Before Instruction	REG1 = 1 W = 2 C = ?			
After Instruction	REG1 = FF W = 2 C = 0 ; result is negative			

## SWAPF **Swap Nibbles in f**

**Syntax:** `[label] SWAPF f,d`

**Operands:**  $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**  $(f<3:0>) \rightarrow (dest<7:4>);$   
 $(f<7:4>) \rightarrow (dest<3:0>)$

**Status Affected:** None

**Encoding:**

0011	10df	ffff
------	------	------

**Description:** The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

**Words:** 1

**Cycles:** 1

**Example** `SWAPF REG1, 0`

**Before Instruction**  
`REG1 = 0xA5`

**After Instruction**  
`REG1 = 0xA5`  
`W = 0X5A`

## TRIS **Load TRIS Register**

**Syntax:** `[label] TRIS f`

**Operands:**  $f = 5, 6$  or  $7$

**Operation:**  $(W) \rightarrow \text{TRIS register } f$

**Status Affected:** None

**Encoding:**

0000	0000	0fff
------	------	------

**Description:** TRIS register 'f' ( $f = 5, 6,$  or  $7^*$ ) is loaded with the contents of the W register

**Words:** 1

**Cycles:** 1

**Example** `TRIS PORTA`

**Before Instruction**  
`W = 0XA5`

**After Instruction**  
`TRISA = 0XA5`

\*A TRIS 7 operation will update the OPTION2 SFR.

## XORLW **Exclusive OR literal with W**

**Syntax:** `[label] XORLW k`

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $(W) .XOR. k \rightarrow (W)$

**Status Affected:** Z

**Encoding:**

1111	kkkk	kkkk
------	------	------

**Description:** The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

**Words:** 1

**Cycles:** 1

**Example:** `XORLW 0xAF`

**Before Instruction**  
`W = 0xB5`

**After Instruction**  
`W = 0x1A`

## XORWF **Exclusive OR W with f**

**Syntax:** `[label] XORWF f,d`

**Operands:**  $0 \leq f \leq 31$   
 $d \in [0,1]$

**Operation:**  $(W) .XOR. (f) \rightarrow (dest)$

**Status Affected:** Z

**Encoding:**

0001	10df	ffff
------	------	------

**Description:** Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

**Words:** 1

**Cycles:** 1

**Example** `XORWF REG, 1`

**Before Instruction**  
`REG = 0xAF`  
`W = 0xB5`

**After Instruction**  
`REG = 0x1A`  
`W = 0xB5`

# PIC16HV540

---

NOTES:

## 9.0 DEVELOPMENT SUPPORT

The PICmicro<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> IDE Software
- Assemblers/Compilers/Linkers
  - MPASM Assembler
  - MPLAB-C17 and MPLAB-C18 C Compilers
  - MPLINK/MPLIB Linker/Librarian
- Simulators
  - MPLAB-SIM Software Simulator
- Emulators
  - MPLAB-ICE Real-Time In-Circuit Emulator
  - PICMASTER<sup>®</sup>/PICMASTER-CE In-Circuit Emulator
  - ICEPIC<sup>™</sup>
- In-Circuit Debugger
  - MPLAB-ICD for PIC16F877
- Device Programmers
  - PRO MATE<sup>®</sup> II Universal Programmer
  - PICSTART<sup>®</sup> Plus Entry-Level Prototype Programmer
- Low-Cost Demonstration Boards
  - SIMICE
  - PICDEM-1
  - PICDEM-2
  - PICDEM-3
  - PICDEM-17
  - SEEVAL<sup>®</sup>
  - KEELOQ<sup>®</sup>

### 9.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. MPLAB is a Windows<sup>®</sup>-based application which contains:

- Multiple functionality
  - editor
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
- A full featured editor
- A project manager
- Customizable tool bar and key mapping
- A status bar
- On-line help

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PICmicro tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - object code

The ability to use MPLAB with Microchip's simulator, MPLAB-SIM, allows a consistent platform and the ability to easily switch from the cost-effective simulator to the full featured emulator with minimal retraining.

### 9.2 MPASM Assembler

MPASM is a full featured universal macro assembler for all PICmicro MCU's. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK.

MPASM has a command line interface and a Windows shell and can be used as a standalone application on a Windows 3.x or greater system. MPASM generates relocatable object files, Intel standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file which contains source lines and generated machine code, and a COD file for MPLAB debugging.

MPASM features include:

- MPASM and MPLINK are integrated into MPLAB projects.
- MPASM allows user defined macros to be created for streamlined assembly.
- MPASM allows conditional assembly for multi purpose source files.
- MPASM directives allow complete control over the assembly process.

### 9.3 MPLAB-C17 and MPLAB-C18 C Compilers

The MPLAB-C17 and MPLAB-C18 Code Development Systems are complete ANSI 'C' compilers and integrated development environments for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

## **9.4 MPLINK/MPLIB Linker/Librarian**

MPLINK is a relocatable linker for MPASM and MPLAB-C17 and MPLAB-C18. It can link relocatable objects from assembly or C source files along with pre-compiled libraries using directives from a linker script.

MPLIB is a librarian for pre-compiled code to be used with MPLINK. When a routine from a library is called from another source file, only the modules that contains that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. MPLIB manages the creation and modification of library files.

MPLINK features include:

- MPLINK works with MPASM and MPLAB-C17 and MPLAB-C18.
- MPLINK allows all memory areas to be defined as sections to provide link-time flexibility.

MPLIB features include:

- MPLIB makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB helps keep code maintainable by grouping related modules together.
- MPLIB commands allow libraries to be created and modules to be added, listed, replaced, deleted, or extracted.

## **9.5 MPLAB-SIM Software Simulator**

The MPLAB-SIM Software Simulator allows code development in a PC host environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file or user-defined key press to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPLAB-C18 and MPASM. The Software Simulator offers the flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

## **9.6 MPLAB-ICE High Performance Universal In-Circuit Emulator with MPLAB IDE**

The MPLAB-ICE Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of MPLAB-ICE is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB-ICE allows expansion to support new PICmicro microcontrollers.

The MPLAB-ICE Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows 3.x/95/98 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE 2000 is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PICmicro MCU.

## **9.7 PICMASTER/PICMASTER CE**

The PICMASTER system from Microchip Technology is a full-featured, professional quality emulator system. This flexible in-circuit emulator provides a high-quality, universal platform for emulating Microchip 8-bit PICmicro microcontrollers (MCUs). PICMASTER systems are sold worldwide, with a CE compliant model available for European Union (EU) countries.

## **9.8 ICEPIC**

ICEPIC is a low-cost in-circuit emulation solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X, and PIC16CXXX families of 8-bit one-time-programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules or daughter boards. The emulator is capable of emulating without target application circuitry being present.

## **9.9 MPLAB-ICD In-Circuit Debugger**

Microchip's In-Circuit Debugger, MPLAB-ICD, is a powerful, low-cost run-time development tool. This tool is based on the flash PIC16F877 and can be used to develop for this and other PICmicro microcontrollers from the PIC16CXXX family. MPLAB-ICD utilizes the In-Circuit Debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming protocol, offers cost-effective in-circuit flash programming and debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time. The MPLAB-ICD is also a programmer for the flash PIC16F87X family.



## **9.10 PRO MATE II Universal Programmer**

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode. PRO MATE II is CE compliant.

The PRO MATE II has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE II can read, verify or program PICmicro devices. It can also set code-protect bits in this mode.

## **9.11 PICSTART Plus Entry Level Development System**

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

PICSTART Plus supports all PICmicro devices with up to 40 pins. Larger pin count devices such as the PIC16C92X, and PIC17C76X may be supported with an adapter socket. PICSTART Plus is CE compliant.

## **9.12 SIMICE Entry-Level Hardware Simulator**

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PICmicro 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

## **9.13 PICDEM-1 Low-Cost PICmicro Demonstration Board**

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with

the PICDEM-1 board, on a PRO MATE II or PICSTART-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

## **9.14 PICDEM-2 Low-Cost PIC16CXX Demonstration Board**

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I<sup>2</sup>C bus and separate headers for connection to an LCD module and a keypad.

## **9.15 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board**

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

## **9.16 PICDEM-17**

The PICDEM-17 is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756, PIC17C762, and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included, and the user may erase it and program it with the other sample programs using the PRO MATE II or PICSTART Plus device programmers and easily debug and test the sample code. In addition, PICDEM-17 supports down-loading of programs to and executing out of external FLASH memory on board. The PICDEM-17 is also usable with the MPLAB-ICE or PICMASTER emulator, and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

## **9.17 SEEVAL Evaluation and Programming System**

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

## **9.18 KEELOQ Evaluation and Programming Tools**

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

**TABLE 9-1: DEVELOPMENT TOOLS FROM MICROCHIP**

Tool	PIC12CXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	24CXX/ 25CXX/ 93CXX	HC5XX	MCRFXXX	MCP2510
<b>Software Tools</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
<b>Development Environment</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
<b>Compiler</b>																		
<b>MPASM/MPLINK</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
<b>MPLAB®-ICE</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
<b>PICMASTER/PICMASTER-CE</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
<b>ICEPIC™ Low-Cost In-Circuit Emulator</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
<b>MPLAB®-ICD In-Circuit Debugger</b>				✓*			✓*			✓								
<b>PICSTART® Plus Low-Cost Universal Dev. Kit</b>	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
<b>PRO MATE® II Universal Programmer</b>	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
<b>SIMICE</b>	✓		✓															
<b>PICDEM-1</b>			✓		✓		✓†		✓			✓						
<b>PICDEM-2</b>				✓†			✓†											
<b>PICDEM-3</b>																		
<b>PICDEM-14A</b>																		
<b>PICDEM-17</b>			✓										✓					
<b>KEELOO® Evaluation Kit</b>																✓		
<b>KEELOO Transponder Kit</b>																✓		
<b>microID™ Programmer's Kit</b>																	✓	
<b>125 kHz microID Developer's Kit</b>																	✓	
<b>125 kHz Anticollision microID Developer's Kit</b>																	✓	
<b>13.56 MHz Anticollision microID Developer's Kit</b>																	✓	
<b>MCP2510 CAN Developer's Kit</b>																	✓	✓

