Section 27. Device Configuration Bits

HIGHLIGHTS

This section of the manual contains the following major topics:

27.1 Introduction

The device configuration bits allow each user to customize certain aspects of the device to the needs of the application. When the device powers up, the state of these bits determines the modes that the device uses. Subsection **27.2 "Configuration Word Bits"** discusses the configuration bits, and the modes that they can be configured to. These bits are mapped in program memory location 2007h. This location is not accessible during normal device operation (can be accessed only during programming mode).

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. The ability to change these settings once they have been programmed depends on the memory technology and the package type.

For Read Only Memory (ROM) devices, these bits are specified at time of ROM code submittal and once the device is masked may not be changed for those devices (would require a new mask code).

For One Time Programmable (OTP) devices, once these bits are programmed ('0'), they may not be changed.

For windowed EPROM devices, once these bits are programmed ('0'), the device must be UV erased to return the configuration word to the erased state. UV erasing the device also erases the program memory.

For Flash devices, these bits may be erased and reprogrammed.

Note:	Microchip does not recommend code protecting windowed devices.

Section 27. Device Configuration Bits

Section **27.2** is forced to the next page for formatting purposes.

27.2 Configuration Word Bits

These configuration bits specify some of the modes of the device, and are programmed by a device programmer, or by using the In-Circuit Serial Programming (ICSP) feature of the midrange devices. The device is not able to read the values of these bits, and there placement is automatically taken care of when you select the device in you device programmer. For additional information, please refer to the Programming Specification of the Device.

Note 1: Always ensure that your device programmer has the same device selected as you are programming.

Note 2: Microchip recommends that the desired configuration bit states be embedded in to the application source code. This is easily done in the MPASM assembler by the use of the CONFIG directive. See Subsection **27.2.1 "MPASM's CONFIG Directive."**

CP1:CP0: Code Protection bits

11 = Code protection off
10 = See device data sheet
01 = See device data sheet
00 = All memory is code protected

Note: Some devices may use more or less bits to determine the code protect. Presently there are also some devices that use only one bit (CP0). For these devices the bit description is:
1 = Code protection off
0 = Code protection on

DP: Data EEPROM Memory Code Protection bit

1 = Code protection off
0 = Data EEPROM Memory is code protected

Note: This bit is used when a device with ROM program memory also has Data EEPROM memory.

BODEN: Brown-out Reset Enable bit

1 = BOR enabled
0 = BOR disabled

Note: Enabling Brown-out Reset automatically enables the Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.

$\overline{\text{PWRTE}}$: Power-up Timer Enable bit

1 = PWRT disabled
0 = PWRT enabled

Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.

Note 2: Some original PICmicros have the polarity of this bit reversed.

Note 3:

MCLRE: $\overline{\text{MCLR}}$ Pin Function Select bit

1 = Pin's function is $\overline{\text{MCLR}}$

0 = Pin's function is as a digital I/O. $\overline{\text{MCLR}}$ is internally tied to V_{DD}.

WDTE: Watchdog Timer Enable bit

1 = WDT enabled
0 = WDT disabled

FOSC1:FOSC0: Oscillator Selection bits

11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

FOSC2:FOSC0: Oscillator Selection bits

111 = EXTRC oscillator, with CLKOUT
110 = EXTRC oscillator
101 = INTRC oscillator, with CLKOUT
100 = INTRC oscillator
011 = Reserved
010 = HS oscillator
001 = XT oscillator
000 = LP oscillator

Unimplemented: Read as '1'

Legend		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
- n = Value at POR reset		u = Unchanged from programmed state

Note: The bit position of the configuration bits is device dependent. Please refer to the device programming specification for bit placement. The use of a Microchip device programmer does not require you to know the bit locations.

27.2.1 MPASM's CONFIG Directive

Microchip's assembler, MPASM, has a nice feature that allows you to specify, in the source code file, the selected states of the configuration bits for this program. This ensures that when programming a device for an application the required configuration is also programmed. This minimizes the risk of programming the wrong device configuration, and wondering why it no longer works in the application.

Example 27-1 show a template for using the CONFIG directive.

