

TMS320C2xx Simulator

Getting Started Guide



TMS320C2xx Simulator Getting Started Guide

SPRU176
April 1996



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.

TRADEMARKS

HP 9000 Series 700, HP-UX, and PA-RISC are trademarks of Hewlett-Packard Company.

MS-DOS and Windows are registered trademarks of Microsoft Corp.

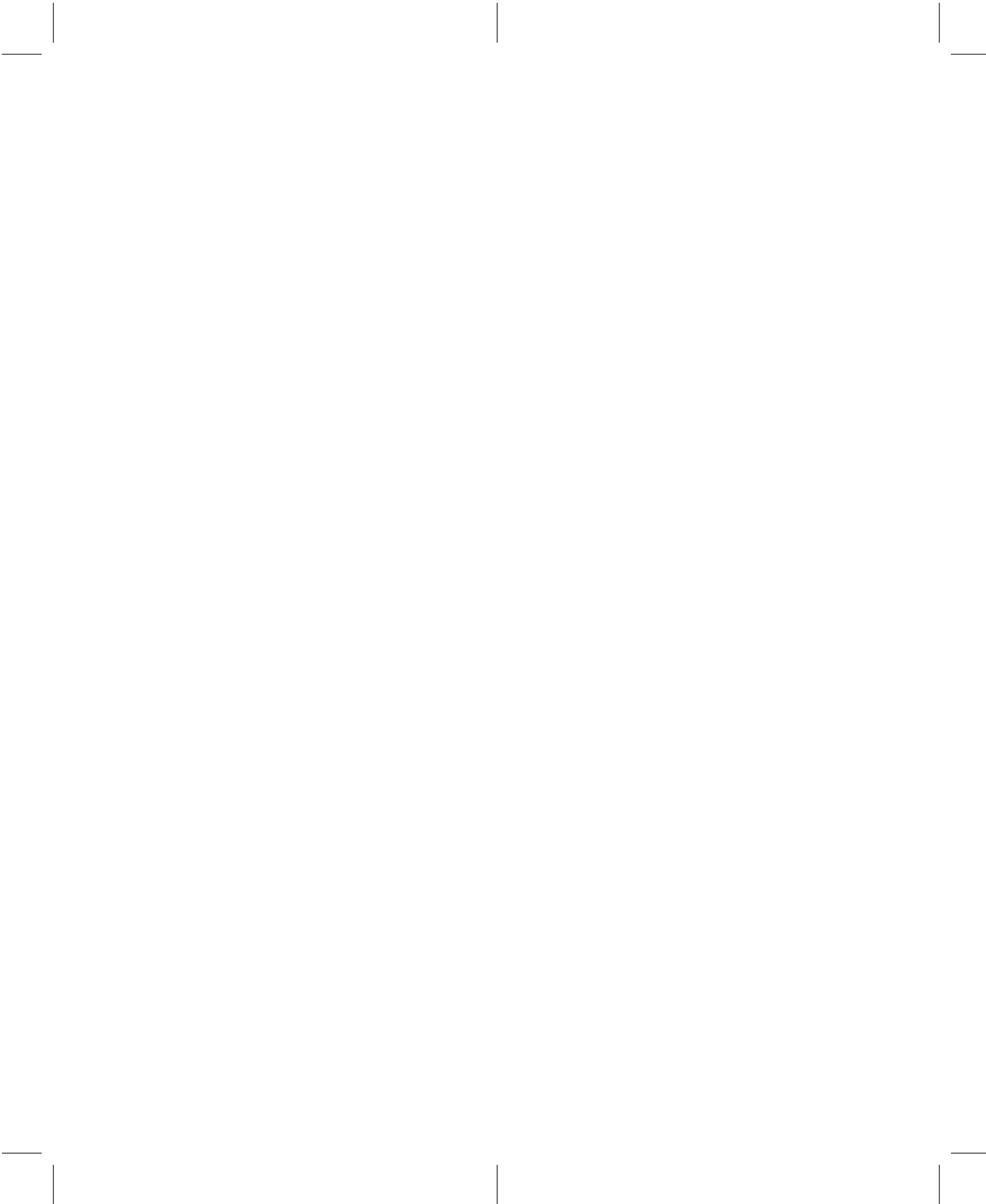
OpenWindows, Solaris, and SunOS are trademarks of Sun Microsystems, Inc.

PC and PC-DOS is a trademark of International Business Machines Corp.

SPARC is a trademark of SPARC International, Inc.

SPARCstation is licensed exclusively to Sun Microsystems, Inc.

X Window System is a trademark of the Massachusetts Institute of Technology.



Contents

1	Installing the Simulator and C Source Debugger With DOS	1-1
	<i>Lists the hardware and software you need to install the simulator and C source debugger; provides installation instructions for PC systems running MS-DOS™ or PC-DOS™.</i>	
1.1	What You Need	1-2
	Hardware checklist	1-2
	Software checklist	1-3
1.2	Step 1: Installing the Simulator and Debugger Software	1-4
1.3	Step 2: Setting Up the Debugger Environment	1-5
	Defining an initialization batch file	1-6
	Changing the autoexec.bat file	1-7
1.4	Step 3: Verifying the Installation	1-8
1.5	Using the Debugger With Windows	1-9
2	Installing the Simulator and C Source Debugger With SunOS	2-1
	<i>Lists the hardware and software you need to install the simulator and C source debugger; provides installation instructions for SPARCstations running SunOS™ or Solaris™.</i>	
2.1	What You Need	2-2
	Hardware checklist	2-2
	Software checklist	2-3
2.2	Step 1: Installing the Simulator and Debugger Software	2-4
	Mounting the CD-ROM	2-4
	Copying the files	2-5
	Unmounting the CD-ROM	2-5
2.3	Step 2: Setting Up the Debugger Environment	2-6
	Reinitializing your shell	2-7
2.4	Step 3: Verifying the Installation	2-8
2.5	Using the Debugger With the X Window System	2-9
	Using the keyboard's special keys	2-9
	Changing the debugger font	2-10
	Color mappings on monochrome screens	2-10

3	Installing the Simulator and C Source Debugger With HP-UX	3-1
	<i>Lists the hardware and software you need to install the simulator and C source debugger; provides installation instructions for HP systems running HP-UX™.</i>	
3.1	What You Need	3-2
	Hardware checklist	3-2
	Software checklist	3-3
3.2	Step 1: Installing the Simulator and Debugger Software	3-4
	Mounting the CD-ROM	3-4
	Copying the files and setting up the simulator	3-4
	Unmounting the CD-ROM	3-5
3.3	Step 2: Setting Up the Debugger Environment	3-6
	Reinitializing your shell	3-7
3.4	Step 3: Verifying the Installation	3-8
3.5	Using the Debugger With the X Window System	3-9
	Using the keyboard's special keys	3-9
	Changing the debugger font	3-10
	Color mappings on monochrome screens	3-10

Installing the Simulator and C Source Debugger With DOS

This chapter provides instructions to help you install the TMS320C2xx simulator and the C source debugger on PC™ systems running MS-DOS or PC-DOS. You can also use the debugger with Windows. When you complete the installation, turn to the *TMS320C2xx C Source Debugger User's Guide*.