\* Contact the Microchip Technology Inc. web site at [www.microchip.com](http://www.microchip.com) for information on how to use the MPLAB®-ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77

\*\* Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

# PIC16HV540

---

NOTES:

## 10.0 ELECTRICAL CHARACTERISTICS - PIC16HV540

### Absolute Maximum Ratings†

Ambient temperature under bias .....	-20°C to +85°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to VSS .....	0 to +16V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS.....	0 to +14V
Voltage on all other pins with respect to VSS .....	-0.6V to (VDD + 0.6V)
Total power dissipation <sup>(1)</sup> .....	800 mW
Max. current out of VSS pin .....	150 mA
Max. current into VDD pin .....	100 mA
Max. current into an input pin (T0CKI only) .....	±500 $\mu$ A
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) .....	±20 mA
Max. output current sunk by any I/O pin .....	25 mA
Max. output current sourced by any I/O pin .....	10 mA
Max. output current sourced by a single I/O port A or B .....	40 mA
Max. output current sourced by a single I/O port A or B .....	50 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**2:** Voltage spikes below VSS at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100 $\Omega$  should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin rather than pulling this pin directly to VSS.

† NOTICE: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

# PIC16HV540

## 10.1 DC Characteristics: PIC16HV540-04, 20 (Commercial) PIC16HV540-04I, 20I (Industrial)

DC Characteristics Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial)				
Characteristic	Sym.	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
Supply Voltage	VDD	3.5 4.5	—	15 15	V V	LP, XT and RC modes HS mode
RAM Data Retention Voltage <sup>(2)</sup>	VDR	—	1.5*	—	V	Device in SLEEP mode
VDD start voltage to ensure Power-on Reset	VPOR	—	VSS	—	V	See section on Power-on Reset for details.
VDD rise rate to ensure Power-on Reset	SVDD	0.05 VDD			V/ms	See Section 7.4 for details on Power-on Reset
Supply Current <sup>(3)</sup> HS option XT and RC <sup>(4)</sup> options LP option	IDD	— — —	5 1.8 300	20 3.3 500	mA mA μA	FOSC = 20 MHz, VDD = 15V, VREG = 5V FOSC = 4 MHz, VDD = 15V, VREG = 5V FOSC = 32 kHz, VDD = 15V, VREG = 5V, WDT disabled
Power-down Current <sup>(5)(6)</sup>	IPD	— — — —	4.5 0.25 1.8 1.4	20 14 10 5	μA μA μA μA	VDD = 15V, VREG = 5V sleep timer enable, BOD disabled VDD = 15V, VREG = 3V sleep timer enable, BOD disabled VDD = 15V, VREG = 5V sleep timer disabled, BOD disabled VDD = 15V, VREG = 3V sleep timer disabled, BOD disabled
Brown-out Current		—	0.5	—	μA	VDD = 15V, VREG = 5V, BOD enabled
Brown-out Detector Threshold	BVDD	2.7 1.8	3.1 2.2	4.2 2.8	V V	VDD = 15V, VREG = 5V* <sup>(7)</sup> VDD = 15V, VREG = 3V* <sup>(7)</sup>
Regulation Voltage	VIO	2 4	3 5	4.5 6	V V	VDD = 15V, VREG = 3V, Unloaded outputs, SLEEP VDD = 15V, VREG = 5V, Unloaded outputs, SLEEP

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ”) column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

**2:** This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

**3:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tristated, pulled to VSS, T0CKI = VDD, MCLR = VDD; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode.

**4:** Does not include current through REXT. The current through the resistor can be estimated by the formula: IR = VDD/2REXT (mA) with REXT in kΩ.

**5:** The power down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

**6:** The oscillator start-up time can be as much as 8 seconds for XT and LP oscillator selection, if the SLEEP mode is exited or during initial power-up.

**7:** See Section 7.6.1 for additional information.

## 10.2 DC Characteristics: PIC16HV540-04, 20 (Commercial) PIC16HV540-04I, 20I (Industrial)

DC Characteristics All Pins Except Power Supply Pins		Standard Operating Conditions (unless otherwise specified) Operating Temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial)					Conditions
Characteristic	Sym.	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions	
<b>Input Low Voltage</b>	VIL						
I/O Ports PORTA		VSS	—	0.10 VREG	V	Pin at Hi-impedance	
MCLR (Schmitt Trigger)		VSS	—	0.10 VREG	V		
T0CKI (Schmitt Trigger)		VSS	—	0.10 VREG	V		
OSC1 (Schmitt Trigger)		VSS	—	0.10 VREG	V		
OSC1		VSS	—	0.3 VREG	V		
I/O Ports PORTB		VSS	—	0.10 VREG	V		
<b>Input High Voltage</b>	VIH						
I/O Ports PORTA		0.25 VREG+0.8V	—	VREG	V	For all VREG	
MCLR (Schmitt Trigger)		0.85 VREG	—	VDD	V		
T0CKI (Schmitt Trigger)		0.85 VREG	—	VDD	V		
OSC1 (Schmitt Trigger)		4.5V	—	VDD	V		
OSC1		4.5V	—	VDD	V		
I/O Ports PORTB		0.25 VREG+0.8V	—	VDD	V		
Hysteresis of Schmitt Trigger inputs	VHYS	0.15 VREG*	—	—	V		
<b>Input Leakage Current<sup>(3)</sup></b>	IIL						
I/O Ports PORTA		-1.0	0.5	+1.0	μA	VSS ≤ VPIN ≤ VIO, Pin at Hi-impedance	
I/O Ports PORTB		-1.0	0.5	+1.0	μA		
MCLR		-5.0		+5.0	μA	VPIN = VSS +0.25V <sup>(2)</sup>	
T0CKI		-3.0	0.5	+3.0	μA		
OSC1		-3.0	0.5	+3.0	μA	VPIN = VDD <sup>(2)</sup>	
						VSS ≤ VPIN ≤ VDD, HS, XT, and LP options	
<b>Output Low Voltage</b>	VOL						
I/O Ports PORTA		—	—	0.6	V	VDD = 15V, VREG = 5V, IOL = 8.7 mA VDD = 15V, VREG = 3V, IOL = 5.0 mA VDD = 15V, VREG = 5V, IOL = 1.2 mA, (RC option only) VDD = 15V, VREG = 3V, IOL = 1.0 mA, (RC option only)	
OSC2/CLKOUT		—	—	0.6	V		
I/O Ports PORTB		—	—	0.6	V		
						VDD = 15V, VIO = 3V, IOH = -2.0 mA VDD = 15V, VIO = 5V, IOH = -3.0 mA VDD = 15V, VIO = 3V, IOH = -0.5 mA (RC option only) VDD = 15V, VIO = 5V, IOH = -1.0 mA (RC option only)	
<b>Output High Voltage</b>	VOH						
I/O ports <sup>(3)</sup> PORTA		VREG-0.7	—	—	V	VDD = 15V, VIO = 3V, IOH = -2.0 mA VDD = 15V, VIO = 5V, IOH = -3.0 mA VDD = 15V, VIO = 3V, IOH = -0.5 mA (RC option only) VDD = 15V, VIO = 5V, IOH = -1.0 mA (RC option only)	
OSC2/CLKOUT		VREG-0.7	—	—	V		
I/O Ports PORTB		VDD-0.7	—	—	V		
						VDD = 15V, VIO = 5V, IOH = -5.4 mA	
<b>Threshold Voltage</b>							
I/O Ports PORTB [7]	VLEV	VDD-1.5	VDD-1.0	VDD-0.5	V	VDD = 15V	

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

**2:** The leakage current on the MCLR/VP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltage.

**3:** Negative current is defined as coming out of the pin.

**4:** For the RC option, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16HV540 be driven with external clock in RC mode.

# PIC16HV540

## 10.3 Timing Parameter Symbology and Load Conditions

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

<b>T</b>		<b>T</b>	Time
<b>F</b>	Frequency		

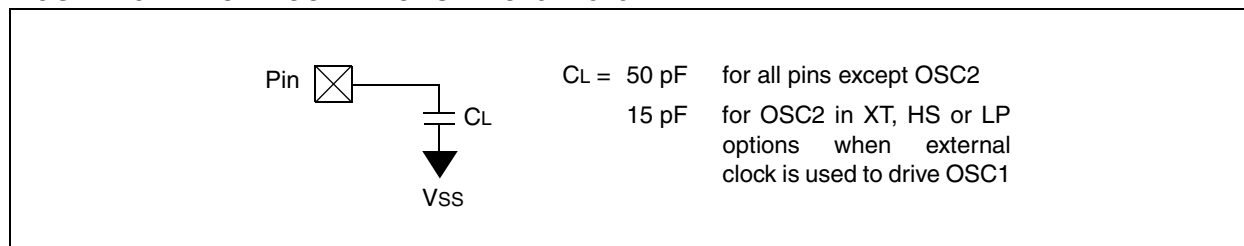
Lowercase subscripts (pp) and their meanings:

<b>pp</b>		mc	MCLR
2	to	osc	oscillator
ck	CLKOUT	os	OSC1
cy	cycle time	t0	T0CKI
drt	device reset timer	wdt	watchdog timer
io	I/O port		

Uppercase letters and their meanings:

<b>S</b>		<b>P</b>	Period
<b>F</b>	Fall	<b>R</b>	Rise
<b>H</b>	High	<b>V</b>	Valid
<b>I</b>	Invalid (Hi-impedance)	<b>Z</b>	Hi-impedance
<b>L</b>	Low		

