TMS320C54x Simulator
Getting Started
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.

Copyright © 1996, Texas Instruments Incorporated
TRADEMARKS

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

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

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

SunOS and OpenWindows 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.
1 Installing the Simulator and C Source Debugger With DOS

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.

1.1 What You’ll 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
   Invoking the new or modified batch file ........................................... 1-6
   Modifying the PATH statement ................................................ 1-6
   Setting up the environment variables .................................... 1-7
1.4 Step 3: Verifying the Installation .......................................................... 1-8
1.5 Using the Simulator With MS-Windows ..................................................... 1-9

2 Installing the Simulator and C Source Debugger With SunOS

Lists the hardware and software you’ll need to install the simulator and C source debugger; provides installation instructions for SPARC systems running SunOS or Solaris.

2.1 What You’ll 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
   Modifying the path shell variable ........................................... 2-6
   Setting up the environment variables .................................... 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
### Contents

3 Installing the Simulator and C Source Debugger With HP-UX ....................... 3-1

Lists the hardware and software you’ll need to install the simulator and C source debugger; provides installation instructions for HP systems running HP-UX.

3.1 What You’ll 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 .................................................. 3-4
3.3 Step 2: Setting Up the Debugger Environment .......................................... 3-5
   Modifying the path shell variable ................................ .... 3-5
   Setting up the environment variables .................................... 3-5
   Reinitializing your shell .................................................... 3-6
3.4 Step 3: Verifying the Installation .................................................. 3-7
3.5 Using the Debugger With the X Window System ......................................... 3-8
   Using the keyboard’s special keys ................................ .... 3-8
   Changing the debugger font ............................................... 3-9
   Color mappings on monochrome screens ................................. 3-9
This chapter provides instructions to help you install the TMS320C54x 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 TMS320C5xx C Source Debugger User’s Guide.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 What You’ll Need</td>
<td>1-2</td>
</tr>
<tr>
<td>1.2 Step 1: Installing the Simulator and Debugger Software</td>
<td>1-4</td>
</tr>
<tr>
<td>1.3 Step 2: Setting Up the Debugger Environment</td>
<td>1-5</td>
</tr>
<tr>
<td>1.4 Step 3: Verifying the Installation</td>
<td>1-8</td>
</tr>
<tr>
<td>1.5 Using the Simulator With MS-Windows</td>
<td>1-9</td>
</tr>
</tbody>
</table>
What You'll Need

1.1 What You'll Need

The following checklists detail items that are shipped with the 'C54x C source debugger and simulator and any 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 1.2M floppy-disk drive; a 386 or higher is highly recommended

- **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 monitor (color recommended)

- **optional hardware**
  A Microsoft-compatible mouse

- **miscellaneous materials**
  Blank, formatted disks
What You’ll Need

Software checklist

- **operating system**
  - MS-DOS or PC-DOS (version 3.0 or later)
  - Optional: MS-Windows (version 3.0 or later)

- **software tools**
  - TMS320C54x assembler and linker
  - Optional: TMS320C54x C compiler

- **optional files**
  - *siminit.cmd* is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a 'C54x 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 defining your own memory map, refer to the Defining A Memory Map chapter in the TMS320C5xx C Source Debugger User’s Guide.

-  † *sim541.cmd, sim542.cmd, sim543.cmd, sim544.cmd, sim545.cmd, sim546.cmd, sim547.cmd, and sim549.cmd* are batch files containing commands that configure a memory map. Each file simulates a different device—'C541, 'C542, 'C543, 'C544, 'C545, 'C546, 'C547, or 'C549.

-  † *init.clr* is a general-purpose screen configuration file. If init.clr isn’t present when you invoke the debugger, the debugger uses the default screen configuration. If you don’t have a color monitor, *mono.clr* is included for monochrome monitors.

-  † *init.25, init.43, and init.50* have been provided for basic 80 x 25, 80 x 43, and 80 x 50 screen sizes, respectively. The init.clr file brings up the debugger in 80 x 25 mode. To bring the debugger up in another mode, copy one of the init.xx files to the init.clr file. When you first invoke 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 Customizing the Debugger Display chapter in the TMS320C5xx 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 how to install the simulator and debugger on a hard-disk system.

