



TMS320C3x Emulator

Installation Guide

1993

Microprocessor Development Systems



Printed in U.S.A., June 1993
2617675-9741 revision *

TMS320C3x Emulator Installation Guide

SPRU122A
June 1993



IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Installing the Emulator and C Source Debugger With DOS

This chapter helps you install the 'C3x emulator board and the C source debugger on a PC running MS-DOS or PC-DOS. You can also use the debugger with MS-Windows. When you complete the installation, turn to the *TMS320C3x C Source Debugger User's Guide*.

Topic	Page
1 What You'll Need	2
Hardware checklist	2
Software checklist	3
2 Step 1: Installing the Emulator Board in Your PC	4
Preparing the emulator board for installation	4
Setting the emulator board into your PC	6
3 Step 2: Connecting the Emulator to Your Target System	8
4 Step 3: Installing the Debugger Software	9
5 Step 4: Setting Up the Debugger Environment	9
Invoking the new or modified batch file	10
Modifying the PATH statement	11
Setting up the environment variables	11
Identifying the correct I/O switches	12
Resetting the emulator	12
6 Step 5: Verifying the Installation	13
Installation error messages	14
7 Using the Debugger With MS-Windows	15

1. What You'll Need

The following checklists detail items that are shipped with the 'C3x C source debugger and emulator and additional items you'll need to use these tools.

Hardware checklist

- | | | |
|--------------------------|--|--|
| <input type="checkbox"/> | host | An IBM PC/AT or 100% compatible ISA/EISA-based PC with a hard-disk system and a 1.2-megabyte floppy-disk drive |
| <input type="checkbox"/> | memory | Minimum of 4 megabytes |
| <input type="checkbox"/> | display | Monochrome or color (color recommended) |
| <input type="checkbox"/> | slot | One 16-bit slot |
| <input type="checkbox"/> | emulator board power requirements | Approximately 1 ampere @ 5 volts (5 watts) |
| <input type="checkbox"/> | target system | A board with a 'C3x |
| <input type="checkbox"/> | connector to target system | 12-pin connector (two rows of six pins)—see the <i>Specifications for Your Target System's Connection to the Emulator</i> appendix in the <i>TMS320C3x C Source Debugger User's Guide</i> for more information about this connector. |
| <input type="checkbox"/> | optional hardware | A Microsoft-compatible mouse |
| <input type="checkbox"/> | | An EGA- or VGA-compatible graphics display card and a large monitor. The debugger has two options that allow you to change the overall size of the debugger display. If you have an EGA- or VGA-compatible graphics card, you can take advantage of some of these larger screen sizes. These larger screen sizes are most effective when used with a large (17" or 19") monitor. (To use a larger screen size, you must invoke the debugger with an appropriate option. For more information about options, refer to the invocation instructions in the <i>TMS320C3x C Source Debugger User's Guide</i> .) |
| <input type="checkbox"/> | miscellaneous materials | Blank, formatted disks |

Software checklist

- | | | |
|--------------------------|-------------------------|--|
| <input type="checkbox"/> | operating system | MS-DOS or PC-DOS (version 3.0 or later)
Optional: MS-Windows (version 3.0 or later) |
| <input type="checkbox"/> | software tools | TMS320 floating-point DSP (C3x/C4x) C compiler, assembler, and linker |
| <input type="checkbox"/> | required file † | <i>emurst.exe</i> resets the 'C3x emulator |
| <input type="checkbox"/> | optional files † | <i>emuinit.cmd</i> is a general-purpose batch file that contains debugger commands. The version of this file that's shipped with the debugger defines a 'C3x memory map. If this file isn't present when you first invoke the debugger, then all memory is invalid at first. When you first start using the debugger, this memory map should be sufficient for your needs. Later, you may want to define your own memory map. For information about setting up your own memory map, refer to the <i>Defining a Memory Map</i> chapter in the <i>TMS320C3x C Source Debugger User's Guide</i> . |
| <input type="checkbox"/> | † | <i>init.clr</i> is a general-purpose screen configuration file. If this file isn't present when you invoke the debugger, the debugger uses the default screen configuration. |
| <input type="checkbox"/> | † | The default configuration is for color monitors; an additional file, <i>mono.clr</i> , can be used for monochrome monitors. When you first start to use the debugger, the default screen configuration should be sufficient for your needs. Later, you may want to define your own custom configuration.