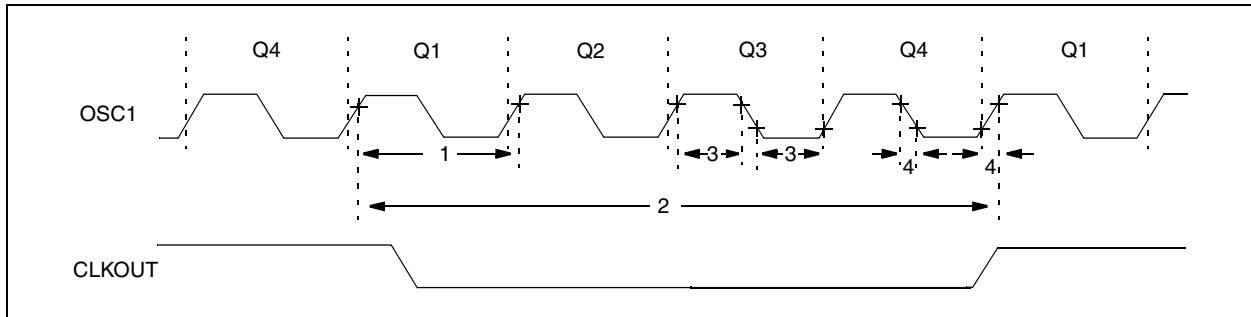
**FIGURE 10-1: LOAD CONDITIONS - PIC16HV540**





## 10.4 Timing Diagrams and Specifications

**FIGURE 10-2: EXTERNAL CLOCK TIMING - PIC16HV540**



**TABLE 10-1: EXTERNAL CLOCK TIMING REQUIREMENTS - PIC16HV540**

AC Characteristics Standard Operating Conditions (unless otherwise specified)							
Operating Temperature 0°C ≤ TA ≤ +70°C (commercial)							
-40°C ≤ TA ≤ +85°C (industrial)							
Parameter No.	Sym.	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
	FOSC	External CLKIN Frequency <sup>(2)</sup>	DC	—	4.0	MHz	RC osc mode
			DC	—	2.0	MHz	HS osc mode
			DC	—	4.0	MHz	XT osc mode
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency <sup>(2)</sup>	DC	—	4.0	MHz	RC osc mode
			0.1	—	2.0	MHz	HS osc mode
			0.1	—	4.0	MHz	XT osc mode
			5	—	200	kHz	LP osc mode
1	TOSC	External CLKIN Period <sup>(2)</sup>	250	—	—	ns	RC osc mode
			250	—	—	ns	HS osc mode
			250	—	—	ns	XT osc mode
			5.0	—	—	μs	LP osc mode
		Oscillator Period <sup>(2)</sup>	250	—	—	ns	RC osc mode
			250	—	10,000	ns	HS osc mode
			250	—	10,000	ns	XT osc mode
			50	—	200	μs	LP osc mode
2	TCY	Instruction Cycle Time <sup>(3)</sup>	—	4/FOSC	—	—	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50*	—	—	ns	XT oscillator
			20*	—	—	ns	HS oscillator
			2.0*	—	—	μs	LP oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25*	ns	XT oscillator
			—	—	25*	ns	HS oscillator
			—	—	50*	ns	LP oscillator

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ”) column is at VREG = 5V, VDD = 9V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

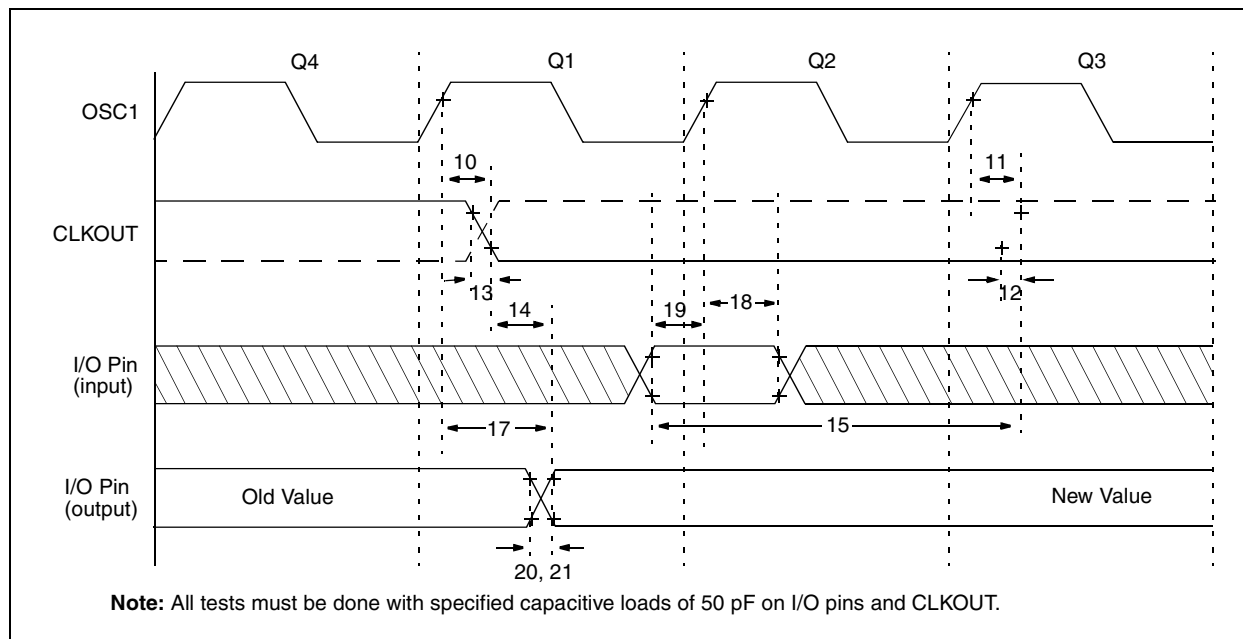
**2:** All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption.

When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

**3:** Instruction cycle period (TCY) equals four times the input oscillator time base period.

# PIC16HV540

**FIGURE 10-3: CLKOUT AND I/O TIMING - PIC16HV540**



**TABLE 10-2: CLKOUT AND I/O TIMING REQUIREMENTS - PIC16HV540**

Standard Operating Conditions (unless otherwise specified)						
AC Characteristics		Operating Temperature	0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial)			
Parameter No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units
10	TosH2ckL	OSC1↑ to CLKOUT↓ <sup>(2)</sup>	—	15	30**	ns
11	TosH2ckH	OSC1↑ to CLKOUT↑ <sup>(2)</sup>	—	15	30**	ns
12	TckR	CLKOUT rise time <sup>(2)</sup>	—	5.0	15**	ns
13	TckF	CLKOUT fall time <sup>(2)</sup>	—	5.0	15**	ns
14	TckL2ioV	CLKOUT↓ to Port out valid <sup>(2)</sup>	—	—	40**	ns
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid <sup>(3)</sup>	—	—	100*	ns
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns
20	TioR	Port output rise time <sup>(3)</sup>	—	10	25**	ns
21	TioF	Port output fall time <sup>(3)</sup>	—	10	25**	ns

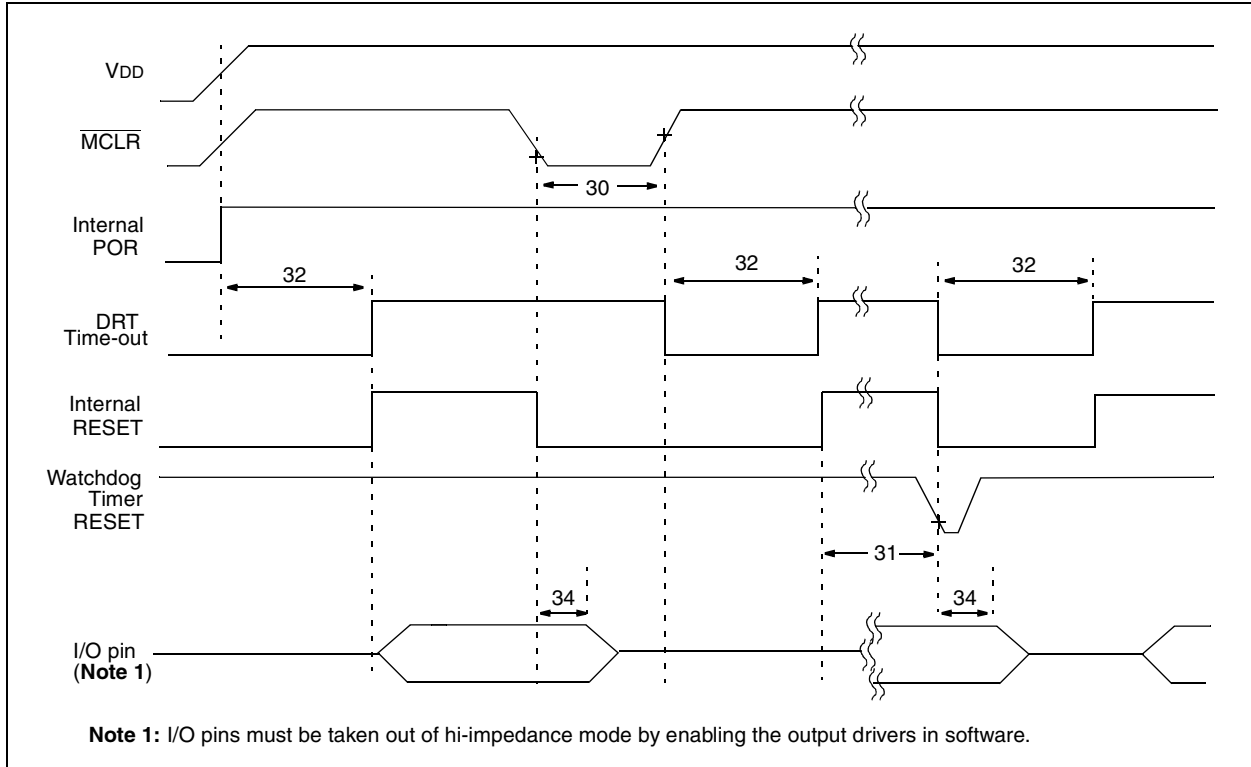
\*\* These parameters are design targets and are not tested. No characterization data available at this time.