1) Make a backup copy of each 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 sim5xx. This directory will contain the 'C54x software.

   MD C:\sim5xx

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

   COPY A:\*.* C:\sim5xx\*.* /V

   The DOS version of the debugger executable is called sim5xx.exe, and the MS-Windows version of the debugger executable is called sim5xxw.exe. Throughout this document, the executable for the debugger is referred to simply as sim5xx.
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 sim5xx directory.
- Define environment variables so that the debugger can find the files it needs.

**Note:**

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

```plaintext
DATE
TIME
ECHO OFF
PATH=
C:\DOS;C:\c5xxtool;C:\sim5xx
SET D_DIR=C:\sim5xx
SET D_SRC=C:\c5xxcode
SET D_OPTIONS=-b
CLS
```

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

```plaintext
PATH=C:\sim5xx;%PATH%
SET D_DIR=C:\sim5xx
SET D_SRC=C:\c5xxcode
SET D_OPTIONS=-b
```
Step 2: Setting Up the Debugger Environment

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:\sim5xx```

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:\sim5xx 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:\sim5xx;%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).
**Step 2: Setting Up the Debugger Environment**

**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.

1. Set up the D_DIR environment variable to identify the sim5xx directory:
   
   ```
   SET D_DIR=C:\sim5xx
   ```
   
   (Be careful not to precede the equal sign with a space.)

   This directory contains auxiliary files (such as siminit.cmd) that the debugger needs.

2. 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 `C54x programs were in a directory named csource on drive C, the D_SRC setup would be:

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

3. 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[b] –d machinename –i pathname –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 *Overview of a Code Development and Debugging System* chapter of the *TMS320C5xx C Source Debugger User’s Guide*.

---

*Installing the Simulator and C Source Debugger With DOS*  
1-7
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:

```
sim5xx c:\sim5xx\sample
```

You should see a display similar to this one:

![Disassembly and Memory Display]

- 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 reenter 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 provides instructions to help you install the 'C54x simulator and the C source debugger on a SPARCstation running SunOS™ or Solaris™. When you complete the installation, turn to the TMS320C5xx C Source Debugger User’s Guide.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 What You’ll Need</td>
<td>2-2</td>
</tr>
<tr>
<td>2.2 Step 1: Installing the Simulator and Debugger Software</td>
<td>2-4</td>
</tr>
<tr>
<td>2.3 Step 2: Setting Up the Debugger Environment</td>
<td>2-6</td>
</tr>
<tr>
<td>2.4 Step 3: Verifying the Installation</td>
<td>2-8</td>
</tr>
<tr>
<td>2.5 Using the Debugger With the X Window System</td>
<td>2-9</td>
</tr>
</tbody>
</table>
2.1 What You’ll Need

The following checklists detail items that are shipped with the \texttt{C54x C source debugger} and simulator and additional items you’ll need to use these tools.

\textit{Hardware checklist}

- host \hspace{1cm} A SPARCstation or a system that is 100\% compatible with a SPARCstation 2 class or higher
- display \hspace{1cm} Monochrome or color monitor (color recommended)
- disk space \hspace{1cm} 2M bytes of disk space
- required hardware \hspace{1cm} CD-ROM drive
- optional hardware \hspace{1cm} Mouse
### What You’ll Need

**Software checklist**

- **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).

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

- **software tools**
  - ’C54x assembler and linker
  - Optional: ’C54x C compiler

- **optional files †**
  - `siminit.cmd` is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a ’C54x 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 defining your own memory map, refer to the Defining A Memory Map chapter in the TMS320C5xx C Source Debugger User’s Guide.
  - `sim541.cmd`, `sim542.cmd`, `sim543.cmd`, `sim544.cmd`, `sim545.cmd`, `sim546.cmd`, `sim547.cmd`, and `sim549.cmd` are batch files containing commands that configure a memory map. Each file simulates a different device—’C541, ’C542, ’C543, ’C544, ’C545, ’C546, ’C547, or ’C549.

- **init.clr** is a general-purpose screen configuration file. If init.clr isn’t present when you invoke the debugger, the debugger uses the default screen configuration.