For information about these files and about setting up your own screen configuration, refer to the <i>Customizing the Debugger Display</i> chapter in the <i>TMS320C3x C Source Debugger User's Guide</i> . |

† Included as part of the debugger package

2. Step 1: Installing the Emulator Board in Your PC

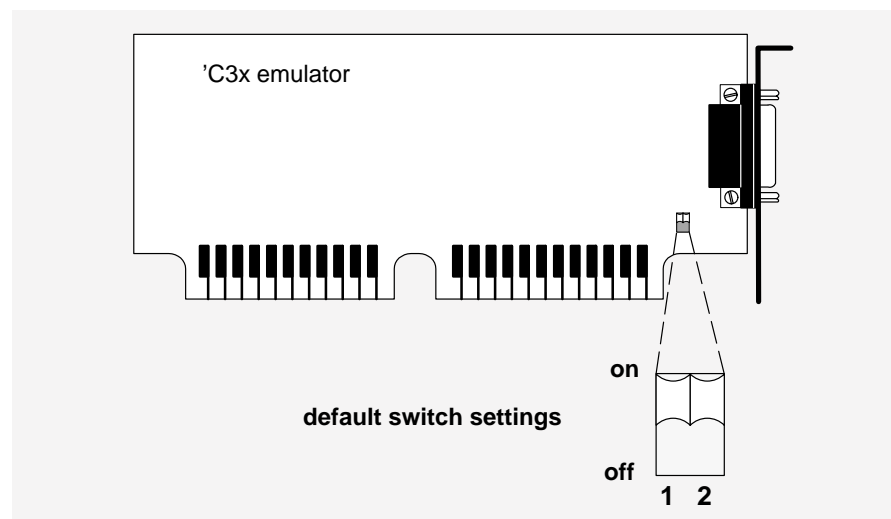
This section contains the hardware installation information for the emulator.

Preparing the emulator board for installation

Before you install the emulator board, you must be sure that the board's switches are set to correctly identify the I/O space that the board can use. The emulator uses 32 bytes of the PC I/O space; two switches on the board identify this space.

Figure 1 shows where these switches are on the emulator and identifies the switch numbers.

Figure 1. Emulator Board I/O Switches



Switches are shipped in the default settings shown here and are listed in Table 1. If you use an I/O space that differs from the default, change the switch settings. Table 1 shows alternate settings.

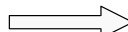
In most cases, you can leave the switch settings in the default position. However, you must ensure that the 'C3x emulator I/O space does not conflict with other bus settings. For example, if you've installed a bus mouse in your system, you may not be able to use the default switch settings for the I/O space—the mouse might use this space. Refer to your PC technical reference manual and your other hardware-board manuals to see if there are any I/O space conflicts. If you find a conflict, use one of the alternate settings shown in Table 1.

Table 1. *Emulator Board Switch Settings*

	Address Range	switch #	
		1	2
default	0x0240–0x025F	on	on
	0x0280–0x029F	on	off
	0x0320–0x033F	off	on
	0x0340–0x035F	off	off

Some of the other installation steps require you to know which switch settings you used. If you reset the I/O switches, note the modified settings here for later reference.