**Note 1:** Data in the Typical ("Typ") column is at VREG = 5V, VDD = 9V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

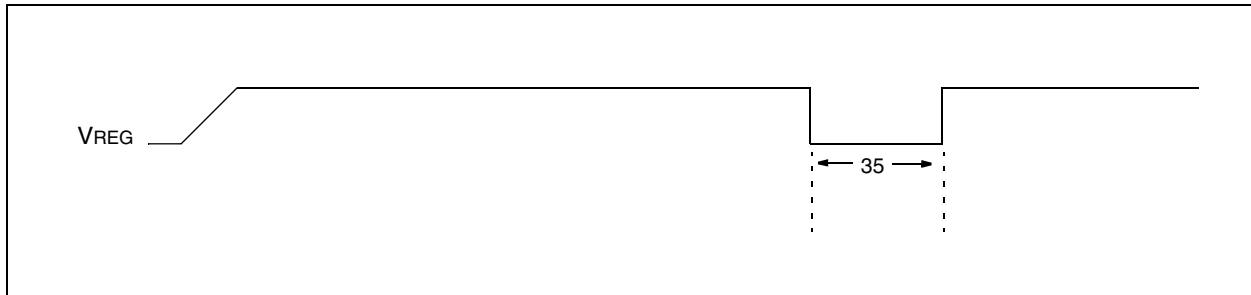
**2:** Measurements are taken in RC Mode where CLKOUT output is 8 x Tosc.

**3:** See Figure 10-1 for loading conditions.

**FIGURE 10-4: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING - PIC16HV540**



**FIGURE 10-5: BROWN-OUT DETECT TIMING**



# PIC16HV540

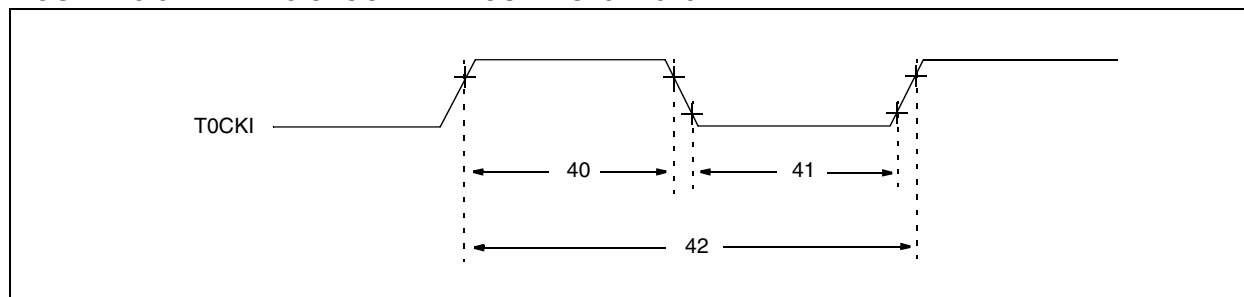
**TABLE 10-3: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER - PIC16HV540**

AC Characteristics		Standard Operating Conditions (unless otherwise specified)					
		Operating Temperature		0°C ≤ TA ≤ +70°C (commercial)			
				-40°C ≤ TA ≤ +85°C (industrial)			
Parameter No.	Sym	Characteristic	Min.	Typ. <sup>(1)</sup>	Max.	Units	Conditions
30	Tmcl	MCLR Pulse Width (low)	2	—	—	μs	VDD = 15V, VREG = 5V
31	Twdt	Watchdog Timer Time-out Period	9.0*	18*	40*	ms	VDD = 15V, VREG = 5V
32	TDRt	Device Reset Timer Period	9.0* 0.55*	18* 1.1*	30* 2.5*	ms	VDD = 15V, VREG = 5V, RC mode
34	Tioz	I/O Hi-impedance from MCLR Low	—	—	100*	ns	
—	Tpc	Pin Change Pulse Width	2	—	—	μs	
35	TBOD	Brown-out Detect Pulse Width	—	2	—	μs	VREG ≤ BVDD

\* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ”) column is at VREG = 5V, VDD = 15V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 10-6: TIMER0 CLOCK TIMINGS - PIC16HV540**



**TABLE 10-4: TIMER0 CLOCK REQUIREMENTS - PIC16HV540**

AC Characteristics		Standard Operating Conditions (unless otherwise specified)					
		Operating Temperature		0°C ≤ TA ≤ +70°C (commercial)			
				-40°C ≤ TA ≤ +85°C (industrial)			
Parameter No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width - No Prescaler	0.5 TCY + 20*	—	—	ns	
		- With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width - No Prescaler	0.5 TCY + 20*	—	—	ns	
		- With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period	20 or $\frac{TCY + 40}{N}$ *	—	—	ns	Whichever is greater. N = Prescale Value (1, 2, 4, ..., 256)

\* These parameters are characterized but not tested.

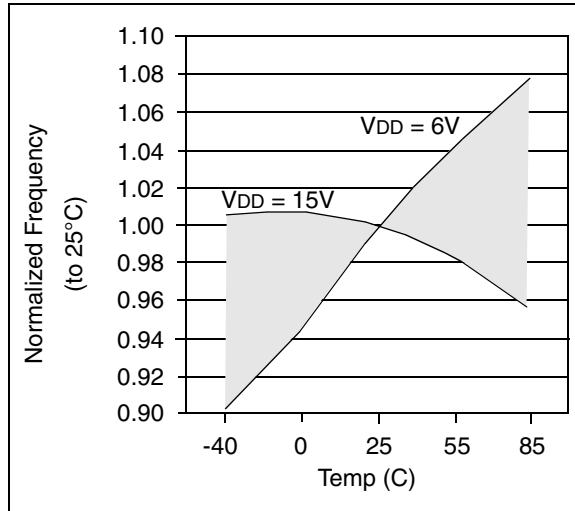
**Note 1:** Data in the Typical (“Typ”) column is at 3.8V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

## 11.0 DC AND AC CHARACTERISTICS - PIC16HV540

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g., outside specified VDD range). This is for information only and devices will operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively, where σ is standard deviation.

**FIGURE 11-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE**

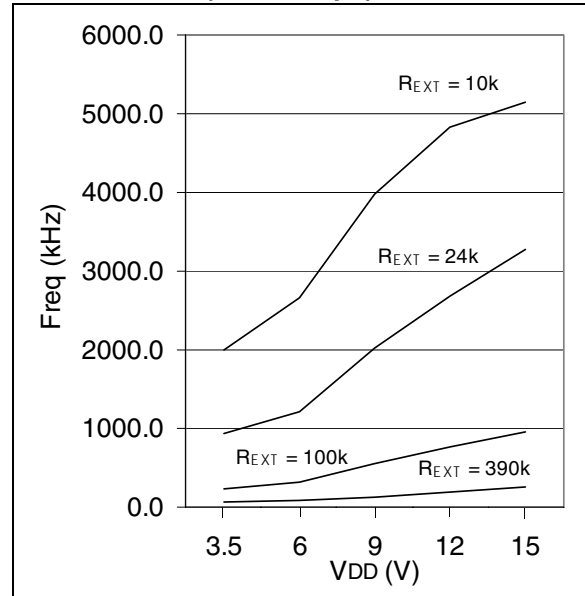


**TABLE 11-1: RC OSCILLATOR FREQUENCIES**

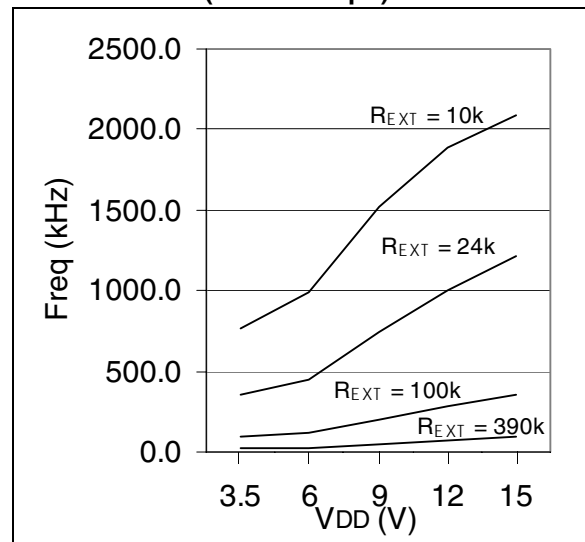
C <sub>EXT</sub>	R <sub>EXT</sub>	Average Fosc, V <sub>IO</sub> = 5V	
		25°C, V <sub>DD</sub> = 6V	25°C, V <sub>DD</sub> = 15V
20 pF	3.3k	4986.7 kHz	(1)
	5k	4233.3 kHz	(1)
	10k	2656.7 kHz	5150.0 kHz
	24k	1223.3 kHz	3286.7 kHz
	100k	325.7 kHz	955.7 kHz
100 pF	390k	79.0 kHz	250.7 kHz
	3.3k	1916.7 kHz	(1)
	5k	1593.3 kHz	(1)
	10k	995.7 kHz	2086.7 kHz
	24k	448.3 kHz	1210.0 kHz
300 pF	100k	116.0 kHz	355.7 kHz
	390k	28.3 kHz	89.7 kHz
	3.3k	744 kHz	(1)
	5k	620.3 kHz	(1)
	10k	382.0 kHz	817.3 kHz
	24k	169.7 kHz	483.0 kHz
	100k	44.1 kHz	135.7 kHz
	390k	10.6 kHz	34.4 kHz

