VHDL VITAL™

Simulation Guide

Actel

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 5579006-11

Release: November 2006

Trademarks

Table of Contents

Introduction

This *VHDL Vital Simulation Guide* contains information about using the ModelSim and Cadence NC-VHDL to simulate designs for Actel devices. Refer to the *Designer User's Guide* for additional information about using the Designer software. Refer to the documentation included with your simulator for information about performing simulation.

Document Organization

The *VHDL Vital Simulation Guide* contains the following sections:

Chapter 1 - Setup contains information about setting up ModelSim and Cadence VHDL simulator.

Chapter 2 - Design Flow describes how to use the VHDL design flow to design an Actel device using the synthesis-tool software, and VHDL simulator software.

Chapter 3 - Generating Netlists contains information regarding how to generate a netlist using the synthesis-tool software.

Chapter 4 - Simulation with ModelSim contains information about simulating for Actel designs using the ModelSim simulator.

Chapter 5 - Simulation with Cadence NC-VHDL contains information about simulating for Actel designs using the Cadence NC-VHDL simulator.

Appendix A - Product Support provides information regarding contacting Actel for customer and technical support.

Document Assumptions

This document assumes the following:

1. You have installed the Designer Series software.
2. You have installed your VHDL VITAL simulator.
3. You are familiar with UNIX workstations and operating systems or with PCs and Windows operating environments.
4. You are familiar with FPGA architecture and FPGA design software.

Document Conventions

This document uses the following conventions:

Information input by the user follows this format:

```
keyboard input
```

The contents of a file follows this format:

```
file contents
```

This document uses the following variables:

- Actel FPGA family libraries are shown as <act_fam>. Substitute the desired Actel FPGA family variable with Fusion, IGLOO™/e, ProASIC3/E, act1, act2 (for ACT2 and 1200XL devices), ACT3, 3200DX, MX, SX, SX-A, eX, Axcelerator, A500k, and APA as needed. For example:

  ```
  vcom -work <vhd_fam> <act_fam>.vhd
  ```

- Compiled VHDL libraries are shown as <vhd_fam>. Substitute <vhd_fam> for the desired VHDL family variable with Fusion, IGLOO/e, ProASIC3/E, act1, act2 (for ACT2 and 1200XL devices), ACT3, 3200DX, MX, SX, SX-A, eX, Axcelerator, A500k, and APA as needed. The VHDL language requires that the library names begin with an alpha character.

Online Help

Designer software comes with online help. Online help specific to each software tool is available in Libero, Designer, SmartGen, Silicon Expert, Silicon Explorer II, and Silicon Sculptor.

1 Setup

This chapter contains information on setting up the ModelSim or Cadence NC-VHDL simulator to simulate Actel designs.

This chapter includes software requirements, steps describing how to compile Actel FPGA libraries, and other setup information for the simulation tool you use.

Software Requirements

The information in this guide applies to the Actel Designer software release R1-2003 or later and IEEE-1076-compliant VHDL simulators. Additionally, this guide contains information about using ModelSim and Cadence NC-VHDL simulators.

For specific information about which versions this release supports, go to the technical support system on the Actel web site (**http://www.actel.com/custsup/search.html**) and enter the following in the Keyword box:

```
third party
```

Migration Libraries

In addition to the standard Actel libraries, Actel provides a set of migration libraries. These libraries contain cores supported in 3.1.1u1 and earlier versions of the Designer Series software and cores possibly needed to retarget designs from a different Actel family. Actel does not recommend using the migration libraries on new designs.

ModelSim

Since the installation path varies for each user and each installation, this document uses "$ALSDIR" to indicate the location where the Actel Designer software is installed. If you are a Unix user, simply create an environment variable called ALSDIR and set its value to the installation path. If you are a Windows user, replace "$ALSDIR" with the installation path in the commands.

Use the following procedure to compile libraries for the ModelSim simulators. Type UNIX commands at the UNIX prompt. Type Windows commands on the command line of the ModelSim Transcript window. The commands below are for Windows. To make the commands work for UNIX, use forward slashes instead of back slashes.