Topic	Page
1.1 What You Need	1-2
1.2 Step 1: Installing the Simulator and Debugger Software	1-4
1.3 Step 2: Setting Up the Debugger Environment	1-5
1.4 Step 3: Verifying the Installation	1-8
1.5 Using the Debugger With Windows	1-9

1.1 What You Need

To install the 'C2xx C source debugger and simulator, you need the items in the following hardware and software checklists.

Hardware checklist

- | | | |
|--------------------------|--------------------------------|---|
| <input type="checkbox"/> | Host | An IBM PC/AT or 100%-compatible ISA/EISA-based PC with a hard-disk system and a 1.2M floppy-disk drive; a 386 or higher is highly recommended |
| <input type="checkbox"/> | Memory | Minimum of 640K bytes; in addition, if you are running under Windows, you'll need at least 256K bytes of extended memory |
| <input type="checkbox"/> | Display | Monochrome or color monitor (color recommended) |
| <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 several 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 the larger screen sizes. The 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 information in the <i>TMS320C2xx 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 | TMS320C1x/C2x/C2xx/C5x assembler and linker
Optional: TMS320C2x/C2xx/C5x C compiler |
| <input type="checkbox"/> | Optional files included with the debugger package | <i>siminit.cmd</i> is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a default 'C2xx memory map. 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 defining your own memory map, refer to the defining a memory map chapter in the <i>TMS320C2xx C Source Debugger User's Guide</i> . If a memory map batch file isn't present when you invoke the debugger, all memory is invalid initially. |
| <input type="checkbox"/> | | <i>sim203.cmd</i> and <i>sim209.cmd</i> are batch files containing commands that configure a memory map of 'C203 and 'C209 devices, respectively. |
| <input type="checkbox"/> | | <i>init.clr</i> is a general-purpose screen configuration file. If <i>init.clr</i> isn't present when you invoke the debugger, the debugger uses the default screen configuration. |
| <input type="checkbox"/> | | <i>init.25</i> , <i>init.43</i> , and <i>init.50</i> have been provided for basic 80 x 25, 80 x 43, and 80 x 50 screen sizes, respectively. The <i>init.clr</i> file brings up the debugger in 80 x 25 mode. To bring the debugger up in another mode, copy one of the <i>init.xx</i> files to the <i>init.clr</i> file. |
| <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 customizing the debugger display section in the *TMS320C2xx C Source Debugger User's Guide*.

1.2 Step 1: Installing the Simulator and Debugger Software

This section explains how to install the simulator and debugger on a hard-disk system.

- 1) Make a backup copy of each product disk(s).
- 2) On your hard disk, create a directory named `sim2xx`. This directory will contain the 'C2xx software.

```
MD C:\sim2xx
```

- 3) Insert the product disk(s) into drive A. Copy the contents of the disk(s).

```
COPY A:\*.* C:\sim2xx\*.* /V
```

The DOS version of the debugger is named `sim2xx.exe`, and the Windows version of the debugger is named `sim2xxw.exe`. Throughout this document, the debugger is referred to simply as `sim2xx`.

1.3 Step 2: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must identify the items that are listed in Table 1–1. You can specify this information either in an initialization batch file (see page 1-6) or in your *autoexec.bat* file (see page 1-7).

Table 1–1. Debugger Environment Variables

To identify . . .	Use a statement with this format . . .
Directory with executable files for the C source debugger	PATH=drive :\ directory
Directory with debugger data files, such as <i>init.cmd</i> and <i>init.clr</i>	SET D_DIR=drive :\ directory
Directory with the program source files that you want to debug	SET D_SRC=drive :\ directory
Address of the emulator port on your PC and other options that you want to use every time that you invoke the debugger	SET D_OPTIONS= [object filename] [options] <div> <div>[object filename]</div> <div>Names the file that you want to load every time that you invoke the debugger.</div> </div> <div> <div>[options]</div> <div>Indicates the port address and other options; for more information, see the <i>TMS320C2xx C Source Debugger User's Guide</i>.</div> </div> <div> <div>-p port address</div> <div>Identifies the emulator port on your PC:</div> <div> <div>378</div> <div>default port address for the XDS510PP (LPT1 on most PCs); to verify the address of the printer port where you connected the XDS510PP, see your PC documentation.</div> </div> <div> <div>240</div> <div>default port address for the XDS510; for more information, see the <i>XDS51x Emulator Installation Guide</i>.</div> </div> </div> <div> <div>-b</div> <div>Selects a screen size of 80 characters by 43 lines (EGA or VGA)</div> </div> <div> <div>-bb</div> <div>Selects a screen size of 80 characters by 50 lines (VGA only)</div> </div> <div> <div>-font height</div> <div>Uses the Windows Terminal font closest in point size to the specified height</div> </div> <div> <div>-i pathname</div> <div>Identifies additional directories</div> </div> <div> <div>-mv version</div> <div>Specifies the memory map to use with the simulator</div> </div> <div> <div>-profile</div> <div>Allows you to enter the profiling environment</div> </div> <div> <div>-s</div> <div>Loads only the symbol table for a named object file</div> </div> <div> <div>-t filename</div> <div>Identifies a new initialization file</div> </div> <div> <div>-v</div> <div>Loads the object code with a minimal symbol table</div> </div>

Note: When you invoke the debugger, you can include **-x** on the command line to override any D_OPTIONS in the initialization file or in your *autoexec.bat* file.

Defining an initialization batch file

To create an initialization file named *initdb.bat*, follow these steps using a text editor; be careful that no spaces precede the equal (=) sign wherever it appears.