**Note 1:** This combination of R, C and VDD draws too much current and prohibits oscillator operation.

**FIGURE 11-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD (C<sub>EXT</sub> = 20pF)**

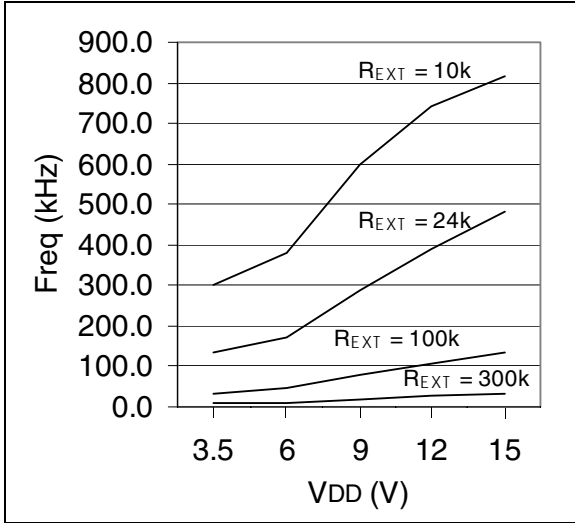


**FIGURE 11-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD (C<sub>EXT</sub> = 100pF)**

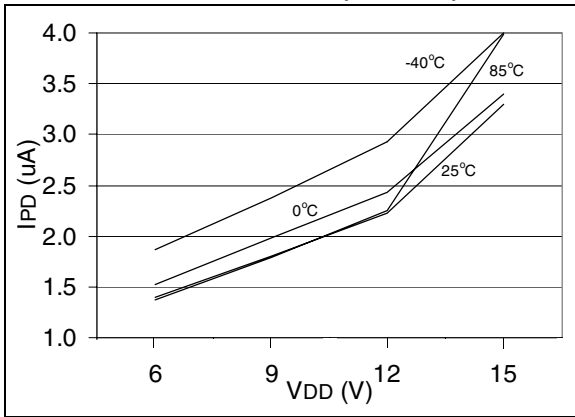


# PIC16HV540

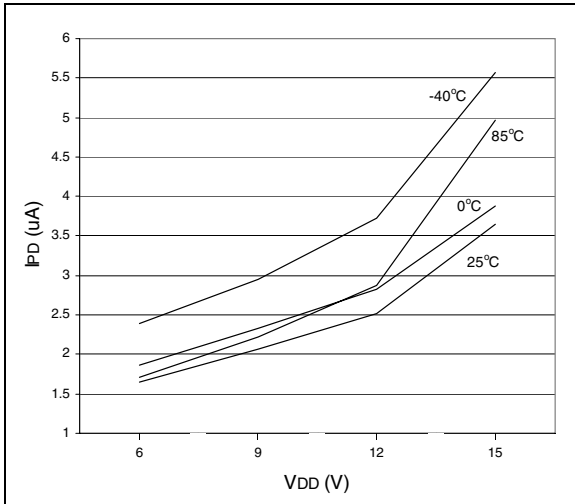
**FIGURE 11-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD (C<sub>EXT</sub> = 300pF)**



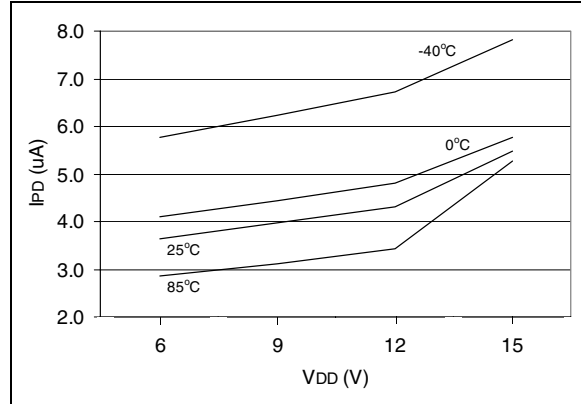
**FIGURE 11-5: TYPICAL I<sub>PD</sub> vs. VDD, WATCHDOG TIMER DISABLED (V<sub>IO</sub> = 5V)**



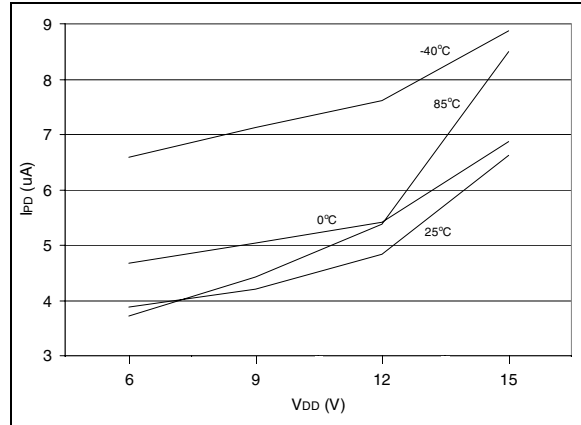
**FIGURE 11-6: MAXIMUM I<sub>PD</sub> vs. VDD, WATCHDOG TIMER DISABLED (V<sub>IO</sub> = 5V)**



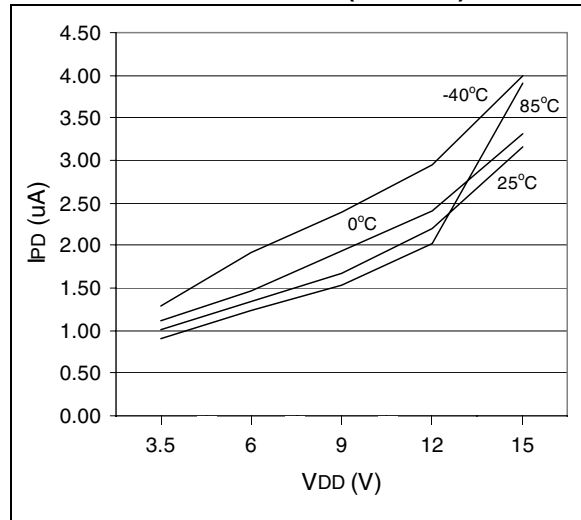
**FIGURE 11-7: TYPICAL I<sub>PD</sub> vs. VDD, WATCHDOG TIMER ENABLED (V<sub>IO</sub> = 5V)**



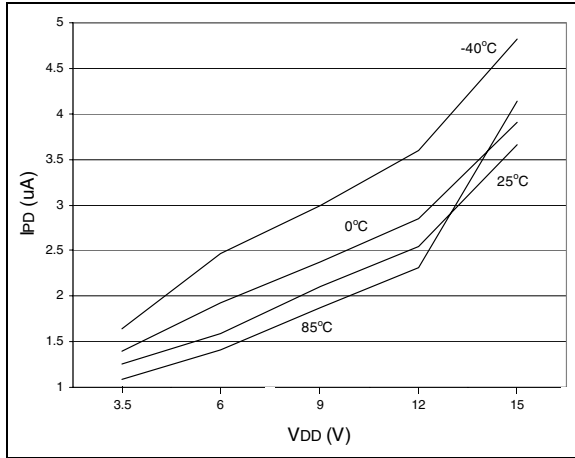
**FIGURE 11-8: MAXIMUM I<sub>PD</sub> vs. VDD, WATCHDOG TIMER ENABLED (V<sub>IO</sub> = 5V)**



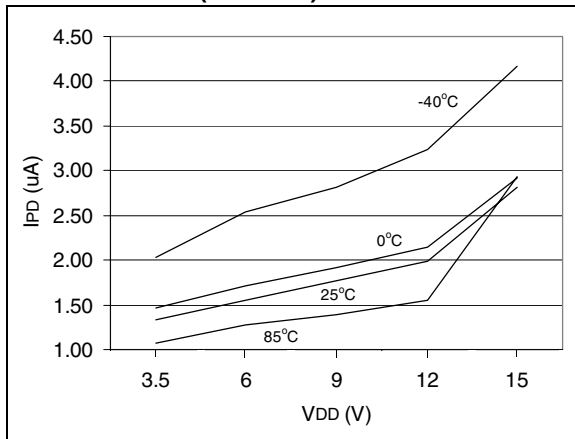
**FIGURE 11-9: TYPICAL I<sub>PD</sub> vs. VDD, WATCHDOG TIMER DISABLED (V<sub>IO</sub> = 3V)**