This procedure compiles an Actel VITAL library in the “$ALSDIR\lib\vtl\95\mti” directory. You must compile the FPGA library models for the Actel VITAL libraries to work properly.

Note: If there is already an MTI directory in the $ALSDIR\lib\vtl\95 directory, compiled libraries may be present, and you may not need to perform the following procedure.

1. **Create a library called "mti" in the "$ALSDIR\lib\vtl\95" directory.**
2. **Invoke the ModelSim simulator (Windows only).**
3. **Change to the "$ALSDIR\lib\vtl\95\mti" directory.** Enter the following command at the prompt:

   ```
   cd $ALSDIR\lib\vtl\95\mti
   ```

4. **Create a <vhd_fam> family library.** Enter the following command at the prompt:

   ```
   vlib <vhd_fam>
   ```

5. **Map the Actel VITAL libray to the <vhd_fam> directory.** Enter the following command at the prompt:

   ```
   vmap <vhd_fam> $ALSDIR\lib\vtl\95\mti\<vhd_fam>
   ```

6. **Compile your VITAL libraries.**

   ```
   vcom -work <vhd_fam> ../<act_fam>.vhd
   ```

   For example, to compile the 40MX library for your simulator, type the following command:

   ```
   vcom -work a40mx ../40mx.vhd
   ```

7. **(Optional) Compile the migration library.** Only perform this step if you need to use the migration library. Type the following command at the prompt:

   ```
   vcom -work <vhd_fam> ../<act_fam>_mig.vhd
   ```

Cadence NC-VHDL

Use the following procedure to compile libraries for the Cadence NC-VHDL simulator. This procedure compiles an Actel VITAL library in the "$ALSDIR/lib/vtl/95/ncvhdl" directory. You must compile the FPGA library models for the Actel VITAL libraries to work properly.

1. **Create a directory called "ncvhdl" in the "$ALSDIR/lib/vtl/95" directory.**
2. **Change to the "$ALSDIR/lib/vtl/95/ncvhdl" directory.**
3. **Create a directory named <vhd_fam>.**
4. **Map the library.** Compile the models and create the "cds.lib" file as follows:

   ```
   INCLUDE $CDS/tools/inca/files/cds.lib
   ```

```
DEFINE <vhd_fam> $ALSDIR/lib/vtl/95/ncvhdl/<vhd_fam>
```

5. **Compile the library.** Type the following command at the prompt:

   ```
   ncvhdl -v93 -work <vhd_fam> -messages $ALSDIR/lib/vtl/95/<act_fam>.vhd
   ```

   For example, to compile the 40MX library for your simulator, type the following command:

   ```
   ncvhdl -v93 -work a40mx -messages $ALSDIR/lib/vtl/95/40mx.vhd
   ```

6. **(Optional) Compile the migration library.** Only perform this step if you are using the migration library. Type the following command at the prompt:

   ```
   ncvhdl -work <vhd_fam> -messages $ALSDIR/lib/vtl/95/ <act_fam>_mig.vhd
   ```

2 Design Flow

This chapter describes the design flow for simulating Actel designs with a VHDL VITAL-compliant simulation tool.

VHDL VITAL Design Flow Illustrated

Figure 2-1 shows the design flow for an Actel FPGA using Designer software and a VITAL-compliant VHDL simulator[1].