Example 27-1:Using the CONFIG Directive, a Source File Template

```
     LIST   p = p16C77      ; List Directive,
;    Revision History
;
     #INCLUDE    <P16C77.INC>    ; Microchip Device Header File
;
     #INCLUDE    <MY_STD.MAC>    ; File which includes my standard macros
     #INCLUDE    <APP.MAC>       ; File which includes macros specific
                                 ;   to this application
;
; Specify Device Configuration Bits
;
     __CONFIG    _XT_OSC & _PWRTE_ON & _BODEN_OFF & _CP_OFF & _WDT_ON
;
     org   0x00            ; Start of Program Memory
RESET_ADDR   :             ; First instruction to execute after a reset


     end
```

The Symbols that are currently in the Microchip Device Header files that make using the CONFIG directive straight forward are shown in Table 27-1. For the symbols available for your device, please refer to that device's Microchip Include file.

Note:	As long as the correct device is specified (in the LIST and INCLUDE file directives), the correct polarity of all bits is ensured.

Table 27-1: __CONFIG Directive Symbols (From Microchip Header Files)

Feature	SYMBOLS
Oscillators	_RC_OSC
	_EXTRC_OSC
	_EXTRC_OSC_CLKOUT
	_EXTRC_OSC_NOCLKOUT
	_INTRC_OSC
	_INTRC_OSC_CLKOUT
	_INTRC_OSC_NOCLKOUT
	_LP_OSC
	_XT_OSC
	_HS_OSC
Watch Dog Timer	_WDT_ON
	_WDT_OFF
Power-up Timer	_PWRTE_ON
	_PWRTE_OFF
Brown-out Reset	_BODEN_ON
	_BODEN_OFF
Master Clear Enable	_MCLRE_ON
	_MCLRE_OFF
Code Protect	_CP_ALL
	_CP_ON
	_CP_75
	_CP_50
	_CP_OFF
Code Protect Data EEPROM	_DP_ON
	_DP_OFF
Code Protect Calibration Space	_CPC_ON
	_CPC_OFF

Note 1: Not all configuration bit symbols may be available on any one device. Please refer to the MIcrochip include file of that device for available symbols.

27.3 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.

27.3.1 ROM Devices

When a ROM device also has Data EEPROM memory, an additional code protect configuration bit may be implemented. The program memory configuration bit is submitted as part of the ROM code submittal. The Data EEPROM memory code protect configuration bit will be an EEPROM bit. When ROM devices complete testing, the EEPROM data memory code protect bit will be programmed to the same state as the program memory code protect bit. That is data EEPROM code protect is off, when program memory code protect is off, and data EEPROM code protect is on for all other selections.

In applications where the device is code protected and the data EEPROM needs to be programmed before the application can be released, the data EEPROM memory must have the entire data EEPROM memory erased. The device programming specification details the steps to do this. Microchip device programmers implement the specified sequence. Once this sequence is complete, the Data EEPROM memory code protect is disabled. This allows the desired data to be programmed into the device. After programming the data EEPROM memory array, the data EEPROM memory code protect configuration bit should be programmed as desired.

27.4 ID Locations

Four memory locations (2000h - 2003h) 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. It is recommended that only the 4 least significant bits of the ID location are used.

27.5 Design Tips

Question 1: ***I have a JW device and I can no longer program it (reads scrambled data or all '0's). What's wrong with the device?***

Answer 1:

Nothing. You probably code protected the device. If this is the case, the device is no longer usable. See Subsection **27.3 "Program Verification/Code Protection"** for more details.

Question 2: ***In converting from a PIC16C74 to a PIC16C74A, my application no longer works.***

Answer 2:

1. Did you re-assemble the source file specifying the PIC16C74A in the INCLUDE file and LIST directives. The use of the CONFIG directive is highly recommended.
2. On the device programmer, did you specify the PIC16C74A, and were all the configuration bits as desired?

Question 3: ***When I erase the device, the program memory is blank but the configuration word is not yet erased.***

Answer 3:

That is by design. Also remember that Microchip does not recommend code protecting windowed devices.

27.6 Related Application Notes

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to Configuration Word are:

Title	Application Note #
No related Application Notes at this time.	

27.7 Revision History

Revision A

This is the initial released revision of the Configuration Word description.