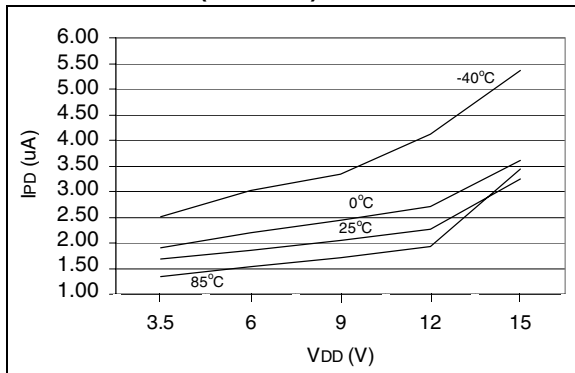
**FIGURE 11-10: MAXIMUM IPD vs. VDD, WATCHDOG TIMER DISABLED (VIO = 3V)**



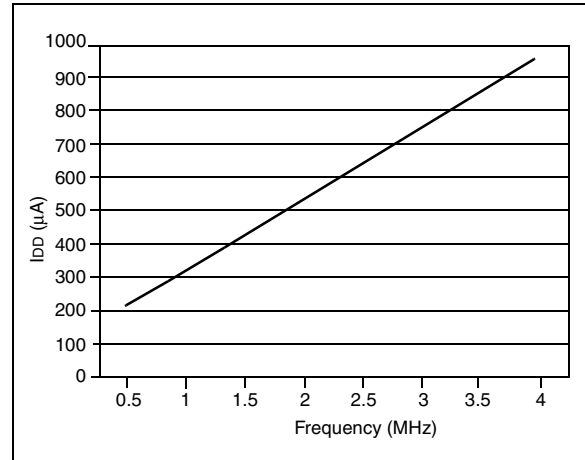
**FIGURE 11-11: TYPICAL IPD vs. VDD, WATCHDOG TIMER ENABLED (VIO = 3V)**



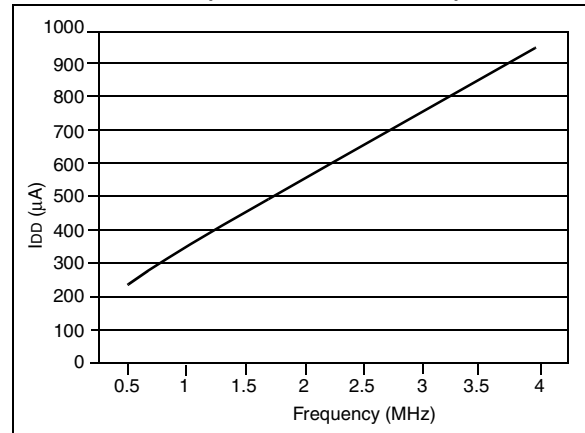
**FIGURE 11-12: MAXIMUM IPD vs. VDD, WATCHDOG TIMER ENABLED (VIO = 3V)**



**FIGURE 11-13: MAXIMUM IDD vs. FREQUENCY, WATCHDOG TIMER DISABLED, RC MODE (VDD = 15V, VIO = 5V, -40°C TO +85°C)**

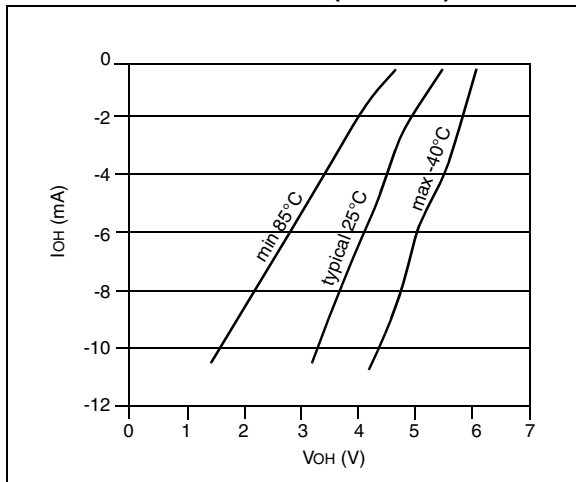


**FIGURE 11-14: MAXIMUM IDD vs. FREQUENCY, WATCHDOG TIMER ENABLED, RC MODE (VDD = 15V, VIO = 5V)**



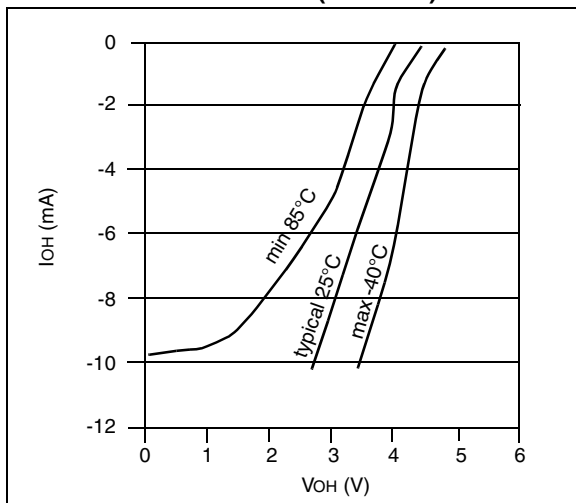
# PIC16HV540

**FIGURE 11-15:  $I_{OH}$  vs.  $V_{OH}$  ON PORTA,  
 $V_{DD} = 15V$  ( $V_{IO} = 5V$ )**



**Note:** Current being applied is being applied simultaneously to all 4 PORTA pins.

**FIGURE 11-16:  $I_{OH}$  vs.  $V_{OH}$  ON PORTA,  
 $V_{DD} = 5V$  ( $V_{IO} = 5V$ )**

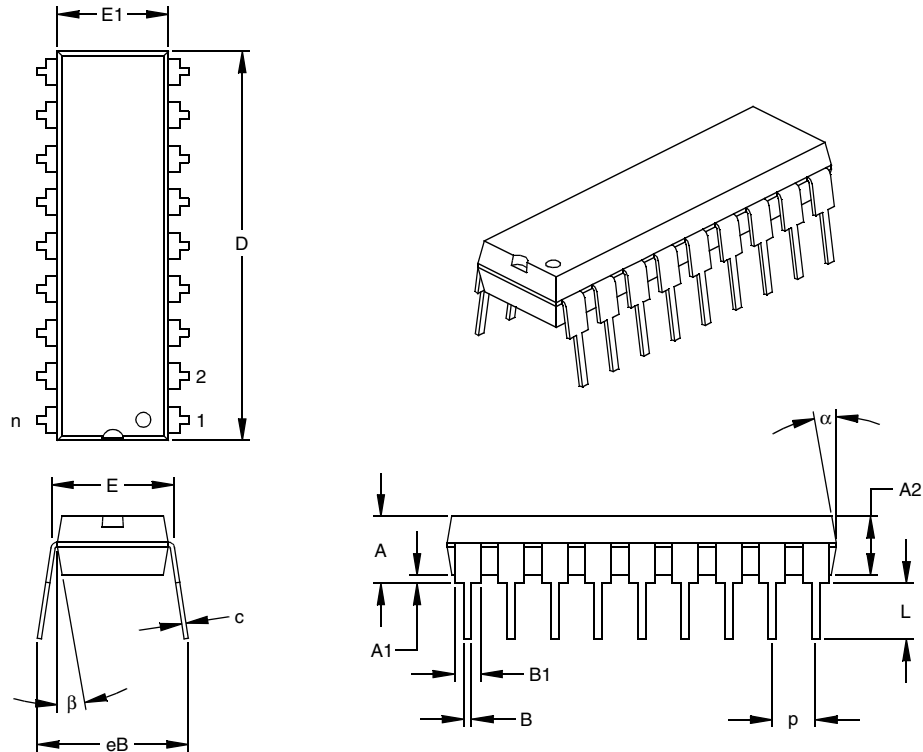


**Note:** Current being applied is being applied simultaneously to all 4 PORTA pins.



## 12.0 PACKAGING INFORMATION

### 12.1 18-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

\* Controlling Parameter  
 § Significant Characteristic

Notes:

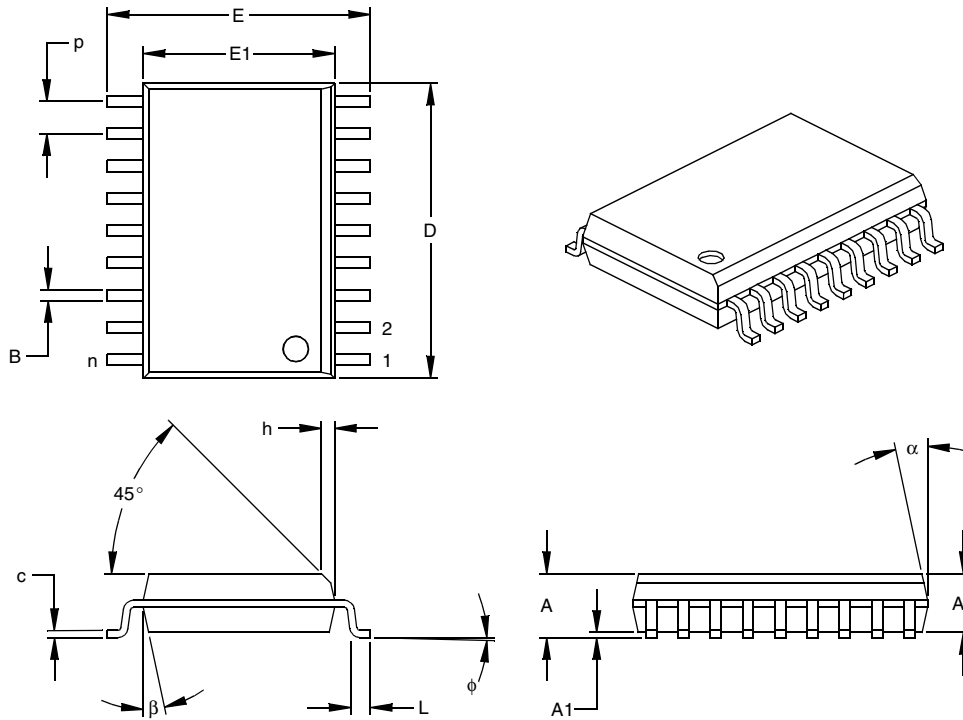
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-007

