# Verilog

## Simulation Guide

**Actel Corporation, Mountain View, CA 94043**

Printed in the United States of America

Part Number: 5579005-8/10.04

Release: October 2004

Actel makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Actel assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Actel Corporation.

## Trademarks

Actel and the Actel logo are registered trademarks of Actel Corporation.

Adobe and Acrobat Reader are registered trademarks of Adobe Systems, Inc.

All other products or brand names mentioned are trademarks or registered trademarks of their respective holders.

# Table of Contents

Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

Document Organization . . . . . . . . . . . . . . . . . . . . . . . 5

Document Assumptions . . . . . . . . . . . . . . . . . . . . . . . 5

Document Conventions . . . . . . . . . . . . . . . . . . . . . . . 6

Online Help . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1 Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

Software Requirements . . . . . . . . . . . . . . . . . . . . . . 7

Verilog Libraries . . . . . . . . . . . . . . . . . . . . . . . . . 7

Compiling Verilog Libraries . . . . . . . . . . . . . . . . . . . . 7

2 Design Flow . . . . . . . . . . . . . . . . . . . . . . . . . . 9

Actel-Verilog Design Flow Illustrated . . . . . . . . . . . . . . . 9

Actel-Verilog Design Flow Overview . . . . . . . . . . . . . . . 10

3 Generating Netlists . . . . . . . . . . . . . . . . . . . . . 13

Generating an EDIF Netlist . . . . . . . . . . . . . . . . . . . 13

Generating a Structural Verilog Netlist . . . . . . . . . . . . . 13

4 Interpreted Simulation . . . . . . . . . . . . . . . . . . . . 15

Example Testbench . . . . . . . . . . . . . . . . . . . . . . . 15

Example Command File . . . . . . . . . . . . . . . . . . . . . 15

Behavioral Simulation . . . . . . . . . . . . . . . . . . . . . . 16

Structural Simulation . . . . . . . . . . . . . . . . . . . . . . 17

Timing Simulation . . . . . . . . . . . . . . . . . . . . . . . . 18

Verilog Switches . . . . . . . . . . . . . . . . . . . . . . . . . 20

5 Simulation with ModelSim . . . . . . . . . . . . . . . . . . . 23

Behavioral Simulation . . . . . . . . . . . . . . . . . . . . . . 23

Structural Simulation . . . . . . . . . . . . . . . . . . . . . . 24

Timing Simulation . . . . . . . . . . . . . . . . . . . . . . . . 25

A Product Support . . . . . . . . . . . . . . . . . . . . . . . . 27

Customer Service . . . . . . . . . . . . . . . . . . . . . . . . 27

Actel Customer Technical Support Center . . . . . . . . . . . . . . . . . . . . 27
Actel Technical Support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
Website . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
Contacting the Customer Technical Support Center . . . . . . . . . . . . . . . . 28

Index . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

# Introduction

The Verilog Simulation Guide contains information about interfacing the Designer Series FPGA development software with Verilog simulation tools. Refer to the Designer online help for additional information about using the Designer software. Refer to the documentation included with your Verilog simulation tool for information about performing simulation.

## Document Organization

The *Verilog Simulation Guide* contains the following chapters:

Chapter 1 - Setup contains information about setting up Verilog libraries for use in simulating Actel designs.

Chapter 2 - Design Flow illustrates and describes the design flow for simulating Actel designs using Verilog Simulation tools.

Chapter 3 - Generating Netlists contains information for generating EDIF and structural Verilog netlists.

Chapter 4 - Interpreted Simulation describes the procedures for performing functional (behavioral and structural) and timing simulation on an Actel design using tools that either interpret library and design files, or compile them on-the-fly.

Chapter 5 - Simulation with ModelSim describes the procedures for performing functional (behavioral and structural) and timing simulation on an Actel design using commands for the ModelSim simulator.

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

## Document Assumptions

This document assumes the following:

1. If you are using a PC, you have installed the Designer Series software in the “c:\actel” directory.
2. You have installed a Verilog simulator.
3. You are familiar with UNIX workstations and operating systems.
4. You are familiar with PCs and Windows operating environments.
5. 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 act1, act2 (for ACT 2 and 1200XL devices), act3, 3200dx, 40mx, 42mx, 54sx, 54sxa, ex, a500k, apa, or axcelerator as needed. For example:

```
vlog -work <act_fam> <act_fam>.v
```

# Online Help

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

# 1

# Setup

This chapter contains information about the directory structure of the Verilog libraries and setting up Verilog libraries for use in simulating Actel designs. Refer to the documentation included with your Verilog simulator for information about setting up your simulation tool.

## Software Requirements

The information in this guide applies to the Actel Designer software release R1-2000 or later and IEEE-1364-compliant Verilog simulators. Additionally, this guides contains information about using the PC and UNIX ModelSim simulators.