Table 2. *Your Switch Settings*

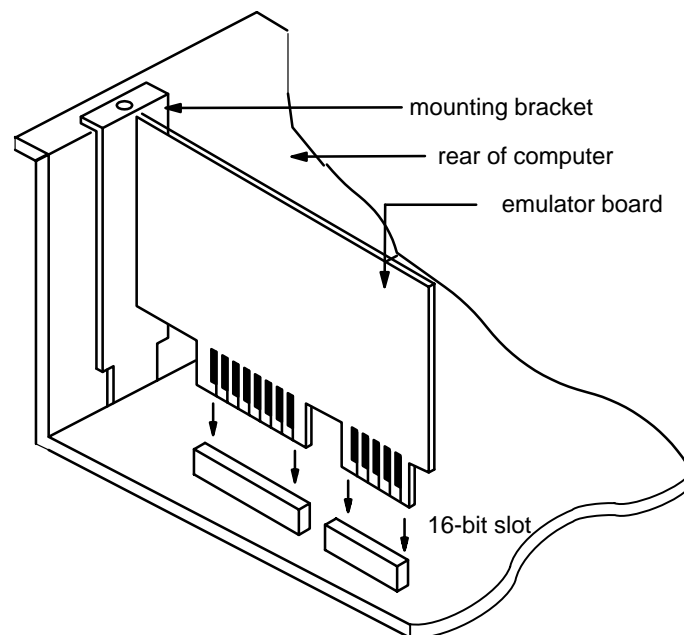
	Address Range	Switch #	
		1	2
			

Setting the emulator board into your PC

After you've prepared the emulator board for installation, follow these steps.

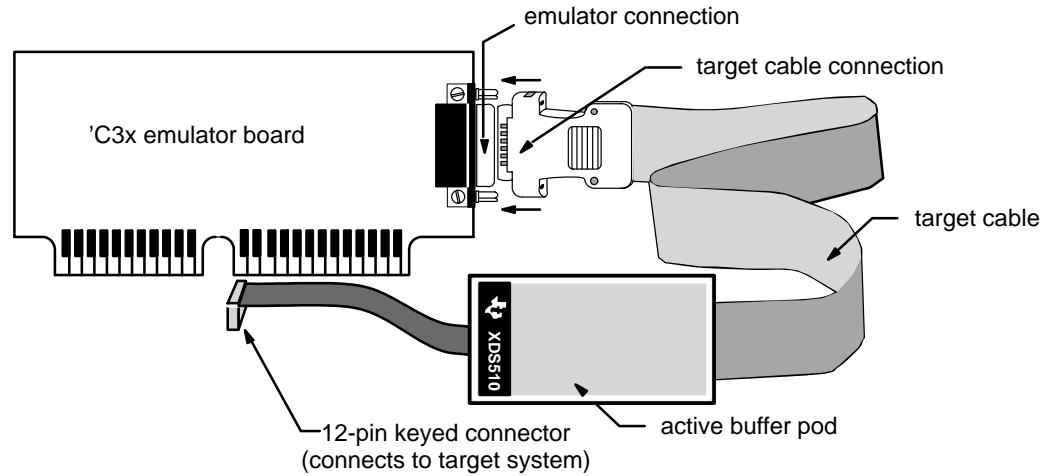
- Step 1:** Turn off your PC's power. Leave the power cord plugged in so that the computer is properly grounded.
- Step 2:** Remove the cover of your PC.
- Step 3:** Remove the mounting bracket from an unused 16-bit slot.
- Step 4:** Install the emulator board in a 16-bit slot (see Figure 2).

Figure 2. Emulator Board Installation



- Step 5:** Tighten down the mounting bracket.
- Step 6:** Plug the emulator target cable into the emulator board (see Figure 3). The cable is a 25-pin DSUB connector, shaped to ensure proper connection.
- Step 7:** Replace the PC cover.
- Step 8:** Turn on the PC's power.

Figure 3. Emulator Target Cable and Board



Don't connect or disconnect the target cable while the PC is powered up.

Be very careful with the target cable connectors. Connect them gently; forcing the connectors into position may damage them

Remember, the connector is keyed. Be sure to connect the cable so that the key fits into its slot.

3. Step 2: Connecting the Emulator to Your Target System

Figure 4 shows a typical setup using the emulator, target cable, and your target system.