- **init.25, init.43, and init.50** have been provided for basic 80 x 25, 80 x 43, and 80 x 50 screen sizes, respectively. The init.clr file brings up the debugger in 80 x 25 mode. To bring the debugger up in another mode, copy one of the init.xx files to the init.clr file.

- The default configuration is for color monitors; an additional file, `mono.clr`, 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 Customizing the Debugger Display chapter in the TMS320C5xx 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 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

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

- If you have SunOS 4.1.x, as root, load the CD-ROM into the drive and enter the following from a command shell:
  ```
  mount –rt hsfs /dev/sr0 /cdrom
  exit
  cd /cdrom
  ```

- If you have SunOS 5.0 or 5.1, as root, 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
  ```

- 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
    ```
  - 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
  ```
Step 1: Installing the Simulator and Debugger Software

Copying the files

After you’ve mounted 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 sim5xx on your hard disk. To create this directory, enter:

   mkdir sim5xx

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

   cp –r * sim5xx

Unmounting the CD-ROM

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

   If you have SunOS 4.1.x, 5.0, or 5.1, as root, 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 the tools work correctly, you must:

- Modify the path shell variable to include the sim5xx 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 sim5xx 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 sim5xx directory:

```
set path = (. /bin /usr/ucb /usr/contrib/bin /usr/bin 
/usr/openwin/bin /user/fred/sim5xx)
```

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 sim5xx directory by defining the D_DIR environment variable like the following:

  ```
  setenv D_DIR "/user/fred/sim5xx"
  ```

  (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.)
Step 2: Setting Up the Debugger Environment

For example, if your C54x programs were in a directory named /user/fred/c5xxsource, the D_SRC setup would be:

```bash
setenv D_SRC "/user/fred/c5xxsource"
```

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:

```bash
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:

- `–b` or `–b[br]`
- `–d` or `–d machinename`
- `–i` or `–i pathname`
- `–profile`
- `–s`
- `–t` or `–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 Overview of a Code Development and Debugging System chapter in the TMS320C5xx 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:

```bash
setenv DISPLAY "machine name"
```

For example, if you are running the debugger on a machine called opie and you want the C54x debugger display to appear on a machine called barney, the DISPLAY setup would be:

```bash
setenv DISPLAY barney:0
```

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

```bash
sim5xx –d barney:0
```

For more information about using the debugger under the X Window system, refer to Section 2.5, 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:

```bash
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:

```
$ sim5xx sample
```

You should see a display similar to this one:

![Display](image)

- 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 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 'C54x 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.

<table>
<thead>
<tr>
<th>Key</th>
<th>Keysym</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 to F10</td>
<td>F1 to F10</td>
</tr>
<tr>
<td>PAGE UP</td>
<td>Prior</td>
</tr>
<tr>
<td>PAGE DOWN</td>
<td>Next</td>
</tr>
<tr>
<td>HOME</td>
<td>Home</td>
</tr>
<tr>
<td>END</td>
<td>End</td>
</tr>
<tr>
<td>INSERT</td>
<td>Insert</td>
</tr>
<tr>
<td>→</td>
<td>Right</td>
</tr>
<tr>
<td>←</td>
<td>Left</td>
</tr>
<tr>
<td>↑</td>
<td>Up</td>
</tr>
<tr>
<td>↓</td>
<td>Down</td>
</tr>
</tbody>
</table>