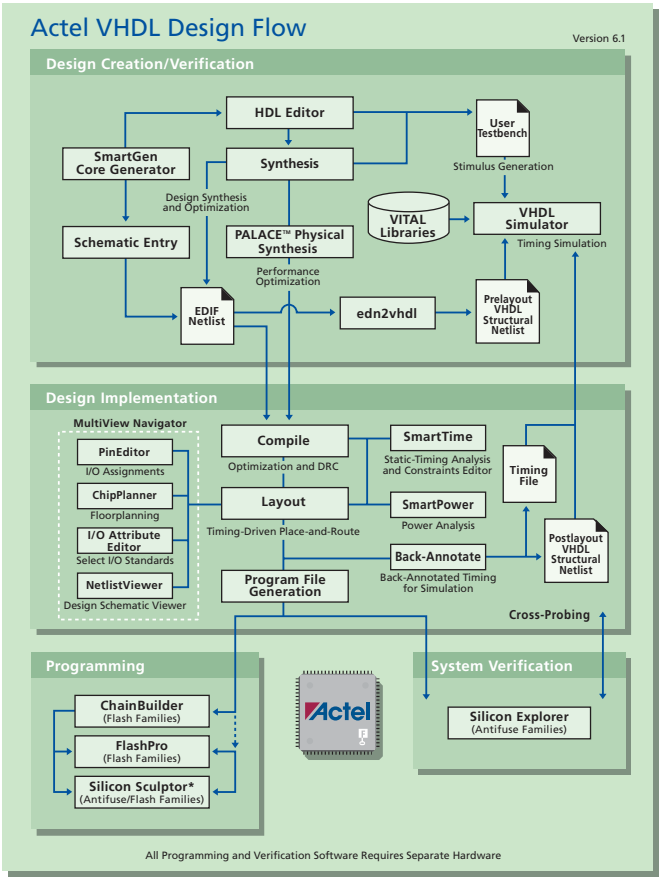

Figure 2-1. Actel-VHDL Design Flow

VHDL VITAL Design Flow Described

The Actel VHDL VITAL design flow has four main steps:

1. Design Creation/Verification
2. Design Implementation
3. Programming
4. System Verification

The following sections detail these steps.

Design Creation/Verification

During design creation/verification, a design is captured in an RTL-level (behavioral) VHDL source file. After capturing the design, you can perform a behavioral simulation of the VHDL file to verify that the VHDL code is correct. The code is then synthesized into an Actel gate-level (structural) VHDL netlist. After synthesis, you can perform an optional pre-layout structural simulation of the design. Finally, an EDIF netlist is generated for use in Designer and a VHDL structural post-layout netlist is generated for timing simulation in a VHDL VITAL-compliant simulator.

VHDL Source Entry

Enter your VHDL design source using a text editor or a context-sensitive HDL editor. Your VHDL design source can contain RTL-level constructs, as well as instantiations of structural elements, such as SmartGen cores.

Behavioral Simulation

Perform a behavioral simulation of your design before synthesis. Behavioral simulation verifies the functionality of your VHDL code. Typically, you use zero delays and a standard VHDL test bench to drive simulation. Refer to the documentation included with your simulation tool for information about performing functional simulation.

Synthesis

After you have created your behavioral VHDL design source, you must synthesize it. Synthesis transforms the behavioral VHDL file into a gate-level netlist and optimizes the design for a target technology. The documentation included with your synthesis tool contains information about performing design synthesis.

1. The grey boxes in Figure 2-1 denote Actel-specific utilities/tools.

EDIF Netlist Generation

After you have created, synthesized, and verified your design, you must generate an Actel EDIF netlist for place-and-route in Designer.

This EDIF netlist is also used to generate a structural VHDL netlist for use in structural simulation.

Structural VHDL Netlist Generation

Generate a gate-level VHDL netlist from your EDIF netlist for use in post-synthesis pre-layout structural simulation by either exporting it from Designer or by using the Actel "edn2vhdl" program.

Structural Simulation

Perform a structural simulation before placing-and-routing. Structural simulation verifies the functionality of your post-synthesis pre-layout structural VHDL netlist. Unit delays included in the compiled Actel VITAL libraries are used. Refer to the documentation included with your simulation tool for information about performing structural simulation.

Design Implementation

During design implementation, you place-and-route a design using Designer. Additionally, you may perform timing analysis on a design in Designer with the Timer tool. After place-and-route, perform postlayout (timing) simulation with a VHDL VITAL-compliant simulator.

Place-and-Route

Use Designer to place-and-route your design. Refer to the *Designer User's Guide* for information about using Designer.

Timing Analysis

Use the Timer tool in Designer to perform static-timing analysis on your design. Refer to the *Timer User's Guide* for information about using Timer.