Figure 4. Typical Setup Using the 'C3x Emulator and Your Target System

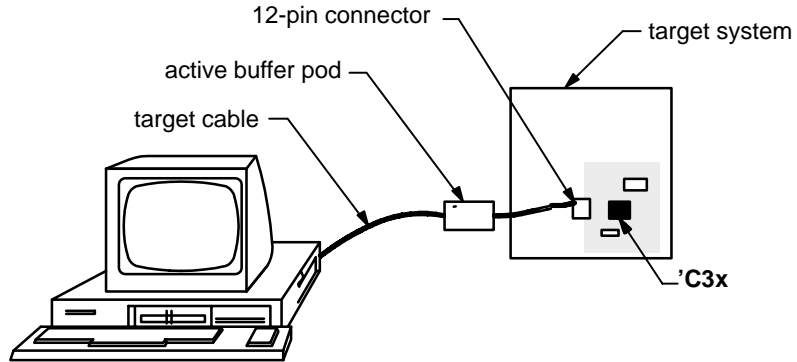
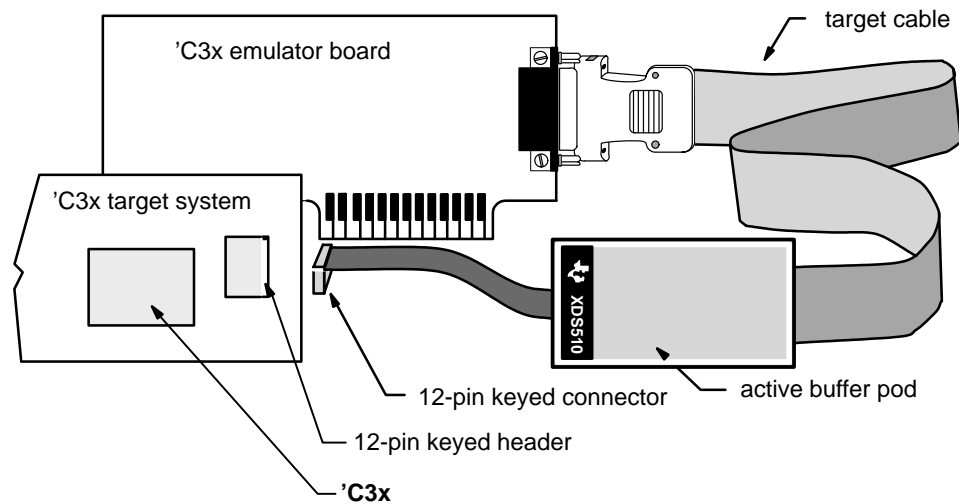


Figure 5 shows how you connect the emulator and target cable to your target system. In most cases, the target system will be a 'C3x board of your own design.

Figure 5. Connecting the 'C3x Emulator to Your Target System



4. Step 3: Installing the Debugger Software

This section explains the process of installing the debugger software on a hard-disk system.

- 1) Make a backup copy the DOS and/or MS-Windows debugger product disk. (If necessary, refer to the DOS manual that came with your computer.)
- 2) On your hard disk or system disk, create a directory named *c3xhll*. This directory will contain the 'C3x C source debugger software. To create this directory, enter:

```
MD C:\C3XHLL
```

- 3) Insert either the DOS or MS-Windows debugger product disk into drive A. Copy the contents of the disk:

```
COPY A:\*.* C:\C3XHLL\*.* /V
```

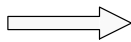
Repeat this step for the other product disk if you want to be able to run both the DOS and MS-Windows versions of the debugger.

The DOS version of the debugger executable file is called *emu3x.exe*, and the MS-Windows version of the debugger executable file is called *emu3xw.exe*. Throughout this document, the executable for the debugger is referred to as simply *emu3x*.

5. Step 4: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- Modify the PATH statement to identify the *c3xhll* directory.
- Define environment variables so that the debugger can find the files it needs.
- Identify any nondefault I/O space used by the emulator.
- Reset the emulator board.



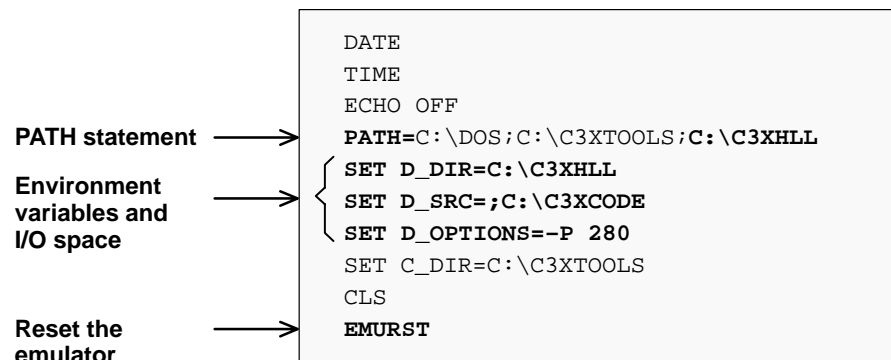
Not only must you do these things before you invoke the debugger for the first time, *you must do them any time you power up or reboot your PC.*

You can accomplish these tasks by entering individual DOS commands, but it's simpler to put the commands in a batch file. You can edit your systems *autoexec.bat* file; in some cases, modifying the *autoexec* may interfere with other applications running on your PC. So, if you prefer, you can create a separate batch file that performs these tasks.

