

TMS320C3x Simulator

*Getting
Started*



TMS320C3x Simulator Getting Started



IMPORTANT NOTICE

Texas Instruments Incorporated (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 current specifications 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.

Please be aware that TI products are not intended for use in life-support appliances, devices, or systems. Use of TI product in such applications requires the written approval of the appropriate TI officer. Certain applications using semiconductor devices may involve potential risks of personal injury, property damage, or loss of life. In order to minimize these risks, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards. Inclusion of TI products in such applications is understood to be fully at the risk of the customer using TI devices or systems.

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

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

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

OpenWindows and SunOS are trademarks of Sun Microsystems, Inc.

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'll 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'll Need	1-2
	Hardware checklist	1-2
	Software checklist	1-2
1.2	Step 1: Installing the Simulator and Debugger Software	1-3
1.3	Step 2: Setting Up the Debugger Environment	1-4
	Invoking the new or modified batch file	1-5
	Modifying the PATH statement	1-5
	Setting up the environment variables	1-6
1.4	Step 3: Verifying the Installation	1-7
1.5	Using the Simulator With MS-Windows	1-8
2	Installing the Simulator and C Source Debugger With SunOS	2-1
	<i>Lists the hardware and software you'll need to install the simulator and C source debugger; provides installation instructions for SPARC systems running SunOS.</i>	
2.1	What You'll Need	2-2
	Hardware checklist	2-2
	Software checklist	2-2
2.2	Step 1: Installing the Simulator and Debugger Software	2-3
2.3	Step 2: Setting Up the Debugger Environment	2-3
	Modifying the path shell variable	2-3
	Setting up the environment variables	2-4
	Reinitializing your shell	2-5
2.4	Step 3: Verifying the Installation	2-6
2.5	Restrictions Associated With the SPARC Version of the Simulator	2-7
2.6	Using the Debugger With the X Window System	2-7
	Using the keyboard's special keys	2-7
	Changing the debugger font	2-8
	Color mappings on monochrome screens	2-8

Installing the Simulator and C Source Debugger With DOS

This chapter helps you install the 'C3x simulator and the C source debugger on PC systems 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
1.1 What You'll Need	1-2
Hardware checklist	1-2
Software checklist	1-2
1.2 Step 1: Installing the Simulator and Debugger Software	1-3
1.3 Step 2: Setting Up the Debugger Environment	1-4
Invoking the new or modified batch file	1-5
Modifying the PATH statement	1-5
Setting up the environment variables	1-6
1.4 Step 3: Verifying the Installation	1-7
1.5 Using the Simulator With MS-Windows	1-8

1.1 What You'll Need

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

Hardware checklist

- host** An IBM PC/AT or 100% compatible ISA/EISA-based PC with a hard-disk system and a 1.2-Mbyte floppy-disk drive
- memory** Minimum of 640K bytes; in addition, if you are running under MS-Windows, you'll need at least 256K bytes of extended memory
- display** Monochrome or color (color recommended)
- optional hardware** A Microsoft-compatible mouse
- 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 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 section in Chapter 1, *Overview of a Code Development and Debugging System*, in the *TMS320C3x C Source Debugger User's Guide*.)
- miscellaneous materials** Blank, formatted disks

Software checklist

- operating system** MS-DOS or PC-DOS (version 3.0 or later)
Optional: MS-Windows (version 3.0 or later)
- software tools** TMS320 floating-point family DSP ('C3x/'C4x) assembler and linker
Optional: TMS320C3x/C4x C compiler
- optional file** † *siminit.cmd* is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a 'C3x memory map. If this file isn't present when you 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 Chapter 5, *Defining a Memory Map*, in the *TMS320C3x C Source Debugger User's Guide*.

† Included as part of the debugger package

1.2 Step 1: Installing the Simulator and Debugger Software

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

- 1) Make a backup copy of the product disk. (If necessary, refer to the manual that came with your computer. Note that the DOS product disk includes both the DOS and MS-Windows version of the debugger executable).
- 2) On your hard disk or system disk, create a directory named *sim3x*. This directory will contain the 'C3x C source debugger software. To create this directory, enter:

```
MD C:\sim3x
```

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

```
COPY A:\*.* C:\sim3x\*.* /V
```

The DOS version of the debugger executable is called *sim3x.exe*, and the MS-Windows version of the debugger executable is called *sim3xw.exe*. Throughout this document, the executable for the debugger is referred to as simply *sim3x*.