Timing Simulation

Perform a timing simulation on your design after place-and-route. Timing simulation verifies that the design meets your timing constraints.

Programming

Program a device with programming software and hardware from Actel or a supported third-party programming system. Refer to the *Designer User's Guide* and the *Activator and APS Programming System Installation and User's Guide* or *Silicon Sculptor User's Guide* for information about programming an Actel device.

System Verification

You can perform system verification on a programmed device using the Actel Silicon Explorer diagnostic tool. Refer to the *Activator and APS Programming System Installation and User's Guide* or *Silicon Explorer Quick Start* for information about using the Silicon Explorer.

3

Generating Netlists

This chapter describes the procedures for generating EDIF and structural VHDL netlists.

Generating an EDIF Netlist

After capturing your schematic or synthesizing your design, generate an EDIF netlist from your schematic capture or synthesis tool. Use the EDIF netlist for place-and-route in Designer. Refer to the documentation included with your schematic capture or synthesis tool for information about generating an EDIF netlist.

Generating a Structural VHDL Netlist

You can generate a structural VHDL netlist using Designer or the "edn2vhdl" program. Use the structural VHDL netlist for structural and timing simulation.

To generate a structural netlist using Designer,

1. **Invoke Designer.**
2. **Import the EDIF netlist.** Choose the Import Source File command from the File menu. This displays the Import Source dialog box. Type the full path name of your EDIF netlist or use the Browse button to select your design. Click OK. In the EDIF Import Options dialog box, specify Generic as the EDIF flavor.
3. **Export a structural VHDL netlist.** From the File menu, choose Export, then Netlist. This displays the Export Netlist Files dialog box. In the Export Netlist Files dialog box, specify the File name and set the Save As type to VHDL. Click OK.

To generate a structural netlist using "edn2vhdl,"

1. **Change to the directory that contains the VHDL design files.**
2. **Type the following command at the UNIX or DOS prompt:**

```
edn2vhdl FAM:{<act_fam>} [ EDNIN:<Edif_File1>[+<Edif_File2...>] ][ VHDOUT:<Vhdl_File>
]<design_name>
```

The "EDNIN" option specifies the EDIF input file(s). You can specify multiple files with the "+" delimiter between file names. The default EDIF input file is <design_name>.edn. The "VHDOUT" option specifies the VHDL output file names. The default VHDL output file is <design_name>.vhd.

4

Simulation with ModelSim

This chapter describes steps to perform Functional (Behavioral and Structural) and Timing simulation for Actel devices using the ModelSim simulator.

The procedures shown are for PC. The same setup procedures work similarly for UNIX. Use forward slashes in place of back slashes. For PC, type commands into the MTI window. For UNIX, type commands into a UNIX window.

Behavioral Simulation

Use the following procedure to perform a behavioral simulation of a design. Refer to the documentation included with your simulation tool for additional information about performing behavioral simulation.

1. **Invoke your ModelSim simulator. (PC only)**
2. **Change directory to your project directory.** This directory must include your VHDL design files and testbench. Type:

   ```
   cd <project_dir>
   ```

3. **Map to the Actel Library.** If any Actel cores are instantiated in your VHDL source, type the following command to map them to the compiled Actel VITAL library.

   ```
   vmap <vhd_fam> $ALSDIR\lib\vtl\95\mti\<vhd_fam>
   ```

   To reference the Actel family library in your VHDL design files, add the following lines to your VHDL design files:

   ```
   library <vhd_fam>;
   use <vhd_fam>.components.all;
   ```

4. **Create a "work" directory.** Type:

   ```
   vlib work
   ```

5. **Map to the "work" directory.** Type the following command:

   ```
   vmap work .\work
   ```

6. **Perform a behavioral simulation of your design.** To perform a behavioral simulation using your V-System or ModelSim simulator, compile your VHDL design and testbench files and run a simulation. For hierarchical designs, compile the lower-level design blocks before the higher-level design blocks.