- 1) To specify the location of the C source debugger executable files and to ensure that this statement does not overwrite PATH statements in other batch files, type:

```
PATH=C:\sim2xx;%PATH%
```

- 2) To specify the directory with the C source debugger data files, type:

```
SET D_DIR=C:\sim2xx
```

- 3) To specify directories that contain the program source files that you want to debug, use the following format to set the D_SRC environment variable:

```
SET D_SRC=pathname1 [pathname2 . . .]
```

- 4) To specify the emulator port address and other options, use the following format to define the D_OPTIONS environment variable:

```
SET D_OPTIONS=[-p port_address] [ options ]
```

- 5) To add the emulator-reset command to the file, type:

```
EMURST
```

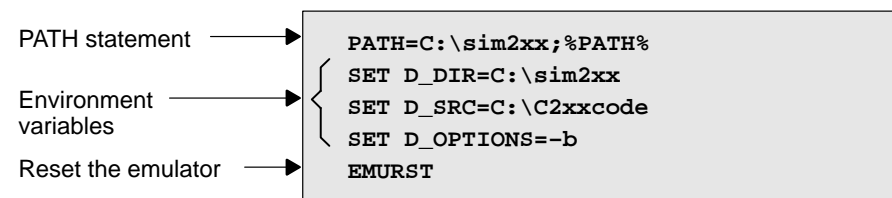
- 6) Save the file as *initdb.bat*, and then exit the text editor.

- 7) Before you start Windows and any time that you power up or reboot your PC, invoke this file from a DOS prompt by entering:

```
INITDB 
```

Figure 1–1 shows a sample initialization file that contains the required path, environment variables, and emulator-reset statement.

Figure 1–1. Sample Initialization Batch File



Changing the autoexec.bat file

If you are sure that no programs will be affected by changing your autoexec.bat file, you can specify the debugger environment in that file. To change your autoexec.bat file, follow these steps using a text editor; be careful that no spaces precede the equal (=) sign wherever it appears.


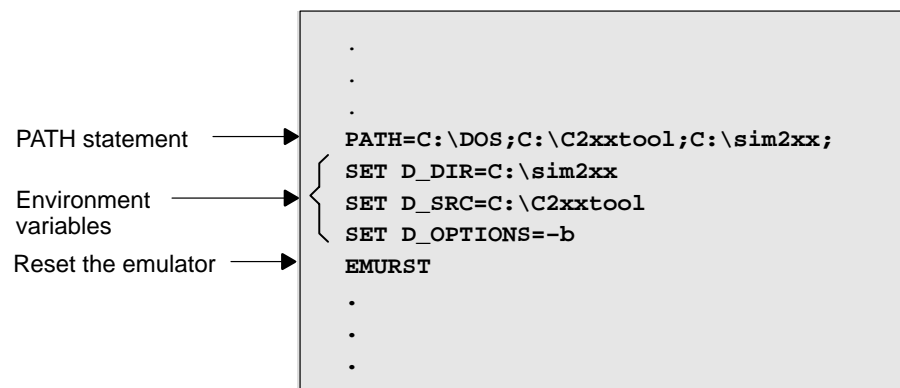
- 1) At the end of the PATH statement in your autoexec.bat file, type:
`;C:\sim2xx;`
- 2) To specify the directory that contains the C source debugger files, type:
`SET D_DIR=C:\sim2xx`
- 3) To specify directories that contain the program source files that you want to debug, use the following format to set the D_SRC environment variable:
`SET D_SRC=pathname1 [pathname2 . . .]`
- 4) To specify the emulator port address and other options, use the following format to define the D_OPTIONS environment variable:
`SET D_OPTIONS=[-p port_address] [options]`
- 5) To add the emulator-reset command to the file, type:
`EMURST`
- 6) Save the file, and then exit the text editor.
- 7) Before you invoke the debugger for the first time, invoke the autoexec.bat file from an DOS prompt by entering:
`AUTOEXEC` 

Figure 1–2 shows a portion of the autoexec.bat file with the required path, environment variables, and emulator-reset statement.

Figure 1–2. Sample autoexec.bat File



```

.
.
.
PATH=C:\DOS;C:\C2xxtool;C:\sim2xx;
SET D_DIR=C:\sim2xx
SET D_SRC=C:\C2xxtool
SET D_OPTIONS=-b
EMURST
.
.
.

```

Annotations:

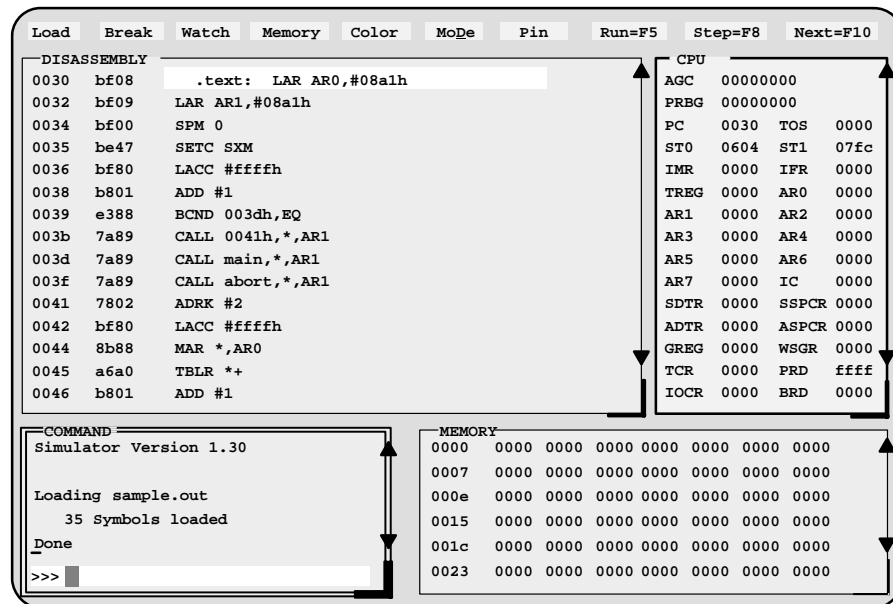
- PATH statement → `PATH=C:\DOS;C:\C2xxtool;C:\sim2xx;`
- Environment variables → `SET D_DIR=C:\sim2xx`, `SET D_SRC=C:\C2xxtool`, `SET D_OPTIONS=-b`
- Reset the emulator → `EMURST`

1.4 Step 3: Verifying the Installation

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

```
sim2xx c:\sim2xx\sample
```

You should see a display similar to this one:



- ☐ If you see a display similar to this one, you have correctly installed your simulator and debugger.
- ☐ If you don't see a similar display, your debugger or simulator may not be installed properly. Go back through the installation instructions and be sure that you have followed each step correctly; then reenter the command above.