1.3 Step 2: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- Modify the PATH statement to identify the sim3x directory.
- Define environment variables so that the debugger can find the files it needs.



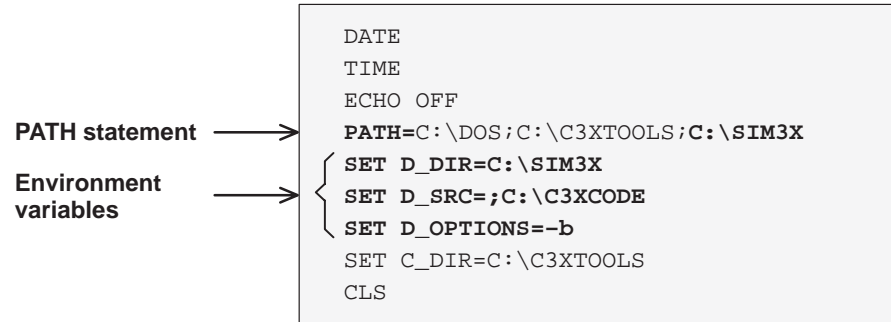
Not only must you do these things before you invoke the debugger for the first time, *you must do them any time you power up or reboot your PC.*

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 system's 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.

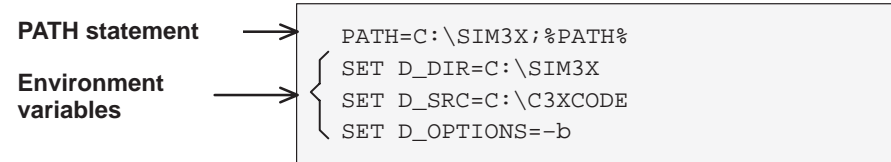
Figure 1–1 (a) shows an example of an autoexec.bat file that contains the suggested modifications (highlighted in bold type). Figure 1–1 (b) shows a sample batch file that you could create instead of editing the autoexec.bat file (for the purpose of discussion, assume that this sample file is named *initdb.bat*). The subsections following the figure explain these modifications.

Figure 1–1. DOS-Command Setup for the Debugger

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



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



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:\SIM3X
```

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:\sim3x at the end of the statement as shown in Figure 1-1 (a).
- If you are creating an initdb.bat file, use a different format for the PATH statement:

```
PATH=C:\SIM3X;%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 `sim3x` directory:

```
SET D_DIR=C:\SIM3X
```

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

This directory contains auxiliary files (such as `siminit.cmd`) 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       -mvversion
-profile     -s           -t filename       -v
-mm
```

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 Chapter 1, *Overview of a Code Development and Debugging System*, in the *TMS320C3x C Source Debugger User's Guide*.

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:

```
sim3x c:\sim3x\sample
```

You should see a display similar to this one:

The screenshot displays the TMS320C3x debugger interface with the following sections:

- DISASSEMBLY:** A table of assembly instructions with addresses, hex values, mnemonics, and operands.

Address	Hex	Mnemonic	Operands
f00075	00f000b2	ABSI	178,DP
f00076	087000f0	LDI	240,DP
f00077	08340074	LDI	@074H,SP
f00078	080b0014	LDI	SP,AR3
f00079	087000f0	LDI	240,DP
f0007a	08280075	LDI	@075H,AR0
f0007b	04e8ffff	CMPI	-1,AR0
f0007c	6a05000c	BZ	f00089
f0007d	08412001	LDI	*AR0++(1),R1
f0007e	6a250008	BZD	f00089
f0007f	08492001	LDI	*AR0++(1),AR1
f00080	08402001	LDI	*AR0++(1),R0
f00081	18610001	SUBI	1,R1
f00082	139b9991	RPTS	R1
f00083	da002120	LDI	*AR0++(1),R0 STI
- CPU:** A table of CPU registers and their values.

Register	Value	Register	Value
PC	00f00076	SP	00000755
R0	00000003	R1	00000005
R2	00000007	R3	00000000
R4	00000000	R5	00000000
R6	00000000	R7	00000000
AR0	00001802	AR1	00000000
AR2	00000000	AR3	00000000
AR4	00000000	AR5	00000000
AR6	00000000	AR7	00000000
IR0	00000000	IR1	00000000
ST	00000000	RC	00000000
RS	00000000	RE	00000000
DP	00000000	BK	00000000
IE	00000000	IF	00000000
- COMMAND:** A text window showing the debugger's startup sequence:


```
TMS3203x, Debugger Version 4.60
Copyright (c) 1989, 1993 Texas In
TMS320C3x
Loading sample.out
Done
>>>
```
- MEMORY:** A table of memory addresses and their values.

Address	Value	Address	Value	Address	Value
000000	0000004b	00000040	00000041	00000042	
000004	00000043	00000044	00000045	00000046	
000008	00000047	00000048	00000049	0000004a	
00000c	00000000	00000000	00000000	00000000	
000010	00000000	00000000	00000000	00000000	
000014	00000000	00000000	00000000	00000000	

- If you see a display similar to this one, you have correctly installed your simulator and debugger.
- If you don't see a 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 re-enter the command above.

1.5 Using the Simulator 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 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 MS-Windows environment. Refer to your MS-Windows manual for details.

You should run MS-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 helps you install the 'C3x simulator and the C source debugger on a SPARCstation running OpenWindows under SunOS version 4.1.x, including Solaris 2.x. When you complete the installation, turn to the *TMS320C3x C Source Debugger User's Guide*.

Topic	Page
2.1 What You'll Need	2-2
Hardware checklist	2-2
Software checklist	2-2
2.2 Step 1: Installing the Simulator and Debugger Software	2-3
2.3 Step 2: Setting Up the Debugger Environment	2-3
Modifying the path shell variable	2-3
Setting up the environment variables	2-4
Reinitializing your shell	2-5
2.4 Step 3: Verifying the Installation	2-6
2.5 Restrictions Associated With the SPARC Version of the Simulator	2-7
2.6 Using the Debugger With the X Window System	2-7
Using the keyboard's special keys	2-7
Changing the debugger font	2-8
Color mappings on monochrome screens	2-8

2.1 What You'll Need

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

Hardware checklist

- host** A SPARCstation with a cartridge tape drive
- display** Monochrome or color monitor (color recommended)

Software checklist

- operating system** SunOS 4.1 or higher (running OpenWindows 2.0 or higher)
- software tools** TMS320 floating-point family DSP ('C3x/'C4x) assembler and linker
Optional: TMS320C3x/C4x C compiler
- optional file** † *siminit.cmd* is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a 'C3x memory map. If this file isn't present when you 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 Chapter 5, *Defining a Memory Map*, in the *TMS320C3x C Source Debugger User's Guide*.

† Included as part of the debugger package

2.2 Step 1: Installing the Simulator and Debugger Software

This section explains the process of installing the simulator and debugger on a hard-disk system. The software package is shipped on a cartridge tape. To install the simulator and debugger, you must restore the directory from the tape.

- 1) Insert the product tape in a cartridge tape drive.
- 2) Create a directory named *sim3x* to contain the 'C3x simulator and debugger software:

```
mkdir sim3x
```

- 3) Make *sim3x* the current directory:

```
cd sim3x
```

- 4) Copy the files from tape to disk:

```
tar xvf /dev/rst8
```

2.3 Step 2: Setting Up the Debugger Environment

To ensure that the tools work correctly, you must:

- Modify the path shell variable to include the *sim3x* directory.
- Define environment variables so that the debugger can find the files it needs.
- Reinitialize your shell.

Modifying the path shell variable

You must include the debugger directory in your shell path. To do this, you must modify your shell configuration file in your home directory (for example, the *.cshrc* file for a C shell). This file must include the pathname to your *sim3x* directory in your path if it is not already there. The following statement is an example of what a typical path-variable definition looks like:

```
set path = ( . /bin /usr/ucb /usr/contrib/bin /usr/bin \  
/usr/openwin/bin)
```

The following is an example of a modified path variable. The part of the path that is boldface is an example of a pathname that identifies the *sim3x* directory:

```
set path = ( . /bin /usr/ucb /usr/contrib/bin /usr/bin \  
/usr/openwin/bin /user/fred/sim3x)
```

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 four environment variables, named `D_DIR`, `D_SRC`, `D_OPTIONS`, and `DISPLAY` (X Window System only). The next four steps tell you how to set up these environment variables; these steps can be performed in your shell configuration file.

- ❑ Set up the `D_DIR` environment variable to identify the `sim3x` directory by defining the `D_DIR` environment variable like the following:

```
setenv D_DIR "/user/fred/sim3x"
```

(Be sure to enclose the directory name within quotes.)

This directory contains auxiliary files (such as `siminit.cmd`) 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 access from the debugger. The general format for doing this is:

```
setenv D_SRC "pathname1;pathname2..."
```

(Be sure to enclose the path names within one set of quotes.)

For example, if your C3x programs were in a directory named `/user/fred/c3xsource`, the `D_SRC` setup would be:

```
setenv D_SRC "/user/fred/c3xsource"
```

- ❑ 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:

```
setenv D_OPTIONS "[object filename] [debugger options]"
```

(Be sure to enclose the options and filenames within one set of quotes.)

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`:

<code>-b</code>	<code>-bb</code>	<code>-i pathname</code>	<code>-mvversion</code>
<code>-profile</code>	<code>-s</code>	<code>-t filename</code>	<code>-v</code>
<code>-mm</code>			

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 Chapter 1, *Overview of a Code Development and Debugging System*, in the *TMS320C3x C Source Debugger User's Guide*.

- If you are using the X Window system, you can use the DISPLAY environment variable to display the debugger on a different machine than the one the debugger is running on. The general format for doing this is:

setenv DISPLAY "machine name"

For example, if you are running the debugger on a machine called opie and you want the 'C3x debugger display to appear on a machine called barney, the DISPLAY setup would be:

```
setenv DISPLAY barney:0
```

You can also display the debugger on a different machine by using the -d option when invoking the debugger.

```
sim3x -d barney:0
```

For more information about using the debugger under the X Window system, refer to Section 2.6, *Using the Debugger With the X Window System*.

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:

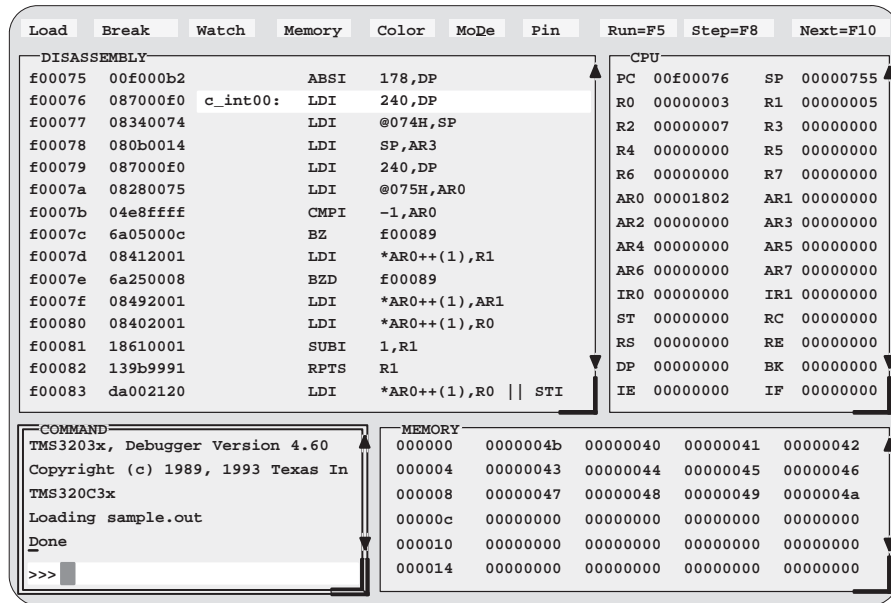
```
source ~/.cshrc
```

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:

`sim3x 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 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 re-enter the command above.

2.5 Restrictions Associated With the SPARC Version of the Simulator

Some restrictions are associated with the SPARC version of the simulator and debugger interface. These restrictions, listed below, override the information presented in Parts II and III of the *TMS320C3x C Source Debugger User's Guide*.

The C source debugger has a very flexible command-entry system; there are usually a variety of ways to perform any specific action. For example, you may be able to perform the same action by typing in a command, using the mouse, or using a function key. However, the alternate-key sequences described in the *TMS320C3x C Source Debugger User's Guide* apply to PCs. No keyboard mapping is provided for SPARC systems. Therefore, you should look for methods that use the mouse, a function key, or a command.

2.6 Using the Debugger With the X Window System

If you're using the X Window System to run the 'C3x 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 fonts of the 'C3x debugger to Courier, add the following line to the `.Xdefaults` file:

```
sim3x*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 2-7
- assembler 1-2, 2-2
- autoexec.bat file 1-4 to 1-6
 - invoking 1-5
 - sample 1-4

B

- b debugger option
 - with D_OPTIONS environment variable 1-6, 2-4
- batch files
 - autoexec.bat 1-4 to 1-6
 - .cshrc 2-3 to 2-5
 - initdb.bat 1-4 to 1-6
 - initialization, siminit.cmd
 - PC systems 1-2
 - SPARC systems 2-2
 - invoking
 - autoexec.bat 1-5
 - .cshrc 2-5
 - initdb.bat 1-5
 - siminit.cmd
 - PC systems 1-2
 - SPARC systems 2-2

C

- c3xhll directory 1-3
- changing the debugger display (font) 2-8
- colors
 - mapping with X Windows 2-8
- compiler 1-2, 2-2
- .cshrc file 2-3 to 2-5
 - invoking 2-5

- customizing the display
 - changing the font 2-8

D

- D_DIR environment variable 1-6, 2-4
- D_OPTIONS environment variable 1-6, 2-4
- D_SRC environment variable 1-6, 2-4
- debugger
 - changing the displayed font 2-8
 - environment setup 1-4 to 1-6, 2-3 to 2-5
 - installation
 - simulator
 - PC systems 1-1 to 1-8
 - SPARC systems 2-1 to 2-8
 - verifying 1-7, 2-5 to 2-6
 - using the X Window System 2-7 to 2-8
 - using with MS-Windows 1-3, 1-8
- default
 - memory map
 - PC systems 1-2
 - SPARC systems 2-2
- directories
 - c3xhll directory 1-3
 - for auxiliary files 1-6, 2-4
 - for debugger software 1-3, 2-3
 - PC systems 1-5
 - SPARC systems 2-3
 - identifying additional source directories 1-6, 2-4
 - sim3x directory
 - PC systems 1-6
 - SPARC systems 2-3, 2-4
- display requirements
 - PC systems 1-2
 - SPARC systems 2-2
- DOS
 - setting up debugger environment 1-4 to 1-6

E

- end key 2-7
- environment variables
 - D_DIR 1-6, 2-4
 - D_OPTIONS 1-6, 2-4
 - D_SRC 1-6, 2-4
 - for debugger options 1-6, 2-4
 - identifying auxiliary directories 1-6, 2-4
 - identifying source directories 1-6, 2-4

F

- F1 key
 - mapping 2-7
- F2 key
 - mapping 2-7
- F3 key
 - mapping 2-7
- F4 key
 - mapping 2-7
- F5 key
 - mapping 2-7
- F6 key
 - mapping 2-7
- F7 key
 - mapping 2-7
- F8 key
 - mapping 2-7
- F9 key
 - mapping 2-7
- F10 key
 - mapping 2-7
- font
 - changing the debugger display 2-8

G

- graphics card requirements 1-2

H

- hardware checklist
 - PC systems 1-2
 - SPARC systems 2-2

Index-2

- home key 2-7
- host system
 - PC systems 1-2
 - SPARC systems 2-2

I

- i debugger option
 - with D_OPTIONS environment variable 1-6, 2-4
- initdb.bat file 1-4 to 1-6
 - invoking 1-5
 - sample 1-4
- initialization batch files
 - siminit.cmd
 - PC systems 1-2
 - SPARC systems 2-2
- insert key 2-7
- installation
 - debugger software
 - PC systems 1-3
 - SPARC systems 2-3
 - verifying
 - PC systems 1-7
 - SPARC systems 2-6
- invoking
 - autoexec.bat file 1-5
 - .cshrc file 2-5
 - initdb.bat file 1-5