The following commands demonstrate how to compile VHDL design and testbench files:

```
vcom -93 <behavioral>.vhd
vcom -93 <test_bench>.vhd
```

To simulate the design, type:

```
vsim <configuration_name>
```

For example:

```
vsim test_adder_behave
```

The entity-architecture pair specified by the configuration named test_adder_behave in the testbench will be simulated.

If your design contains a PLL core, use a 1ps resolution:

```
vsim -t ps <configuration_name>
```

For example:

```
vsim -t ps test_adder_behave
```

Structural Simulation

Use the following procedure to perform structural simulation.

1. **Generate a structural VHDL netlist.**

   If you are using Synopsys Design Compiler, generate a structural VHDL netlist using this tool.

   If you are using other synthesis tools, generate a gate-level VHDL from your EDIF netlist by either exporting it from Designer or by using the edn2vhdl program.

To generate a structural netlist using "edn2vhdl,"

2. **Change to the directory that contains the VHDL design files.**
3. **Type the following command at the UNIX or DOS prompt:**

```
edn2vhdl FAM:{<act_fam>} [ EDNIN:<Edif_File1>[+<Edif_File2...>] ][ VHDOUT:<Vhdl_File>
]<design_name>
```

The "EDNIN" option specifies the EDIF input file(s). You can specify multiple files with the "+" delimiter between file names. The default EDIF input file is <design_name>.edn. The

"VHDOUT" option specifies the VHDL output file names. The default VHDL output file is <design_name>.vhd.

To generate a netlist using Designer Series software,

4. **Export a structural VHDL netlist.** From the File menu, choose Export, then Netlist. This displays the Export Netlist Files dialog box. In the Export Netlist Files dialog box, specify the File name and set the Save As type to VHDL. Click OK.

   Note: The VHDL generated by both Designer and the edn2vhdl program will use std_logic for all ports. The bus ports will be in the same bit order as they appear in the EDIF netlist.

5. **Map to the Actel VITAL library.** Run the following command to map the compiled Actel VITAL library.

   ```
   vmap <vhd_fam> $ALSDIR\lib\vtl\95\mti\<vhd_fam>
   ```

6. **Compile the structural netlist.** Compile your VHDL design and testbench files. The following commands demonstrate how to compile VHDL design and testbench files:

   ```
   vcom -just e -93 <structural>.vhd
   vcom -just a -93 <structural>.vhd
   vcom <test_bench>.vhd
   ```

   Note: First, the application compiles the entities. Then, it compiles the architectures, as required for VHDL netlists written by some tools.

7. **Run the structural simulation.** To simulate your design, type:

   ```
   vsim <configuration_name>
   ```

   For example:

   ```
   vsim test_adder_structure
   ```

   The entity-architecture pair specified by the configuration named test_adder_structure in the testbench will be simulated.

   If your design contains a PLL core, use a 1ps resolution:

   ```
   vsim -t ps <configuration_name>
   ```

   For example:

   ```
   vsim -t ps test_adder_structure
   ```

Timing Simulation

Use the following procedure to perform timing simulation.

1. **Place-and-route your design in Designer.** Refer to the *Designer User's Guide* for information about using Designer.
2. **Extract timing information for your design.** From the File menu, click Export. Then, click Timing Files. Choose SDF and click Save (or click Back Annotate). The Back Annotate dialog box is displayed. Create a <design_name>.sdf file by specifying SDF as the CAE type. Click OK.
3. **Compile the structural netlist.** To perform a timing simulation using your V-System or ModelSim simulator, compile your VHDL design and testbench files, if they have not already been compiled for a structural simulation, and run a simulation. The following commands demonstrate how to compile VHDL design and testbench files:

   ```
   vcom -just e -93 <structural>.vhd
   vcom -just a -93 <structural>.vhd
   vcom <test_bench>.vhd
   ```

   Note: Performing the previous steps compiles the entities first and then the architectures, as required for VHDL netlists written by some tools.
4. **Run the back-annotation simulation using the timing information in the SDF file.** Type:

   ```
   vsim -sdf[max|typ|min] /<region>=<design name>.sdf -c
   <configuration_name>
   ```

   The <region> option specifies the region (or path) to an instance in a design where back annotation begins. You can use it to specify a particular FPGA instance in a larger system design or testbench that you wish to back annotate. For example:

   ```
   vsim -sdfmax /uut=adder.sdf -c test_adder_structural
   ```

   In this example, the entity "adder" has been instantiated as instance "uut" in the testbench. The entity-architecture pair specified by the configuration named "test_adder_structural" in the testbench will be simulated using the maximum delays specified in the SDF file.