1.5 Using the Debugger With Windows

If you're using 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 display is clipped. Note that when the display is clipped, it can't be scrolled.

You may want to create an icon to make it easier to invoke the debugger from within the Windows environment. Refer to your Windows manual for details.

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

Installing the Simulator and C Source Debugger With SunOS

This chapter provides instructions to help you install the 'C2xx simulator and the C source debugger on a SPARCstation running SunOS™ or Solaris™. When you complete the installation, turn to the *TMS320C2xx C Source Debugger User's Guide*.

Topic	Page
2.1 What You Need	2-2
2.2 Step 1: Installing the Simulator and Debugger Software	2-4
2.3 Step 2: Setting Up the Debugger Environment	2-6
2.4 Step 3: Verifying the Installation	2-8
2.5 Using the Debugger With the X Window System	2-9

2.1 What You Need

To install the 'C2xx C source debugger and simulator, you need the items in the following hardware and software checklists.

Hardware checklist

<input type="checkbox"/>	Host	A SPARCstation or a system that is 100% compatible with a SPARCstation 2 class or higher
<input type="checkbox"/>	Display	Monochrome or color monitor (color recommended)
<input type="checkbox"/>	Disk space	2M bytes of disk space
<input type="checkbox"/>	Required hardware	CD-ROM drive
<input type="checkbox"/>	Optional hardware	Mouse

Software checklist

- | | | |
|--------------------------|--|---|
| <input type="checkbox"/> | Operating system | OpenWindows™ version 3.0 (or higher) running under SunOS version 4.1.3 (or higher) or SunOS version 5.x (also known as Solaris 2.x). |
| <input type="checkbox"/> | Root privileges | If you are running SunOS 4.1.x, 5.0, or 5.1, you <i>must</i> have root privileges to mount and unmount the CD-ROM. If you don't, get help from your system administrator. |
| <input type="checkbox"/> | Software tools | TMS320C1x/'C2x/'C2xx/'C5x assembler and linker
Optional: TMS320C2x/'C2xx/'C5x C compiler |
| <input type="checkbox"/> | Optional files included with the debugger package | <i>siminit.cmd</i> is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a 'C2xx memory map. 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 defining your own memory map, refer to the defining a memory map chapter in the <i>TMS320C2xx C Source Debugger User's Guide</i> . If a memory map batch file isn't present when you invoke the debugger, all memory is invalid initially. |
| <input type="checkbox"/> | | <i>sim203.cmd</i> and <i>sim209.cmd</i> are batch files containing commands that configure a memory map for 'C203 and 'C209 devices, respectively. |
| <input type="checkbox"/> | | <i>init.clr</i> is a general-purpose screen configuration file. If <i>init.clr</i> isn't present when you invoke the debugger, the debugger uses the default screen configuration. |
| <input type="checkbox"/> | | <i>init.25</i> , <i>init.43</i> , and <i>init.50</i> have been provided for basic 80 x 25, 80 x 43, and 80 x 50 screen sizes, respectively. The <i>init.clr</i> file brings up the debugger in 80 x 25 mode. To bring the debugger up in another mode, copy one of the <i>init.xx</i> files to the <i>init.clr</i> file. |
| <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 customization information in the *TMS320C2xx C Source Debugger User's Guide*.

2.2 Step 1: Installing the Simulator and Debugger Software

This section explains how to install the simulator and debugger software on your hard-disk system. The software package is shipped on a CD-ROM. To install the software, you must mount the CD-ROM, copy the files, and unmount the CD-ROM.

Mounting the CD-ROM

Note: Root Privileges

If you are running SunOS 4.1.x, 5.0, or 5.1, you *must* have root privileges to mount the CD-ROM. If you don't, get help from your system administrator.

The steps to mount the CD-ROM vary according to your operating-system version:

- ☐ If you have SunOS 4.1.x, load the CD-ROM into the drive and enter the following from a command shell:

```
mount -rt hsfs /dev/sr0 /cdrom
exit
cd /cdrom/sparc
```

- ☐ If you have SunOS 5.0 or 5.1, load the CD-ROM into the drive and enter the following from a command shell:

```
mount -rF hsfs /dev/sr0 /cdrom
exit
cd /cdrom/cdrom0/sparc
```

- ☐ If you have SunOS 5.2 or higher:

- ☒ If your CD-ROM drive is already attached, load the CD-ROM into the drive and enter the following from a command shell:

```
cd /cdrom/cdrom0/sparc
```

- ☒ If you do not have a CD-ROM drive attached, you must shut down your system to the PROM level, attach the CD-ROM drive, and enter the following:

```
boot -r
```

After you log into your system, load the CD-ROM into the drive and enter the following from a command shell:

```
cd /cdrom/cdrom0/sparc
```

Copying the files

After you mount the CD-ROM, you must create the directory that will contain the debugger software and copy the software to that directory.

- 1) Create a directory named *sim2xx* on your hard disk. To create this directory, enter:

```
mkdir /your_pathname/sim2xx
```

- 2) Copy the files from the CD-ROM to your hard-disk system:

```
cp -r * /your_pathname/sim2xx
```

Unmounting the CD-ROM

Note: Root Privileges

If you are running SunOS 4.1.x, 5.0, or 5.1, you *must* have root privileges to unmount the CD-ROM. If you don't, get help from your system administrator.

You must unmount the CD-ROM after copying the files.

- ☐ If you have SunOS 4.1.x, 5.0, or 5.1, enter the following from a command shell:

```
cd  
umount /cdrom  
eject /dev/sr0  
exit
```

- ☐ If you have SunOS 5.2 or higher, enter the following from a command shell:

```
cd  
eject
```

2.3 Step 2: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must identify the items that are listed in Table 2–1. You specify this information in your shell configuration file in your home directory (for example, the `.cshrc` file for a C shell). After modifying your shell configuration file, you must reinitialize it.