K

- key sequences, restrictions
 - SPARC systems 2-7
- keyboard
 - mapping keys 2-7 to 2-8
- keys
 - special keys with the X Window System 2-7 to 2-8
- keysym label 2-7 to 2-8

L

- labels
 - keysym 2-7 to 2-8
- linker 1-2, 2-2

M

- mapping keys for use with X Windows 2-7 to 2-8
- memory
 - default map
 - PC systems* 1-2
 - SPARC systems* 2-2
 - mapping
 - PC systems* 1-2
 - siminit.cmd file*
 - PC systems 1-2
 - SPARC systems 2-2
 - SPARC systems* 2-2
 - requirements 1-2
- mm debugger option
 - with `D_OPTIONS` environment variable 1-6, 2-4
- monochrome monitors, color mapping with X Windows 2-8
- mouse
 - requirements
 - PC systems* 1-2
- MS-Windows
 - using with the debugger 1-3, 1-8
- mv debugger option
 - with `D_OPTIONS` environment variable 1-6, 2-4

O

- operating system
 - PC systems 1-2
 - SPARC systems 2-2
- optional files
 - PC systems 1-2
 - SPARC systems 2-2

P

- page-up/page-down keys 2-7 to 2-8
- PATH statement 1-5, 2-3
- profile debugger option
 - with `D_OPTIONS` environment variable 1-6, 2-4

R

- required tools
 - PC systems 1-2
 - SPARC systems 2-2
- restrictions
 - SPARC systems 2-6

S

- s debugger option
 - with `D_OPTIONS` environment variable 1-6, 2-4
- sim3x command
 - options
 - D_OPTIONS environment variable* 1-6, 2-4
 - verifying the installation 1-7, 2-5 to 2-6
- sim3x directory
 - PC systems 1-6
 - SPARC systems 2-3, 2-4
- siminit.cmd file
 - PC systems 1-2
 - SPARC systems 2-2
- simulator
 - PC systems
 - additional tools* 1-2
 - debugger environment* 1-4 to 1-6
 - debugger installation* 1-1 to 1-8
 - verifying 1-7
 - host system* 1-2
 - installation*
 - software 1-3
 - verifying 1-7
 - operating system* 1-2
 - requirements*
 - display 1-2
 - graphics card 1-2
 - hardware 1-2
 - memory 1-2
 - mouse 1-2
 - software 1-2
 - restrictions
 - color displays, SPARC systems* 2-7
 - keyboard mapping, SPARC systems* 2-7
 - SPARC systems* 2-7

SPARC systems

- additional tools* 2-2
- debugger environment* 2-3 to 2-5
- debugger installation* 2-1 to 2-8
 - verifying 2-6
- host system* 2-2
- installation*
 - software 2-3
 - verifying 2-6
- operating system* 2-2
- requirements*
 - display 2-2
 - hardware 2-2
 - software 2-2
- restrictions* 2-7

software checklist

- PC systems 1-2
- SPARC systems 2-2

special keys

- X Window System 2-7 to 2-8

T

- t debugger option
 - with D_OPTIONS environment variable 1-6, 2-4

U

utilities

- xev 2-7 to 2-8
- xmodmap 2-7 to 2-8
- xrdb 2-8

V

- v debugger option
 - with D_OPTIONS environment variable 1-6, 2-4
- verifying
 - installation
 - PC systems 1-7
 - SPARC systems 2-6

X

- x debugger option 1-6, 2-4
- X Window System
 - changing the displayed font 2-8
 - color mapping 2-8
 - special keys 2-7 to 2-8
 - using with the debugger 2-7 to 2-8
 - xev utility 2-7 to 2-8
 - xmodmap utility 2-7 to 2-8
- .Xdefaults file
 - changing the displayed debugger font 2-8
- xev utility 2-7 to 2-8
- xmodmap utility 2-7 to 2-8