If your design contains a PLL core, use a 1ps resolution:

```
vsim -t ps -sdf[max|typ|min] /<region>=<design name>.sdf -c
<configuration_name>
```

For example:

```
vsim -t ps -sdfmax /uut=adder.sdf -c test_adder_structural
```

5
Simulation with Cadence NC-VHDL

This chapter describes steps to perform Functional (Behavioral and Structural) and Timing simulation for Actel devices using the Cadence NC-VHDL simulator.

Behavioral Simulation

After you have coded the VHDL descriptions of the logic blocks, test and debug the design using the NC-VHDL simulator. Use the following procedure to perform behavioral simulation.

1. **Create a work directory and a cds.lib file in the project directory.** At the UNIX prompt, type:

   ```
   mkdir work
   ```

2. **Using a text editor, create a cds.lib file and enter the following lines:**

   ```
   INCLUDE $CDS/tools/inca/files/cds.lib
   ```

3. **Map to the Actel VITAL library.** If any Actel cores are instantiated in your VHDL source, add the following lines to your cds.lib file to map them to the compiled Actel VITAL library.

   ```
   DEFINE <vhd_fam> $ALSDIR/lib/vtl/95/ncvhdl/<vhd_fam>
   DEFINE WORK ./work
   ```

   Add the following lines to your VHDL design files to reference the Actel Family library in your VHDL design files:

   ```
   library <vhd_fam>;
   use <vhd_fam>.components.all;
   ```

4. **Compile the VHDL design and testbench files.** Type:

   ```
   ncvhdl -work work -messages <behavioral>.vhd
   ncvhdl -work work -messages <test_bench>.vhd
   ```

5. **Elaborate the design.** Type the following commands at the prompt:

   ```
   ncelab -work work -messages <configuration_name>
   ```

   In the above, <configuration_name> is the name of the configuration in the testbench.

   For example

   ```
   ncelab -work work -messages test_add_behave
   ```

The entity-architecture pair specified by the configuration named test_adder_behave in the testbench will be elaborated.

6. **Simulate the design.** Type the following command at the prompt:

```
ncsim -batch -run <configuration_name>
```

In the above, <configuration_name> is the name of the configuration in the testbench.

For example:

```
ncsim -batch -run test_add_behave
```

Structural Simulation

Use the following procedure to perform structural simulation.

1. **Map to the Actel VITAL library.** Create a cds.lib file in the project directory as follows:

```
INCLUDE $CDS/tools/inca/files/cds.lib
DEFINE <vhd_fam> $ALSDIR/lib/vtl/95/ncvhdl/<vhd_fam>
DEFINE WORK ./work
```

2. **Compile the structural netlist.** Compile your VHDL design and testbench files. If you have not already generated a structural netlist, go to "Generating a Structural VHDL Netlist" on page 15. The following commands demonstrate how to compile VHDL design and testbench files:

```
ncvhdl -work work -messages <structural>.vhd
ncvhdl -work work -messages <test_bench>.vhd
```

3. **Elaborate the design.** Type:

```
ncelab -work work -messages <configuration_name>
```

In the above, <configuration_name> is the name of your configuration that binds the testbench entity and architecture.

For example:

```
ncelab -work work -messages test_adder_structure
```

In the example, "test_adder_structure" is the name of configuration for the testbench.

4. **Simulate the design.** Type following at the prompt:

```
ncsim -batch -run <configuration_name>
```

In the above, <configuration_name> is name of the testbench configuration.