Table 2–1. Debugger Environment Variables

To identify . . .	Use a statement with this format . . .	
Directory with executable files for the C source debugger	set path = (. /directory)	
Directory with debugger data files, such as <i>init.cmd</i> and <i>init.clr</i>	setenv D_DIR "/directory"	
Directory with the program source files that you want to debug	setenv D_SRC "/directory"	
For an X Window system, display the debugger on a different machine (see Section 2.5, <i>Using the Debugger With the X Window System</i> , on page 2-9)	setenv DISPLAY "machinename"	
Options that you want to use every time that you invoke the debugger	setenv D_OPTIONS [object filename] [options]	
	[object filename]	Names the file that you want to load every time that you invoke the debugger.
	[options]	Initialization options; for more information, see the <i>TMS320C2xx C Source Debugger User's Guide</i> .
	-b	Selects a screen size of 80 characters by 43 lines (EGA or VGA)
	-bb	Selects a screen size of 80 characters by 50 lines (VGA only)
	-d machine	For an X Window system, display the debugger on a different machine. Use instead of the DISPLAY environment variable.
	-i pathname	Identifies additional directories
	-mv version	Specifies the memory map to use with the simulator
	-profile	Allows you to enter the profiling environment
	-s	Loads only the symbol table for a named object file
	-t filename	Identifies a new initialization file
	-v	Loads the object code with a minimal symbol table


Note: When you invoke the debugger, you can include `-x` on the command line to override any `D_OPTIONS` in the initialization file or in your `autoexec.bat` file.

Figure 2–1. Sample Shell Configuration File for an X Window System

```
set path statement → set path = (. /bin /usr/ucb /usr/contrib/bin /usr/bin \  
                    /usr/openwin/bin /user/fred/sim2xx)  
Environment variables → { setenv D_DIR "/user/fred/sim2xx"  
                        setenv D_SRC "/user/fred/C2xxsource"  
                        setenv DISPLAY "barney:0"  
                        setenv D_OPTIONS "-b"  
Reset the emulator → enurst
```


Reinitializing your shell

When you modify your shell configuration file, you must ensure that the changes are made to your current session. For example, if you are using a C shell, use this command to reread the `.cshrc` file:

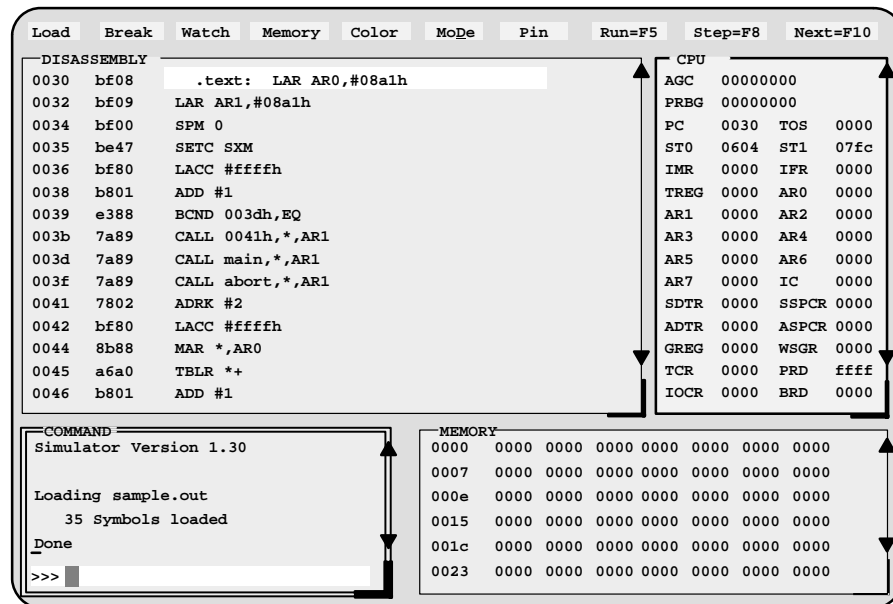
```
source ~/.cshrc 
```

2.4 Step 3: Verifying the Installation

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

sim2xx sample 

You should see a display similar to this one:



- ☐ If you see a display similar to this one, you have correctly installed your simulator and debugger.
- ☐ If you don't see a similar display, then your debugger or simulator may not be installed properly. Go back through the installation instructions and be sure that you have followed each step correctly; then reenter the command above.

2.5 Using the Debugger With the X Window System

If you're using the X Window System to run the 'C2xx debugger, you need to know about the keyboard's special keys, the debugger fonts, and using the debugger on a monochrome monitor.

Using the keyboard's special keys

The debugger uses some special keys that you can map differently from your particular keyboard. Some keyboards, such as the Sun Type 5 keyboard, may have these special symbols on separate keys. Other keyboards, such as the Sun Type 4 keyboard, do not have the special keys.

The special keys that the debugger uses are shown in the following table with their corresponding keysym. A *keysym* is a label that interprets a keystroke; it allows you to modify the action of a key on the keyboard.

Key	Keysym
(F1) to (F10)	F1 to F10
(PAGE UP)	Prior
(PAGE DOWN)	Next
(HOME)	Home
(END)	End
(INSERT)	Insert
(→)	Right
(←)	Left
(↑)	Up
(↓)	Down

Use the X utility `xev` to check the keysyms that are associated with your keyboard. If you need to change the keysym definitions, use the `xmodmap` utility. For example, you could create a file that contains the following commands and use that file with `xmodmap` to change a Sun Type 4 keyboard to match the keys listed above:

```
keysym R13      = End
keysym Down     = Down
keysym F35      = Next
keysym Left     = Left
keysym Right    = Right
keysym F27      = Home
keysym Up       = Up
keysym F29      = Prior
keysym Insert   = Insert
```

Refer to your X Window System documentation for more information about using `xev` and `xmodmap`.

Changing the debugger font

You can change the font of the debugger screen by using the `xrdb` utility and modifying the `.Xdefaults` file in your root directory. For example, to change the 'C2xx debugger fonts to Courier, add the following line to the `.Xdefaults` file:

```
sim2xx*font:courier
```

For more information about using `xrdb` to change the font, refer to your X Window System documentation.

Color mappings on monochrome screens

Although a color monitor is recommended, the following table shows the color mappings for monochrome screens:

Color	Appearance on Monochrome Screen
black	black
blue	black
green	white
cyan	white
red	black
magenta	black
yellow	white
white	white

Installing the Simulator and C Source Debugger With HP-UX

This chapter describes how to install the 'C2xx simulator and the C source debugger on an HP9000 series 700™ PA-RISC™ system running HP-UX. When you complete the installation, turn to the *TMS320C2xx C Source Debugger User's Guide*.

Topic	Page
3.1 What You Need	3-2
3.2 Step 1: Installing the Simulator and Debugger Software	3-4
3.3 Step 2: Setting Up the Debugger Environment	3-6
3.4 Step 3: Verifying the Installation	3-8
3.5 Using the Debugger With the X Window System	3-9

3.1 What You Need

To install the 'C2xx C source debugger and simulator, you need the items in the following hardware and software checklists.