# PIC16HV540

## 12.2 18-Lead Plastic Small Outline (SO) – Wide, 300 mil (SOIC)



Dimension Limits	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

\* Controlling Parameter

§ Significant Characteristic

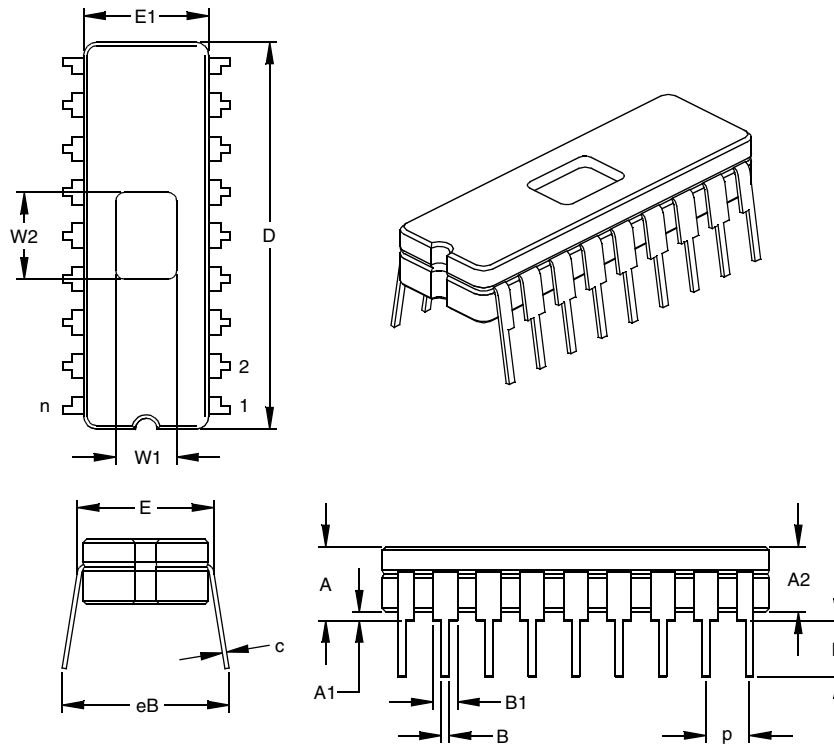
### Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-051

## 12.3 18-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)

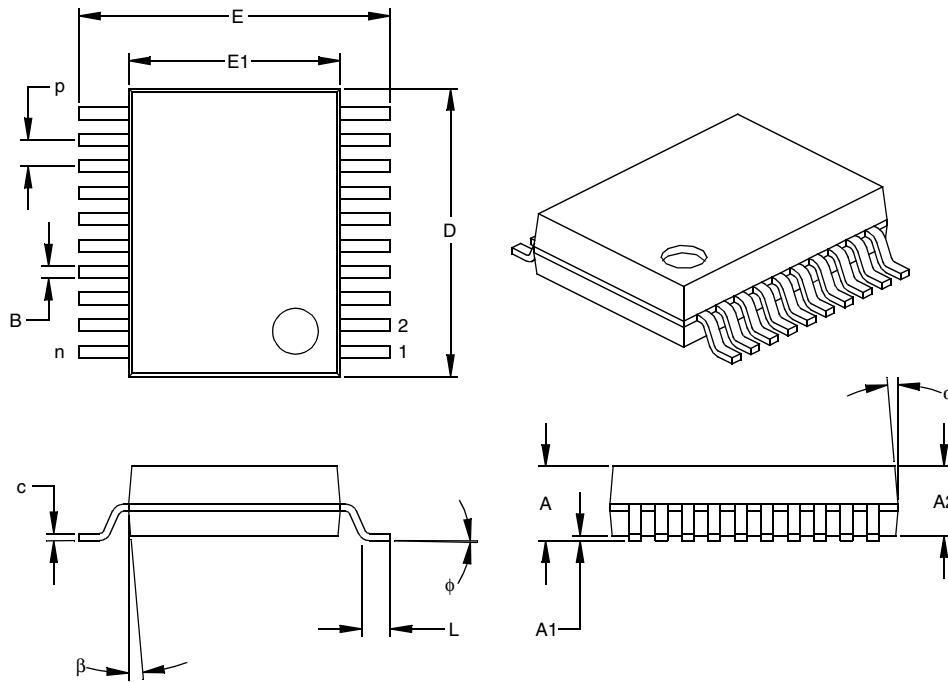


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.170	.183	.195	4.32	4.64	4.95
Ceramic Package Height	A2	.155	.160	.165	3.94	4.06	4.19
Standoff	A1	.015	.023	.030	0.38	0.57	0.76
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Ceramic Pkg. Width	E1	.285	.290	.295	7.24	7.37	7.49
Overall Length	D	.880	.900	.920	22.35	22.86	23.37
Tip to Seating Plane	L	.125	.138	.150	3.18	3.49	3.81
Lead Thickness	c	.008	.010	.012	0.20	0.25	0.30
Upper Lead Width	B1	.050	.055	.060	1.27	1.40	1.52
Lower Lead Width	B	.016	.019	.021	0.41	0.47	0.53
Overall Row Spacing	§ eB	.345	.385	.425	8.76	9.78	10.80
Window Width	W1	.130	.140	.150	3.30	3.56	3.81
Window Length	W2	.190	.200	.210	4.83	5.08	5.33

\* Controlling Parameter  
 § Significant Characteristic  
 JEDEC Equivalent: MO-036  
 Drawing No. C04-010

# PIC16HV540

## 12.4 20-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)



Dimension Limits	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.322	7.59	7.85	8.18
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.278	.284	.289	7.06	7.20	7.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

\* Controlling Parameter

§ Significant Characteristic

**Notes:**

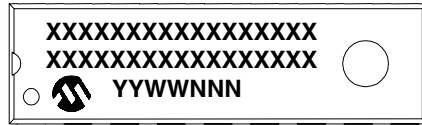
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-150

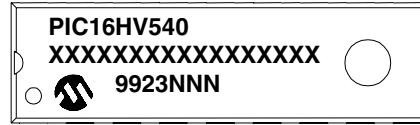
Drawing No. C04-072

## 12.5 Package Marking Information

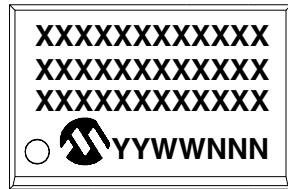
18-Lead PDIP



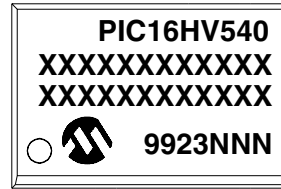
Example



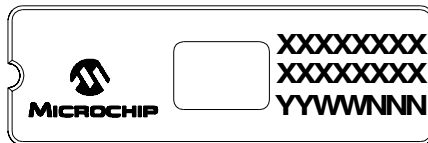
18-Lead SOIC



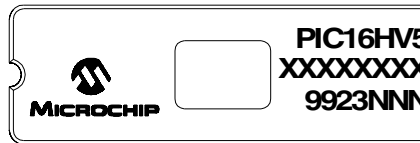
Example



18-Lead CERDIP Windowed



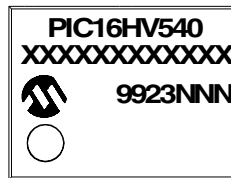
Example



20-Lead SSOP



Example



**Legend:** MM...M Microchip part number information  
 XX...X Customer specific information\*  
 YY Year code (last 2 digits of calendar year)  
 WW Week code (week of January 1 is week '01')  
 NNN Alphanumeric traceability code

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

\* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

# PIC16HV540

---

NOTES:

## INDEX

### A

Absolute Maximum Ratings .....	61
ALU .....	7
Applications .....	3
Architectural Overview .....	7
Assembler	
MPASM Assembler .....	55

### B

Block Diagram	
On-Chip Reset Circuit .....	35
PIC16C5X Series .....	8
Timer0 .....	25
TMR0/WDT Prescaler .....	29
Watchdog Timer .....	40
Brown-out Detect .....	31

### C

Carry bit .....	7
Clocking Scheme .....	10
Code Protection .....	31, 42
Configuration Bits .....	31
Configuration Word .....	31
PIC16CR54C .....	31

### D

DC and AC Characteristics - PIC16CR54C .....	69
DC Characteristics .....	62
Development Support .....	55
Device Varieties .....	5
Digit Carry bit .....	7

### E

Electrical Characteristics	
PIC16CR54C .....	61
Enhanced Watchdog Timer (WDT) .....	31
Errata .....	2
External Power-On Reset Circuit .....	36

### F

Family of Devices	
PIC16C5X .....	4
Features .....	1
FSR .....	35
FSR Register .....	17

### I

I/O Interfacing .....	19
I/O Ports .....	19
I/O Programming Considerations .....	22
INDF .....	35
INDF Register .....	17
Indirect Data Addressing .....	17
Instruction Cycle .....	10
Instruction Flow/Pipelining .....	10
Instruction Set Summary .....	43

### K

KeeLoq® Evaluation and Programming Tools .....	58
--	----

### L

Load Conditions .....	64
Loading of PC .....	16

### M

MCLR .....	35
Memory Map .....	11
PIC16C54s/CR54s/C55s .....	11
Memory Organization .....	11
Data Memory .....	11
Program Memory .....	11
MPLAB Integrated Development Environment Software ..	55

### O

One-Time-Programmable (OTP) Devices .....	5
OPTION Register .....	14
OSC selection .....	31
Oscillator Configurations .....	32
Oscillator Types	
HS .....	32
LP .....	32
RC .....	32
XT .....	32

### P

Package Marking Information .....	77
Packaging Information .....	73
PC .....	16, 35
PICDEM-1 Low-Cost PICmicro Demo Board .....	57
PICDEM-2 Low-Cost PIC16CXX Demo Board .....	57
PICDEM-3 Low-Cost PIC16CXXX Demo Board .....	57
PICSTART® Plus Entry Level Development System .....	57
pin diagrams .....	1
POR	

Device Reset Timer (DRT) .....	31, 38
$\overline{PD}$ .....	34, 41
Power-On Reset (POR) .....	31, 35, 36
$\overline{TO}$ .....	34, 41

PORTA .....	19, 35
PORTB .....	19, 35
Power-Down Mode .....	41
Prescaler .....	28
PRO MATE® II Universal Programmer .....	57
Program Counter .....	16

### Q

Q cycles .....	10
Quick-Turnaround-Production (QTP) Devices .....	5

### R

RC Oscillator .....	33
Read-Modify-Write .....	22
Register File Map .....	11
Registers	
Special Function .....	11
Reset .....	31, 34

### S

SEEVAL® Evaluation and Programming System .....	58
Serialized Quick-Turnaround-Production (SQTP) Devices ..	5
SLEEP .....	31, 41
Software Simulator (MPLAB-SIM) .....	56
Special Features of the CPU .....	31
Special Function Registers .....	11
Stack .....	16
STATUS .....	35
STATUS Register .....	7, 13

### T

Timer0	
Switching Prescaler Assignment .....	28
Timer0 (TMR0) Module .....	25
TMR0 with External Clock .....	27
Timing Diagrams and Specifications .....	65
Timing Parameter Symbolology and Load Conditions .....	64
TRIS Registers .....	19

### U

UV Erasable Devices .....	5
---------------------------	---

### W

W .....	35
Wake-up from SLEEP .....	41
Wake-up from SLEEP on Pin Change .....	31
Watchdog Timer (WDT) .....	39

# PIC16HV540

---

Period .....	39
Programming Considerations .....	39
WWW, On-Line Support .....	2
<b>Z</b>	
Zero bit .....	7



## ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

### Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

**[www.microchip.com](http://www.microchip.com)**

The file transfer site is available by using an FTP service to connect to:

**<ftp://ftp.microchip.com>**

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

## Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and  
1-480-786-7302 for the rest of the world.