For example:

```
ncsim -batch -run test_adder_structure
```

Here the name of the testbench configuration is "test_adder_structure."

Timing Simulation

Use the following procedure to perform timing simulation.

1. **Place-and-route your design in Designer.** Refer to the *Designer User's Guide* for information about using Designer.
2. **Extract timing information for your design.** From the File menu, click Export. Then, click Timing Files. Choose SDF and click Save (or click Back Annotate). The Back Annotate dialog box is displayed. Create a <design_name>.sdf file by specifying SDF as the CAE type. Click OK.
3. **Map Actel VITAL library.** If not already done for structural simulation, create a cds.lib file in the project directory using the following command:

   ```
   INCLUDE $CDS/tools/inca/files/cds.lib
   DEFINE <vhd_fam> $ALSDIR/lib/vtl/95/ncvhdl/<vhd_fam>
   DEFINE WORK ./work
   ```

4. **Compile the structural netlist.** Compile your VHDL design and testbench files. If you have not already generated a structural netlist, go to "Generating a Structural VHDL Netlist" on page 15. The following commands demonstrate how to compile VHDL design and testbench files:

   ```
   ncvhdl -work work -messages <structural>.vhd
   ncvhdl -work work -messages <test_bench>.vhd
   ```

5. **Compile the SDF file.** Type:

   ```
   ncsdfc <design_name>.sdf
   ```

6. **Write an SDF command file.** Using a text editor, create the "<design_name>.sdf_cmd" file. Include the following lines:

   ```
   COMPILED_SDF_FILE = "<design_name>.sdf.x"
   ```

```
SCOPE = :<UUT>,

MTM_CONTROL = "MINUMUM/TYPICAL/MAXIMUM";
```

In the above, <design_name> is the name of the top-level entity, <UUT> is the instance of the top-level entity in the testbench, and <configuration_name> is the name of your configuration that binds the testbench entity and architecture.

7. **Elaborate the design.** Type:

```
ncelab -work work -messages -sdf_cmd_file <design_name>.sdf_cmd
<configuration_name>
```

For example:

```
ncelab -work work -messages -sdf_cmd_file adder.sdf_cmd
test_adder_structure
```

In the above example, "adder" is the name of the top-level entity, "test_adder_structure" is the name of the configuration for the testbench.

8. **Simulate the design.** Type the following at the prompt:

```
ncsim -batch -run <configuration_name>
```

In the above, <configuration_name> is the name of the testbench configuration.

For example:

```
ncsim -batch -run test_adder_structure
```

Here the name of the testbench configuration is test_adder_structure.

A

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From Northeast and North Central U.S.A., call **650.318.4480**
From Southeast and Southwest U.S.A., call **650. 318.4480**
From South Central U.S.A., call **650.318.4434**
From Northwest U.S.A., call **650.318.4434**
From Canada, call **650.318.4480**
From Europe, call **650.318.4252** or **+44 (0)1276.401500**
From Japan, call **650.318.4743**
From the rest of the world, call **650.318.4743**
Fax, from anywhere in the world **650. 318.8044**

Actel Customer Technical Support Center

Actel staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Actel Technical Support

Visit the Actel Customer Support website (www.actelcom/.custsup/search.html) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's home page, at www.actel.com.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is tech@actel.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460
800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. Sales office listings can be found at www.actel.com/contact/offices/index.html.

Index

T

U

V

W

For more information about Actel's products, visit our website at http://www.actel.com

Actel Corporation *• 2061 Stierlin Court • Mountain View, CA 94043 USA*
Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060

Actel Europe Ltd*. • River Court, Meadows Business Park • Station Approach, Blackwater • Camberley, Surrey GU17 9AB*
United Kingdom • Phone +44 (0) 1276 609 300 • Fax +44 (0) 1276 607 540

Actel Japan *• EXOS Ebisu Bldg. 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan*
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • www.jp.actel.com

Actel Hong Kong *• Suite 2114, Two Pacific Place • 88 Queensway, Admiralty Hong Kong*
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn

5579006-11/11.06