Hardware checklist

<input type="checkbox"/>	Host	An HP9000 series 700 PA-RISC system
<input type="checkbox"/>	Display	Monochrome or color (color recommended)
<input type="checkbox"/>	Disk space	2M bytes of disk space
<input type="checkbox"/>	Required hardware	CD-ROM drive
<input type="checkbox"/>	Optional hardware	Mouse

Software checklist

- | | | |
|--------------------------|--|---|
| <input type="checkbox"/> | Operating system | HP-UX 9.x or higher. |
| <input type="checkbox"/> | Root privileges | You <i>must</i> have root privileges to mount and unmount the CD-ROM. If you don't, get help from your system administrator. |
| <input type="checkbox"/> | Software tools | TMS320C2xx assembler and linker
Optional: TMS320C2xx C compiler |
| <input type="checkbox"/> | Optional files included with the debugger package | <i>siminit.cmd</i> is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a 'C2xx memory map. 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 defining your own memory map, refer to the defining a memory map chapter in the <i>TMS320C2xx C Source Debugger User's Guide</i> . If a memory map batch file isn't present when you invoke the debugger, all memory is invalid initially. |
| <input type="checkbox"/> | | <i>sim203.cmd</i> and <i>sim209.cmd</i> are batch files containing commands that configure a memory map of 'C203 and 'C209 devices, respectively. |
| <input type="checkbox"/> | | <i>init.clr</i> is a general-purpose screen configuration file. If <i>init.clr</i> isn't present when you invoke the debugger, the debugger uses the default screen configuration. |
| <input type="checkbox"/> | | <i>init.25</i> , <i>init.43</i> , and <i>init.50</i> have been provided for basic 80×25, 80×43, and 80×50 screen sizes, respectively. The <i>init.clr</i> file brings up the debugger in 80×25 mode. To bring the debugger up in another mode, copy one of the <i>init.xx</i> files to the <i>init.clr</i> file. |
| <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 customization information in the <i>TMS320C2xx C Source Debugger User's Guide</i> . |

3.2 Step 1: Installing the Simulator and Debugger Software

This section explains how to install the simulator and debugger software on your hard-disk system. The software package is shipped on a CD-ROM. To install the software, you must mount the CD-ROM, copy the files, and unmount the CD-ROM.

Mounting the CD-ROM



Note: Root Privileges

You *must* have root privileges to mount the CD-ROM. If you don't, get help from your system administrator.

You can mount the CD-ROM using the UNIX mount command or the SAM (system administration manager):

- ☐ To use the UNIX mount command, follow these steps:

- 1) To mount the CD-ROM, enter:

```
mount -rt cdfs /dev/dsk/your_cdrom_device /cdrom   
exit 
```

- 2) Make the hp700 directory on the CD-ROM the current directory. For example, if the CD-ROM is mounted at /cdrom, enter:


```
cd /cdrom/hp700 
```

- ☐ To use SAM to mount the CD-ROM, see *System Administration Tasks*, the HP documentation about SAM, for instructions.

Copying the files

After you mount the CD-ROM, you must create the directory that will contain the debugger software and copy the software to that directory.

- 1) Create a directory named *sim2xx* on your hard disk. To create this directory, enter:

```
mkdir /your_pathname/sim2xx 
```

- 2) Copy the files from the CD-ROM to your hard-disk system:

```
cp -r * /your_pathname/sim2xx 
```


Unmounting the CD-ROM

Note: Root Privileges

You *must* have root privileges to unmount the CD-ROM. If you don't, get help from your system administrator.

You must unmount the CD-ROM after copying the files. Enter:

```
cd [2]
umount /cdrom [2]
exit [2]
```

3.3 Step 2: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must identify the items that are listed in Table 3–1. You specify this information in your shell configuration file in your home directory (for example, the `.cshrc` file for a C shell). After modifying your shell configuration file, you must reinitialize it.

Table 3–1. Debugger Environment Variables

To identify . . .	Use a statement with this format . . .	
Directory with executable files for the C source debugger	set path = (. /directory)	
Directory with debugger data files, such as <i>init.cmd</i> and <i>init.clr</i>	setenv D_DIR "/directory"	
Directory with the program source files that you want to debug	setenv D_SRC "/directory"	
For an X Window system, display the debugger on a different machine (see Section 3.5, <i>Using the Debugger With the X Window System</i> , on page 3-9.)	setenv DISPLAY "machinename"	
Options that you want to use every time that you invoke the debugger	setenv D_OPTIONS [object filename] [options]	
	[object filename]	Names the file that you want to load every time that you invoke the debugger.
	[options]	Initialization options; for more information, see the <i>TMS320C2xx C Source Debugger User's Guide</i> .
	-b	Selects a screen size of 80 characters by 43 lines (EGA or VGA)
	-bb	Selects a screen size of 80 characters by 50 lines (VGA only)
	-d machine	For an X Window system, display the debugger on a different machine. Use instead of the DISPLAY environment variable.
	-i pathname	Identifies additional directories
	-mv version	Specifies the memory map to use with the simulator
	-profile	Allows you to enter the profiling environment
	-s	Loads only the symbol table for a named object file
	-t filename	Identifies a new initialization file
	-v	Loads the object code with a minimal symbol table


Note: When you invoke the debugger, you can include `-x` on the command line to override any D_OPTIONS in the initialization file or in your `autoexec.bat` file.