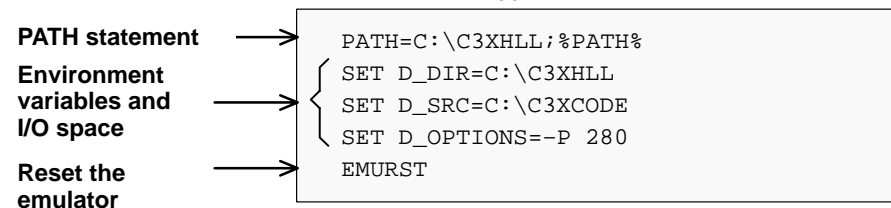
Figure 6 (a) shows an example of an autoexec.bat file that contains the suggested modifications (highlighted in bold type). Figure 6 (b) shows a sample batch file that you could create instead of editing the autoexec.bat file (for the purpose of discussion, assume that this sample file is named *initdb.bat*). The subsections following the figure explain these modifications.

Figure 6. DOS-Command Setup for the Debugger

(a) Sample autoexec.bat file to use with the debugger and emulator



(b) Sample initdb.bat file to use with the debugger and emulator



Invoking the new or modified batch file

- If you modify the autoexec.bat file, be sure to invoke it before invoking the debugger for the first time. To invoke this file, enter:

AUTOEXEC

- If you create an initdb.bat file, you must invoke it before invoking the debugger for the first time. If you are using MS-Windows, invoke initdb.bat *before* entering MS-Windows. You'll need to invoke initdb.bat any time that you power up or reboot your PC. To invoke this file, enter:

INITDB

Modifying the PATH statement

Define a path to the debugger directory. The general format for doing this is:

```
PATH=C:\C3XHLL
```

This allows you to invoke the debugger without specifying the name of the directory that contains the debugger executable file.

- If you are modifying an autoexec that already contains a PATH statement, simply include ;C:\c3xhll at the end of the statement, as shown in Figure 6 (a).
- If you are creating an initdb.bat file, use a different format for the PATH statement, as shown in Figure 6 (b):

```
PATH=C:\C3XHLL;%PATH%
```

The addition of ;%path% ensures that this PATH statement won't undo PATH statements in any other batch files (including the autoexec.bat file).

Setting up the environment variables

An environment variable is a special system symbol that the debugger uses for finding or obtaining certain types of information. The debugger uses three environment variables named D_DIR, D_SRC, and D_OPTIONS. The next three steps tell you how to set up these environment variables. The format for doing this is the same for both the autoexec.bat and initdb.bat files.

- Set up the D_DIR environment variable to identify the c3xhll directory:

```
SET D_DIR=C:\C3XHLL
```

(Be careful not to precede the equal sign with a space.)

This directory contains auxiliary files (emurst, emuinit.cmd, etc.) that the debugger needs.

- Set up the D_SRC environment variable to identify any directories that contain program source files that you'll want to look at while you're debugging code. The general format for doing this is:

```
SET D_SRC=pathname1;pathname2...
```

(Be careful not to precede the equal sign with a space.)

For example, if your 'C3x programs were in a directory named *csource* on drive C, the D_SRC setup would be:

```
SET D_SRC=C:\CSOURCE
```

- You can use several options when you invoke the debugger. If you use the same options over and over, it's convenient to specify them with D_OPTIONS. The general format for doing this is:

SET D_OPTIONS= [*object filename*] [*debugger options*]

(Be careful not to precede the equal sign with a space.)