991103

Trademarks: The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER, PRO MATE and MPLAB are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. *FlexROM* and *fuzzyLAB* are trademarks and SQTP is a service mark of Microchip in the U.S.A.

All other trademarks mentioned herein are the property of their respective companies.

# PIC16HV540

---

---

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent  
RE: Reader Response  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC16HV540

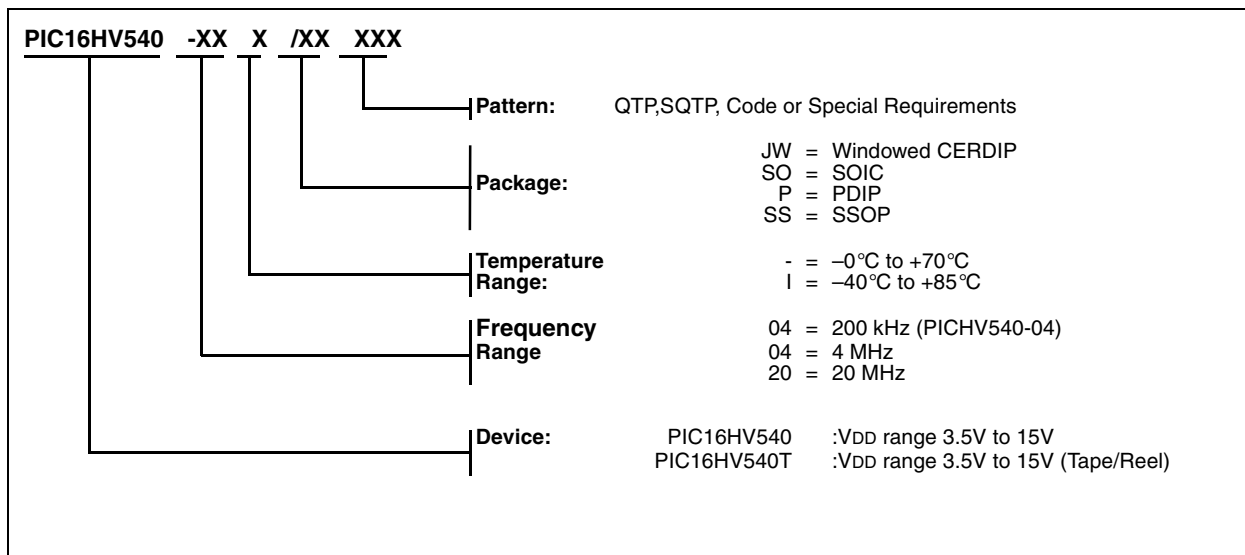
Literature Number: DS40197B

Questions:

1. What are the best features of this document?  
\_\_\_\_\_  
\_\_\_\_\_
2. How does this document meet your hardware and software development needs?  
\_\_\_\_\_  
\_\_\_\_\_
3. Do you find the organization of this data sheet easy to follow? If not, why?  
\_\_\_\_\_  
\_\_\_\_\_
4. What additions to the data sheet do you think would enhance the structure and subject?  
\_\_\_\_\_  
\_\_\_\_\_
5. What deletions from the data sheet could be made without affecting the overall usefulness?  
\_\_\_\_\_  
\_\_\_\_\_
6. Is there any incorrect or misleading information (what and where)?  
\_\_\_\_\_  
\_\_\_\_\_
7. How would you improve this document?  
\_\_\_\_\_  
\_\_\_\_\_
8. How would you improve our software, systems, and silicon products?  
\_\_\_\_\_  
\_\_\_\_\_

## PIC16HV540 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



### Sales and Support

#### Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277.
3. The Microchip Worldwide Site ([www.microchip.com](http://www.microchip.com))

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

#### New Customer Notification System

Register on our web site ([www.microchip.com/cn](http://www.microchip.com/cn)) to receive the most current information on our products.

---

---

**Note the following details of the code protection feature on PICmicro® MCUs.**

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

---

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.*



# MICROCHIP

## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200 Fax: 480-792-7277  
Technical Support: 480-792-7627  
Web Address: <http://www.microchip.com>

#### Rocky Mountain

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7966 Fax: 480-792-7456

#### Atlanta

500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770-640-0034 Fax: 770-640-0307

#### Boston

2 Lan Drive, Suite 120  
Westford, MA 01886  
Tel: 978-692-3848 Fax: 978-692-3821

#### Chicago

333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071 Fax: 630-285-0075

#### Dallas

4570 Westgrove Drive, Suite 160  
Addison, TX 75001  
Tel: 972-818-7423 Fax: 972-818-2924

#### Detroit

Tri-Atria Office Building  
32255 Northwestern Highway, Suite 190  
Farmington Hills, MI 48334  
Tel: 248-538-2250 Fax: 248-538-2260

#### Kokomo

2767 S. Albright Road  
Kokomo, Indiana 46902  
Tel: 765-864-8360 Fax: 765-864-8387

#### Los Angeles

18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888 Fax: 949-263-1338

#### New York

150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 631-273-5305 Fax: 631-273-5335

#### San Jose

Microchip Technology Inc.  
2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408-436-7950 Fax: 408-436-7955

#### Toronto

6285 Northam Drive, Suite 108  
Mississauga, Ontario L4V 1X5, Canada  
Tel: 905-673-0699 Fax: 905-673-6509

### ASIA/PACIFIC

#### Australia

Microchip Technology Australia Pty Ltd  
Suite 22, 41 Rawson Street  
Epping 2121, NSW  
Australia  
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

#### China - Beijing

Microchip Technology Consulting (Shanghai)  
Co., Ltd., Beijing Liaison Office  
Unit 915  
Bei Hai Wan Tai Bldg.  
No. 6 Chaoyangmen Beidajie  
Beijing, 100027, No. China  
Tel: 86-10-85282100 Fax: 86-10-85282104

#### China - Chengdu

Microchip Technology Consulting (Shanghai)  
Co., Ltd., Chengdu Liaison Office  
Rm. 2401, 24th Floor,  
Ming Xing Financial Tower  
No. 88 TIDU Street  
Chengdu 610016, China  
Tel: 86-28-6766200 Fax: 86-28-6766599

#### China - Fuzhou

Microchip Technology Consulting (Shanghai)  
Co., Ltd., Fuzhou Liaison Office  
Unit 28F, World Trade Plaza  
No. 71 Wusi Road  
Fuzhou 350001, China  
Tel: 86-591-7503506 Fax: 86-591-7503521

#### China - Shanghai

Microchip Technology Consulting (Shanghai)  
Co., Ltd.  
Room 701, Bldg. B  
Far East International Plaza  
No. 317 Xian Xia Road  
Shanghai, 200051  
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

#### China - Shenzhen

Microchip Technology Consulting (Shanghai)  
Co., Ltd., Shenzhen Liaison Office  
Rm. 1315, 13/F, Shenzhen Kerry Centre,  
Renminnan Lu  
Shenzhen 518001, China  
Tel: 86-755-2350361 Fax: 86-755-2366086

#### Hong Kong

Microchip Technology Hongkong Ltd.  
Unit 901-6, Tower 2, Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2401-1200 Fax: 852-2401-3431

#### India

Microchip Technology Inc.  
India Liaison Office  
Divyasree Chambers  
1 Floor, Wing A (A3/A4)  
No. 11, O'Shaugnessey Road  
Bangalore, 560 025, India  
Tel: 91-80-2290061 Fax: 91-80-2290062

### Japan

Microchip Technology Japan K.K.  
Benex S-1 6F  
3-18-20, Shinyokohama  
Kohoku-Ku, Yokohama-shi  
Kanagawa, 222-0033, Japan  
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

### Korea

Microchip Technology Korea  
168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-Ku  
Seoul, Korea 135-882  
Tel: 82-2-554-7200 Fax: 82-2-558-5934

### Singapore

Microchip Technology Singapore Pte Ltd.  
200 Middle Road  
#07-02 Prime Centre  
Singapore, 188980  
Tel: 65-6334-8870 Fax: 65-6334-8850

### Taiwan

Microchip Technology Taiwan  
11F-3, No. 207  
Tung Hua North Road  
Taipei, 105, Taiwan  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### Denmark

Microchip Technology Nordic ApS  
Regus Business Centre  
Lautrup høj 1-3  
Ballerup DK-2750 Denmark  
Tel: 45 4420 9895 Fax: 45 4420 9910

#### France

Microchip Technology SARL  
Parc d'Activite du Moulin de Massy  
43 Rue du Saule Trapu  
Batiment A - 1er Etage  
91300 Massy, France  
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

#### Germany

Microchip Technology GmbH  
Gustav-Heinemann Ring 125  
D-81739 Munich, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

#### Italy

Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-039-65791-1 Fax: 39-039-6899883

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Winnersh Triangle  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44 118 921 5869 Fax: 44-118 921-5820

03/01/02