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

```
third party
```

If you are using HP-UX, you must also set the following variable:

```
setenv SHLIB_PATH $ALSDIR/lib
```

Refer to the Designer online help and the documentation included with your simulation tool for additional information about setting environment variables.

## Verilog Libraries

### Migration Libraries

In addition to the standard Actel libraries, Actel provides a set of migration libraries. These libraries contain macros supported in 3.1.1u1 and earlier versions of the Designer Series software and macros possibly needed to retarget designs from a different Actel family. Actel does not recommend using the migration libraries on new designs. The Verilog migration libraries are installed in the “<actel_install_directory>/lib/vlog/<act_fam>_mig.v” directories.

Note: Verilog migration libraries are not available for A500K, APA, and ProASIC3/E

## Compiling Verilog Libraries

Before simulating a design with the ModelSim Verilog simulator, you must compile the Actel Verilog libraries. This section describes the procedures. Refer to the documentation included with your simulation tool for additional information about compiling libraries.

## ModelSim

Use the following procedure to compile Verilog libraries for the ModelSim simulator. Type UNIX commands at the UNIX prompt. The commands below are for PC. To compile the Verilog libraries:

1. **Create a directory called "mti" in the appropriate directory:**

   "$ALSDIR\lib\vlog"

2. **Invoke the simulator (PC only).**
3. **Change to the "$ALSDIR\lib\vlog\mti" directory.** Type the following command at the prompt:

   ```
   cd $ALSDIR\lib\vlog\mti
   ```

4. **Create an <act_fam> family library directory for the simulator.** Type the following command:

   ```
   vlib <act_fam>
   ```

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

   ```
   vlog -work <act_fam> $ALSDIR\lib\vlog\<act_fam>.v
   ```

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

   ```
   vlog -work <act_fam> $ALSDIR\lib\vlog\<act_fam>_mig.v
   ```

   Note: There are no verilog migration libraries available for A500K, APA, and ProASIC3/E

2

# Design Flow

This chapter describes the design flow for creating Actel designs using Verilog simulation and the Designer software.

## Actel-Verilog Design Flow Illustrated

Figure 2-1 shows the design flow for an Actel device using CAE software, a Verilog simulation tool, and the Designer software[1].

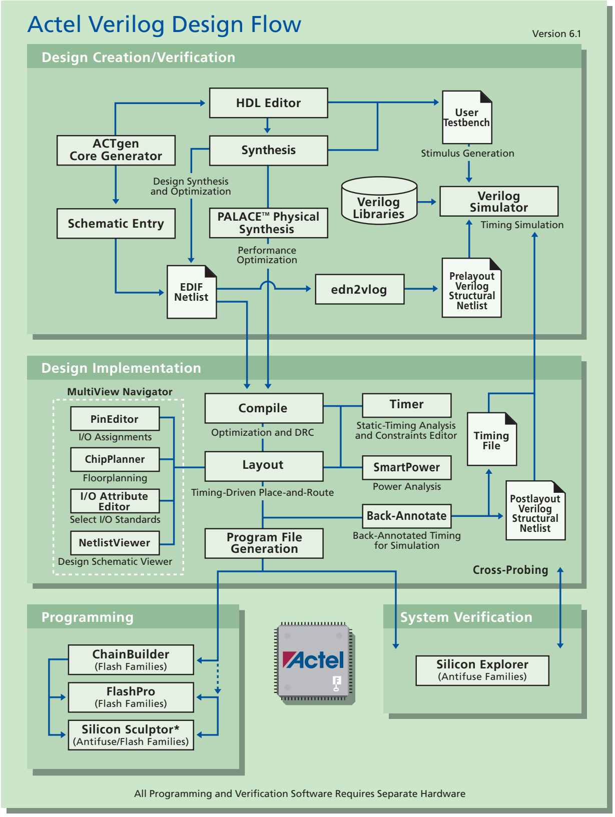

Figure 2-2. Actel-Verilog Design Flow

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

# Actel-Verilog Design Flow Overview

The Actel-Verilog design flow has four main steps:

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

The following sections describe these steps.

## Design Creation/Verification

During design creation/verification, a design is captured as a schematic or as an RTL-level (behavioral) Verilog HDL source file.

If your design is a Verilog HDL source file, you can perform a behavioral simulation to verify that the HDL code is correct. The code is then synthesized into an Actel gate-level (structural) Verilog HDL netlist. After synthesis, you can perform a structural simulation of the design. Finally, you generate an EDIF netlist for use in Designer and a structural Verilog netlist for structural and timing simulation.

If your design is a schematic, you generate an EDIF netlist for use in Designer and a structural Verilog netlist for structural and timing simulation. You do not perform behavioral simulation or synthesis.

### Design Capture

Enter your schematic using a third-party schematic-capture tool or create your Verilog HDL source file using a text editor or a context-sensitive HDL editor. Your Verilog HDL design source can contain RTL-level constructs, as well as instantiations of structural elements, such as ACTgen macros. Refer to the documentation included with your design-capture tool for information about design capture.