This tells the debugger to load the specified object file and use the specified options each time you invoke the debugger. These are the options that you can identify with D_OPTIONS:

```

-b          -bb          -i pathname      -p port address
-profile    -s          -t filename      -v
    
```

Note that you can override D_OPTIONS by invoking the debugger with the -x option.

For more information about options, see the invocation instructions in the *TMS320C3x C Source Debugger User's Guide*.

Identifying the correct I/O switches

Refer to your entries in Table 2 (page 5). If you didn't modify the I/O switches, skip this step.

If you modified the I/O switch settings, you must use the debugger's -p option to identify the I/O space that the emulator is using. You can do this each time you invoke the debugger, or you can specify this information by using the D_OPTIONS environment variable. Table 3 lists the nondefault I/O switch setting and the appropriate line that you can add to the autoexec.bat or initdb.bat file.

Table 3. Identifying Nondefault I/O Address Space

Address Range	switch #		Add this line to the batch file
	1	2	
0x0280-0x029F	on	off	SET D_OPTIONS=-p 280
0x0320-0x033F	off	on	SET D_OPTIONS=-p 320
0x0340-0x035F	off	off	SET D_OPTIONS=-p 340

Resetting the emulator

To reset the emulator, add this line to the autoexec.bat or initdb.bat file:

```
emurst
```


6. Step 5: Verifying the Installation

To ensure that you have correctly installed the emulator and debugger software, enter this command at the system prompt:

```
emu3x c:\c3xh11\sample
```

You should see a display similar to this one:

The screenshot displays the emu3x debugger interface with the following sections:

- DISASSEMBLY:** A list of instructions with addresses, disassembly codes, and comments. For example, 80985d: 00809938 ABS1 IOF,R0. Comments include 'c_int00:' and 'exit:'.
- CPU:** A vertical list of registers and their values, such as PC 0080985e, SP 00000755, R0 00000003, etc.
- COMMAND:** A text window showing copyright information for Texas Instruments, Silicon Revision 2, Emulator Revision 3, and the message 'Loading sample.out Done'. The prompt '>>>' is visible.
- MEMORY:** A grid showing memory addresses and their corresponding hexadecimal values, ranging from 00000000 to 00000020.

- If you see a display similar to this one, you have correctly installed your emulator and debugger.
- If you see a display in which the lines of code show ADD instructions, your emulator board may not be installed snugly. Check your board to see if it is correctly installed, and re-enter the command above.
- If you see a display in which the lines of code say *Invalid address* or the fields in the MEMORY window are shown in red, the debugger may not be able to find the emuinit.cmd file. Check for the file in the directories specified by the D_SRC environment variable, or ensure that the file is in the current directory. Re-enter the command above.

- If you don't see a display, then your debugger or board may not be installed properly. Go back through the installation instructions and be sure that you have followed each step correctly; then re-enter the command above.

Installation error messages

While invoking the debugger, you may see the following message:

```
CANNOT INITIALIZE THE TARGET SYSTEM ! !  
- Check I/O configuration  
- Check cabling and target power
```

Check these areas for possible problems:

- Is the target power on?
- Is the emulator board installed snugly?
- Is the device installed snugly?
- Is the cable connecting your emulator and target system loose?
- Is your target board getting the correct voltage?
- Does the emurst command appear at the end of either your autoexec.bat or initdb.bat file? This command must be executed *after* you powered up the target board.
- Is your port address set correctly:
 - Check to be sure the `-p` option used with the `D_OPTIONS` environment variable matches the I/O address defined by your switch settings (refer to *Your Switch Settings*, Table 2, and *Identifying Nondefault I/O Address Space*, Table 3).
 - Check for a conflict in address space with another bus setting. If you have a conflict, change the switches on your board to one of the alternate settings in Table 1. Modify the `-p` option of the `D_OPTIONS` environment variable to reflect the change in your switch settings.

After you have checked all of the above, repeat the verification instructions in Section 6.

7. Using the Debugger With MS-Windows

If you're using MS-Windows, you can freely move or resize the debugger display on the screen. If the resized display is bigger than the debugger requires, the extra space is not used. If the resized display is smaller than required, the debugger display is clipped. Note that when the display is clipped, it can't be scrolled.

You should run MS-Windows in either the standard mode or the 386 enhanced mode to get the best results.