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 'C54x debugger fonts to Courier, add the following line to the .Xdefaults file:

```
    sim5xx*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:

<table>
<thead>
<tr>
<th>Color</th>
<th>Appearance on Monochrome Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>black</td>
</tr>
<tr>
<td>blue</td>
<td>black</td>
</tr>
<tr>
<td>green</td>
<td>white</td>
</tr>
<tr>
<td>cyan</td>
<td>white</td>
</tr>
<tr>
<td>red</td>
<td>black</td>
</tr>
<tr>
<td>magenta</td>
<td>black</td>
</tr>
<tr>
<td>yellow</td>
<td>white</td>
</tr>
<tr>
<td>white</td>
<td>white</td>
</tr>
</tbody>
</table>
This chapter helps you install the 'C54x simulator and the C source debugger on a HP9000 series 700™ PA-RISC™ system running HP-UX™. When you complete the installation, turn to the TMS320C5xx C Source Debugger User’s Guide.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 What You’ll Need</td>
<td>2-2</td>
</tr>
<tr>
<td>2.2 Step 1: Installing the Simulator and Debugger Software</td>
<td>2-4</td>
</tr>
<tr>
<td>2.3 Step 2: Setting Up the Debugger Environment</td>
<td>2-6</td>
</tr>
<tr>
<td>2.4 Step 3: Verifying the Installation</td>
<td>2-8</td>
</tr>
<tr>
<td>2.5 Using the Debugger With the X Window System</td>
<td>2-9</td>
</tr>
</tbody>
</table>
3.1 What You’ll Need

The following checklists detail items that are shipped with the 'C54x C source debugger and simulator and additional items you’ll need to use these tools.

**Hardware checklist**

- **host** An HP9000 Series 700 PA-RISC system
- **display** Monochrome or color (color recommended)
- **disk space** 2M bytes of disk space
- **required hardware** CD-ROM drive
- **optional hardware** Mouse
What You'll Need

**Software checklist**

- **operating system**  
  HP-UX 9.x or higher.

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

- **software tools**  
  ‘C54x assembler and linker  
  Optional: ‘C54x C compiler

- **optional files †**  
  *siminit.cmd* is a general-purpose batch file that contains debugger commands. This batch file, shipped with the debugger, defines a ‘C54x 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 defining your own memory map, refer to the *Defining A Memory Map* chapter in the *TMS320C5xx C Source Debugger User’s Guide*.

  †  
  *sim541.cmd, sim542.cmd, sim543.cmd, sim544.cmd, sim545.cmd, sim546.cmd, sim547.cmd, and sim549.cmd* are batch files containing commands that configure a memory map. Each file simulates a different device—‘C541, ‘C542, ‘C543, ‘C544, ‘C545, ‘C546, ‘C547, or ‘C549.

  †  
  *init.clr* is a general-purpose screen configuration file. If init.clr isn’t present when you invoke the debugger, the debugger uses the default screen configuration.

  †  
  *init.25, init.43, and init.50* have been provided for basic 80×25, 80×43, and 80×50 screen sizes, respectively. The init.clr file brings up the debugger in 80×25 mode. To bring the debugger up in another mode, copy one of the init.xx files to the init.clr file.

  †  
  The default configuration is for color monitors; an additional file, *mono.clr*, 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 *Customizing the Debugger Display* chapter in the *TMS320C5xx C Source Debugger User’s Guide*.

† Included as part of the debugger package
3.2 Step 1: Installing the Simulator and Debugger Software

This section explains the process of installing 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**

As root, you can mount the CD-ROM using the UNIX mount command or the SAM (system administration manager):

- To use the UNIX mount command, enter:
  
  ```
  mount -rt cdfs /dev/dsk/your_cdrom_device /cdrom
  exit
  ```

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

  ```
  cd /cdrom/hp
  ```

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

**Copying the files and setting up the simulator**

After you’ve mounted 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 `sim5xx` on your hard disk. To create this directory, enter:

   ```
   mkdir sim5xx
   ```

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

   ```
   cd /cdrom/hp
   ```

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

   ```
   cp -r * sim5xx
   ```

**Unmounting the CD-ROM**

You must unmount the CD-ROM after copying the files. As root, enter:

```
exit
```
3.3 Step 2: Setting Up the Debugger Environment

To ensure that the tools work correctly, you must:

- Modify the path shell variable to include the sim5xx 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 sim5xx 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 sim5xx directory:

```
set path = (. /bin /usr/ucb /usr/contrib/bin /usr/bin /
/usr/openwin/bin /user/fred/sim5xx)
```

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 sim5xx directory by defining the D_DIR environment variable like the following:

  ```
  setenv D_DIR "/user/fred/sim5xx"
  ```

  (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.)
Step 2: Setting Up the Debugger Environment

For example, if your C54x programs were in a directory named `/user/fred/c5xxsource`, the `D_SRC` setup would be:

```
setenv D_SRC "/user/fred/c5xxsource"
```

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`:

- `-b[b]` - `machinename`
- `-i` - `pathname`
- `-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 Overview of a Code Development and Debugging System chapter in the TMS320C5xx 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 `C54x 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.