Figure 3–1. Sample Shell Configuration File for an X Window System

```
set path statement → set path = (. /bin /usr/ucb /usr/contrib/bin /usr/bin \  
                    /usr/openwin/bin /user/fred/sim2xx)  
Environment variables → { setenv D_DIR "/user/fred/sim2xx"  
                        setenv D_SRC "/user/fred/C2xxsource"  
                        setenv DISPLAY "barney:0"  
                        setenv D_OPTIONS "-b"  
Reset the emulator → enurst
```

Reinitializing your shell

When you modify your shell configuration file, you must ensure that the changes are made to your current session. For example, if you are using a C shell, use this command to reread the `.cshrc` file:

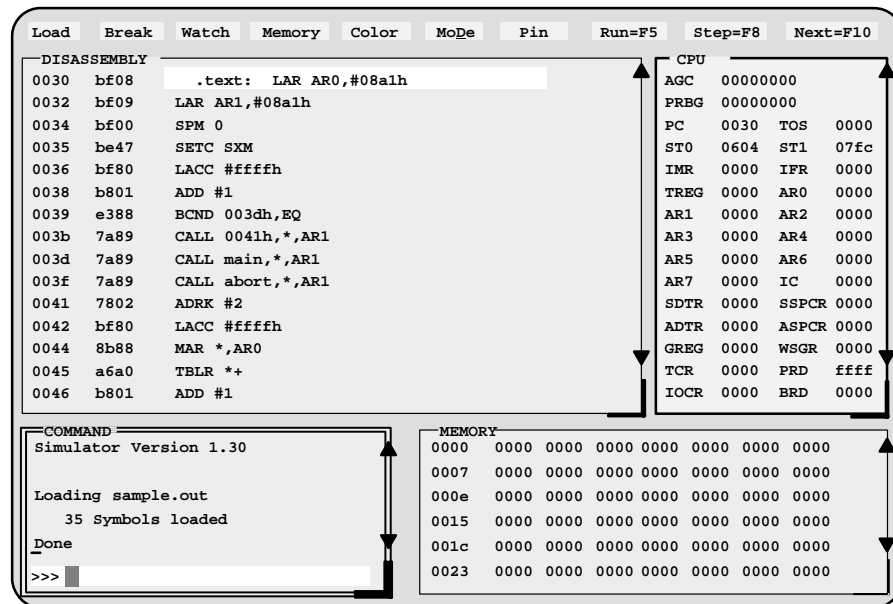
```
source ~/.cshrc 
```

3.4 Step 3: Verifying the Installation

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

sim2xx sample 

You should see a display similar to this one:



- ☐ If you see a display similar to this one, you have correctly installed your simulator and debugger.
- ☐ If you don't see a similar display, then your debugger or simulator may not be installed properly. Go back through the installation instructions and be sure that you have followed each step correctly; then reenter the command above.

3.5 Using the Debugger With the X Window System

If you're using the X Window System to run the 'C2xx debugger, you need to know about the keyboard's special keys, the debugger fonts, and using the debugger on a monochrome monitor.

Using the keyboard's special keys

The debugger uses some special keys that you can map differently from your particular keyboard. Some keyboards may have these special symbols on separate keys. Other keyboards do not have the special keys.

The special keys that the debugger uses are shown in the following table with their corresponding keysym. A *keysym* is a label that interprets a keystroke; it allows you to modify the action of a key on the keyboard.

Key	Keysym
(F1) to (F10)	F1 to F10
(PAGE UP)	Prior
(PAGE DOWN)	Next
(HOME)	Home
(END)	End
(INSERT)	Insert
→	Right
←	Left
↑	Up
↓	Down

Use the X utility `xev` to check the keysyms that are associated with your keyboard. If you need to change the keysym definitions, use the `xmodmap` utility. For example, you could create a file that contains the following commands and use that file with `xmodmap` to change a keyboard to match the keys listed above:

```
keysym R13      = End
keysym Down     = Down
keysym F35      = Next
keysym Left     = Left
keysym Right    = Right
keysym F27      = Home
keysym Up       = Up
keysym F29      = Prior
keysym Insert   = Insert
```

Refer to your X Window System documentation for more information about using `xev` and `xmodmap`.

Changing the debugger font

You can change the font of the debugger screen by using the `xrdb` utility and modifying the `.Xdefaults` file in your root directory. For example, to change the fonts of the 'C2xx debugger to Courier, add the following line to the `.Xdefaults` file:

```
sim2xx*font:courier
```

For more information about using `xrdb` to change the font, refer to your X Window System documentation.

Color mappings on monochrome screens

Although a color monitor is recommended, the following table shows the color mappings for monochrome screens:

Color	Appearance on Monochrome Screen
black	black
blue	black
green	white
cyan	white
red	black
magenta	black
yellow	white
white	white

Index

A

- arrow keys
 - HP systems 3-9
 - SPARC systems 2-9
- assembler
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
- autoexec.bat file
 - environmental variables 1-5
 - sample 1-7

B

- b debugger option
 - HP systems 3-6
 - PC systems 1-5
 - SPARC systems 2-6
- batch files
 - autoexec.bat 1-7
 - .cshrc
 - HP systems 3-6 to 3-7
 - SPARC systems 2-6 to 2-7
 - init.clr
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
 - initdb.bat 1-6 to 1-7
 - initialization 2-3
 - mono.clr
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
 - sim2xx.cmd
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3

- batch files (continued)
 - siminit.cmd
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
- bb debugger option
 - HP systems 3-6
 - PC systems 1-5
 - SPARC systems 2-6

C

- CD-ROM
 - mounting
 - HP systems 3-4
 - SPARC systems 2-4
 - requirements
 - HP systems 3-2
 - SPARC systems 2-2
 - retrieving files from
 - HP systems 3-4
 - SPARC systems 2-5
 - unmounting
 - HP systems 3-5
 - SPARC systems 2-5
- color mapping
 - HP systems 3-10
 - SPARC systems 2-10
- compiler
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
- .cshrc file
 - contents
 - HP systems 3-6 to 3-7
 - SPARC systems 2-6 to 2-7
 - invoking
 - HP systems 3-7
 - SPARC systems 2-7

- .cshrc file (continued)
 - sample
 - HP systems* 3-7
 - SPARC systems* 2-7
- customizing the display
 - HP systems* 3-3, 3-10
 - PC systems* 1-2, 1-3
 - SPARC systems* 2-3, 2-10

D

- D_DIR environment variable
 - HP systems* 3-6
 - PC systems* 1-5
 - SPARC systems* 2-6
- D_OPTIONS environment variable
 - HP systems* 3-6
 - PC systems* 1-5
 - SPARC systems* 2-6
- D_SRC environment variable
 - HP systems* 3-6
 - PC systems* 1-5
 - SPARC systems* 2-6
- debugger
 - environment setup
 - HP systems* 3-6 to 3-7
 - PC systems* 1-5 to 1-7
 - SPARC systems* 2-6 to 2-7
 - font changes
 - HP systems* 3-10
 - SPARC systems* 2-10
 - installation
 - procedure*
 - HP systems* 3-4
 - PC systems* 1-4
 - SPARC systems* 2-4
 - verification*
 - HP systems* 3-8
 - PC systems* 1-8
 - SPARC systems* 2-8
 - Windows 1-4, 1-9
 - X Window System
 - HP systems* 3-9 to 3-10
 - SPARC systems* 2-9 to 2-10
- default
 - memory map
 - HP systems* 3-3
 - PC systems* 1-3
 - SPARC systems* 2-3

- default (continued)
 - screen configuration file
 - HP systems* 3-3
 - PC systems* 1-3
 - SPARC systems* 2-3
- directories
 - auxiliary files
 - HP systems* 3-6
 - PC systems* 1-5
 - SPARC systems* 2-6
 - debugger software
 - HP systems* 3-4, 3-6
 - PC systems* 1-4, 1-5
 - SPARC systems* 2-5, 2-6
 - identifying additional source directories
 - HP systems* 3-6
 - PC systems* 1-5
 - SPARC systems* 2-6
 - sim2xx
 - HP systems* 3-4, 3-6
 - PC systems* 1-4, 1-5
 - SPARC systems* 2-5, 2-6
- disk space requirements
 - HP systems* 3-2
 - SPARC systems* 2-2
- display
 - font changes
 - HP systems* 3-10
 - SPARC systems* 2-10
 - requirements
 - HP systems* 3-2
 - PC systems* 1-2
 - SPARC systems* 2-2
- DISPLAY environment variable
 - HP systems* 3-6
 - SPARC systems* 2-6

E

- end key
 - HP systems* 3-9
 - SPARC systems* 2-9
- environment variables
 - HP systems* 3-6
 - PC systems* 1-5
 - SPARC systems* 2-6

F

- font changes
 - HP systems 3-10
 - SPARC systems 2-10
- font debugger option 1-5
- function key mapping
 - HP systems 3-9
 - SPARC systems 2-9

G

- graphics card requirements 1-2

H

- hardware checklist
 - HP systems 3-2
 - PC systems 1-2
 - SPARC systems 2-2
- home key
 - HP systems 3-9
 - SPARC systems 2-9
- host
 - HP systems 3-2
 - PC systems 1-2
 - SPARC systems 2-2

I

- i debugger option
 - HP systems 3-6
 - PC systems 1-5
 - SPARC systems 2-6
- init.clr file
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
- initdb.bat file 1-6
- initialization batch files
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
- insert key
 - HP systems 3-9
 - SPARC systems 2-9

- installation
 - debugger
 - HP systems 3-4
 - PC systems 1-4
 - SPARC systems 2-4
 - simulator
 - HP systems 3-4
 - PC systems 1-4
 - SPARC systems 2-4
 - verifying
 - HP systems 3-8
 - PC systems 1-8
 - SPARC systems 2-8

K

- keyboard mapping
 - HP systems 3-9
 - SPARC systems 2-9
- keysym label
 - HP systems 3-9
 - SPARC systems 2-9

L

- linker
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3

M

- machine to display on
 - HP systems 3-6
 - SPARC systems 2-6
- mapping keys
 - HP systems 3-9
 - SPARC systems 2-9
- memory
 - default map
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
 - mapping
 - HP systems 3-3
 - PC systems 1-3
 - SPARC systems 2-3
 - requirements 1-2

mono.clr file
 HP systems 3-3
 PC systems 1-3
 SPARC systems 2-3

monochrome monitors color mapping
 HP systems 3-10
 SPARC systems 2-10

mouse requirements
 HP systems 3-2
 PC systems 1-2
 SPARC systems 2-2

O

operating system
 HP systems 3-3
 PC systems 1-3
 SPARC systems 2-3

optional files
 HP systems 3-3
 PC systems 1-3
 SPARC systems 2-3

P

-p debugger option 1-5

page-down key
 HP systems 3-9
 SPARC systems 2-9

page-up key
 HP systems 3-9
 SPARC systems 2-9

PATH statement 1-5

permissions
 HP systems 3-3
 SPARC systems 2-3

-profile debugger option
 HP systems 3-6
 PC systems 1-5
 SPARC systems 2-6

R

retrieving files from CD-ROM
 HP systems 3-4
 SPARC systems 2-5

root privileges
 HP systems 3-3
 SPARC systems 2-3

S

-s debugger option
 HP systems 3-6
 PC systems 1-5
 SPARC systems 2-6

screen configurations
 HP systems 3-3
 PC systems 1-3
 SPARC systems 2-3

setpath statement
 HP systems 3-6
 SPARC systems 2-6

sim2xx
 command options
 HP systems 3-6
 PC systems 1-5
 SPARC systems 2-6
 directory
 HP systems 3-4, 3-6
 PC systems 1-4, 1-5
 SPARC systems 2-5, 2-6
 verifying the installation
 HP systems 3-8
 PC systems 1-8
 SPARC systems 2-8

sim2xx.cmd file
 HP systems 3-3
 PC systems 1-3
 SPARC systems 2-3

siminit.cmd file
 HP systems 3-3
 PC systems 1-3
 SPARC systems 2-3

simulator
 environment setup
 HP systems 3-6 to 3-7
 PC systems 1-5 to 1-7
 SPARC systems 2-6 to 2-7
 installation
 procedure
 HP systems 3-4
 PC systems 1-4
 SPARC systems 2-4

simulator installation (continued)

verification

- HP systems 3-8
- PC systems 1-8
- SPARC systems 2-8

software

checklist

- HP systems* 3-3
- PC systems* 1-3
- SPARC systems* 2-3

tools

- HP systems* 3-3
- PC systems* 1-3
- SPARC systems* 2-3

special keys

- HP systems 3-9
- SPARC systems 2-9

T

-t debugger option

- HP systems 3-6
- PC systems 1-5
- SPARC systems 2-6

U

utilities

xev

- HP systems* 3-9
- SPARC systems* 2-9

xmodmap

- HP systems* 3-9
- SPARC systems* 2-9

xrdb

- HP systems* 3-10
- SPARC systems* 2-10

V

-v debugger option

- HP systems 3-6
- PC systems 1-5
- SPARC systems 2-6

verifying the installation

- HP systems 3-8
- PC systems 1-8
- SPARC systems 2-8

W

Windows 1-9

X

-x debugger option

- HP systems 3-6
- PC systems 1-5
- SPARC systems 2-6

X Window System

- HP systems 3-9 to 3-10
- SPARC systems 2-9 to 2-10

.Xdefaults file

- HP systems 3-10
- SPARC systems 2-10

xev utility

- HP systems 3-9
- SPARC systems 2-9

xmodmap utility

- HP systems 3-9
- SPARC systems 2-9

xrdb utility

- HP systems 3-10
- SPARC systems 2-10

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.