```
sim5xx -d barney:0
```

For more information about using the debugger under the X Window system, refer to Section 2.5, 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
```
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:

```
$ sim5xx 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 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 'C54x 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.

<table>
<thead>
<tr>
<th>Key</th>
<th>Keysym</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 to F10</td>
<td>F1 to F10</td>
</tr>
<tr>
<td>PAGE UP</td>
<td>Prior</td>
</tr>
<tr>
<td>PAGE DOWN</td>
<td>Next</td>
</tr>
<tr>
<td>HOME</td>
<td>Home</td>
</tr>
<tr>
<td>END</td>
<td>End</td>
</tr>
<tr>
<td>INSERT</td>
<td>Insert</td>
</tr>
<tr>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>Up</td>
<td>Up</td>
</tr>
<tr>
<td>Down</td>
<td>Down</td>
</tr>
</tbody>
</table>

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 'C54x debugger to Courier, add the following line to the .Xdefaults file:

```
sim5xx*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:

<table>
<thead>
<tr>
<th>Color</th>
<th>Appearance on Monochrome Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>black</td>
</tr>
<tr>
<td>blue</td>
<td>black</td>
</tr>
<tr>
<td>green</td>
<td>white</td>
</tr>
<tr>
<td>cyan</td>
<td>white</td>
</tr>
<tr>
<td>red</td>
<td>black</td>
</tr>
<tr>
<td>magenta</td>
<td>black</td>
</tr>
<tr>
<td>yellow</td>
<td>white</td>
</tr>
<tr>
<td>white</td>
<td>white</td>
</tr>
</tbody>
</table>
A

arrow keys
  HP systems 3-8
  SPARC systems 2-9

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

autoexec.bat file
  environmental variables 1-7
  invoking 1-6
  modifying 1-6
  sample 1-5

B

–b debugger option
  HP systems 3-6
  PC systems 1-7
  SPARC systems 2-7

batch files
  autoexec.bat 1-5 to 1-7
  .cshrc
    HP systems 3-5 to 3-6
    SPARC systems 2-6 to 2-7
  init.cdr
    HP systems 3-3
    PC systems 1-3
    SPARC systems 2-3
  initdb.bat 1-5 to 1-7
  initialization, siminit.cmd, SPARC systems 2-3
  mono.clr
    HP systems 2-3
    PC systems 1-3
    SPARC systems 2-3

batch files (continued)
  sim54x.cmd
    HP systems 3-3
    PC systems 1-3
    SPARC systems 2-3
  siminit.cmd
    HP systems 3-3
    PC systems 1-3
    SPARC systems 2-3

C

CD-ROM
  mounting
    HP systems 3-4
    SPARC systems 2-4
  requirements
    HP systems 3-4
    SPARC systems 2-4
  retrieving files from
    HP systems 3-4
    SPARC systems 2-4
  unmounting
    HP systems 2-4
    SPARC systems 2-4

  color mapping
    HP systems 3-9
    SPARC systems 2-10

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

  .cshrc file
    HP systems 3-5 to 3-6
    SPARC systems 2-6 to 2-7
Index

customizing the display
  HP systems 3-3, 3-9
  PC systems 1-2, 1-3
  SPARC systems 2-3, 2-10

D
–d debugger option
  HP systems 3-6
  PC systems 1-7
  SPARC systems 2-7

D_DIR environment variable
  HP systems 3-5
  PC systems 1-7
  SPARC systems 2-6

D_OPTIONS environment variable
  HP systems 3-6
  PC systems 1-7
  SPARC systems 2-7

D_SRC environment variable
  HP systems 3-5
  PC systems 1-7
  SPARC systems 2-6

directory
  HP systems 3-4
  PC systems 1-4
  SPARC systems 2-4

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

memory map
  HP systems 5-3
  PC systems 1-3
  SPARC systems 2-3

screen configuration file
  PC systems 1-3
  SPARC systems 2-3

directories
  auxiliary files
    HP systems 3-5
    PC systems 1-7
    SPARC systems 2-6

debugger software
  HP systems 3-4, 3-5
  PC systems 1-4, 1-6
  SPARC systems 2-5, 2-6

identifying additional source directories
  HP systems 3-5
  PC systems 1-7
  SPARC systems 2-6

sim5xx
  HP systems 3-4, 3-5
  PC systems 1-4, 1-7
  SPARC systems 2-5, 2-6

disk space requirements
  HP systems 3-2
  SPARC systems 2-2

display
  font changes
    HP systems 3-3
    PC systems 1-2
    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 2-7

E
end key
  HP systems 3-8
  SPARC systems 2-9

environment variables
  HP systems 3-5, 3-6
  PC systems 1-7
  SPARC systems 2-6, 2-7
F
font changes
   HP systems 3-9
   PC systems 1-2
   SPARC systems 2-10
function key mapping
   HP systems 3-8
   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 3-8
   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 1-7
   SPARC systems 2-7
init.clr file
   HP systems 3-3
   PC systems 1-3
   SPARC systems 2-3
initdb.bat file
   invoking 1-6
   sample 1-5
initialization batch files
   HP systems 3-3
   PC systems 1-3
   SPARC systems 2-3
insert key
   HP systems 3-8
   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 3-7
      PC systems 1-6
      SPARC systems 2-8
invoking
   autoexec.bat file 1-6
   .cshrc file
      HP systems 3-6
      SPARC systems 2-7
   initdb.bat file 1-6

K
keyboard mapping
   HP systems 3-8
   SPARC systems 2-9
keysym label
   HP systems 3-8
   SPARC systems 2-9

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

M
mapping keys
   HP systems 3-8
   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
  HP systems 1-2

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

monochrome monitors color mapping
  HP systems 3-9
  SPARC systems 2-10

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

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

page-down key
  HP systems 3-8
  SPARC systems 2-9

page-up key
  HP systems 3-8
  SPARC systems 2-9

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

permissions
  HP systems 3-3
  SPARC systems 2-3

–profile debugger option
  HP systems 3-6
  PC systems 1-7
  SPARC systems 2-7

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

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

–s debugger option
  HP systems 3-6
  PC systems 1-7
  SPARC systems 2-7

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

sim54x.cmd file
  HP systems 3-3
  PC systems 1-3
  SPARC systems 2-3

sim5xx
  command options
    HP systems 3-6
    PC systems 1-7
    SPARC systems 2-7
  directory
    HP systems 3-4
    PC systems 1-4
    SPARC systems 2-5
  verifying the installation
    HP systems 3-7
    SPARC systems 2-8

siminit.cmd file
  HP systems 3-3
  PC systems 1-3
  SPARC systems 2-3
simulator
  environment setup
    HP systems 3-5 to 3-6
    PC systems 1-5 to 1-7
  installation
    procedure
      HP systems 3-4
      PC systems 1-4
      SPARC systems 2-4
  verification
    HP systems 2-7
    PC systems 1-8
    SPARC systems 2-8
  SPARC systems, debugger environment 2-6 to 2-7
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 3-8
  SPARC systems 2-9
utilities
  xev
    HP systems 3-8
    SPARC systems 2-9
utilities (continued)
  xmodmap
    HP systems 3-8
    SPARC systems 2-9
  xrdb
    HP systems 3-9
    SPARC systems 2-10
V
  –v debugger option
    HP systems 3-6
    PC systems 1-7
    SPARC systems 2-7
  verifying the installation
    HP systems 3-7
    PC systems 1-8
    SPARC systems 2-8
X
  –x debugger option
    HP systems 3-6
    PC systems 1-7
    SPARC systems 2-7
  X Window System
    HP systems 3-8 to 3-9
    SPARC systems 2-9 to 2-10
  .Xdefaults file
    HP systems 3-9
    SPARC systems 2-10
xev utility
  HP systems 3-8
  SPARC systems 2-9
xmodmap utility
  HP systems 3-8
  SPARC systems 2-9
xrdb utility
  HP systems 3-9
  SPARC systems 2-10