### Behavioral Simulation

Perform a behavioral simulation of your design before synthesis. Behavioral simulation verifies the functionality of your Verilog HDL code. You can use a standard Verilog HDL testbench to drive simulation. Refer to "Behavioral Simulation" on page 16 and the documentation included with your simulation tool for information about performing functional simulation.

### Synthesis

After you have created your Verilog HDL source file, you must synthesize it before placing-and-routing it in Designer. Synthesis transforms the Verilog HDL source file into a gate-level netlist and optimizes the design for a target technology. Refer to the documentation included with your synthesis tool for information about performing design synthesis.

### EDIF Netlist Generation

After you have created, synthesized (if your design is an HDL source file), and verified your design, you must generate an EDIF netlist or import verilog netlists for place-and-route in Designer. If your design is a Verilog HDL source file, use the EDIF netlist to generate a structural Verilog netlist. Refer to "Generating an EDIF Netlist" on page 13 and the documentation included with your schematic-capture or synthesis tool for information about generating an EDIF netlist.

### Structural Verilog Netlist Generation

Generate a structural Verilog netlist from your EDIF netlist for use in structural and timing simulation by either exporting it from Designer or by using the Actel "edn2vlog" program. Refer to "Generating a Structural Verilog Netlist" on page 13 for information about generating a structural netlist.

### Structural Simulation

Perform a structural simulation of your design before placing-and-routing it. Structural simulation verifies the functionality of your structural Verilog netlist. Use default unit delays included in the Verilog libraries for every gate. Refer to "Structural Simulation" on page 17 and 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 can perform static-timing analysis on a design in Designer with the Timer tool. After place-and-route, you can perform postlayout (timing) simulation with a Verilog simulator.

### Place-and-Route

**Use Designer to place-and-route your design.** Make sure to specify Verilog as the Naming Style when importing the EDIF netlist into Designer. Refer to the *Designer User's Guide* for information about using Designer.

### Static-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 of your design after placing-and-routing it. Timing simulation requires information extracted from Designer, which overrides default unit delays in the Actel Verilog libraries. Refer to "Timing Simulation" on page 18 and the documentation included with your simulation tool for information about performing timing simulation.

## Programming

Program a device with programming software and hardware from Actel or a supported third-party programming system. Refer to the *Designer Online Help* 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 or *Silicon Explorer User's Guide* for information about using the Silicon Explorer.

3

# Generating Netlists

This chapter describes the procedures for generating EDIF and structural Verilog 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.

Make sure to specify Verilog for the naming style when importing the EDIF netlist into Designer.

## Generating a Structural Verilog Netlist

You can generate structural Verilog netlist using Designer or the “edn2vlog” program. Use the structural Verilog netlist for structural and timing simulation.

### *To generate a structural Verilog netlist using Designer,*

1. **Invoke Designer.**
2. **Import the EDIF netlist.** Choose the Import Netlist File command from the File menu. The Import Netlist dialog box is displayed. Specify EDIF as the Netlist Type, GENERIC as the Edif flavor, and Verilog as the Naming Style. Type the full path name of your EDIF netlist or use the Browse button to select your design. Click OK.
3. **Export a structural Verilog netlist.** Choose the Export command from the File menu and then choose Netlist. The Export Netlist File dialog box is displayed. In the pull-down menu, click Verilog Files (*.v) and enter the name of the Verilog file you want to save.

### *To generate a structural netlist using edn2vlog,*

1. **Change to the directory that contains the EDIF netlist.**
2. **Type the following command at the UNIX or DOS prompt:**

```
edn2vlog FAM:{<act_fam>}
[ EDNIN:<EdifFile1>{+<EdifFile2...>} ]
[ VLGOUT:<Verilog_File> ]
[ SIMTEMP:<Stimulus_template_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 “VLOGOUT” option specifies the Verilog output file names. The default Verilog output file is

<design_name>.v. The “SIMTEMP” option instructs the program to generate a Verilog stimulus template file. If you do not specify this option, the program does not generate a stimulus file.

# 4

# Interpreted Simulation

This chapter describes the procedures for performing functional (behavioral and structural) and timing simulation on an Actel design by using tools that either interpret library and design files or compile them on-the-fly. Cadence NC-Verilog, Simucad SilosIII, and Synopsys VCS are simulators in this category.

This chapter includes information about creating a testbench and information about creating a command file to run a simulation in batch mode. Also, this chapter includes a description of some common Verilog simulation switches. Refer to the documentation included with your simulation tool for additional information about testbenches, command files, switches, and simulation.

## Example Testbench

You can use a testbench to apply test vectors or patterns to a design during simulation to compare input and output patterns. The file can instantiate the top-level design, using a Verilog-predefined command, such as "$readmemb," "$monitor," and "$display." To use the testbench, it must be in the current project directory. The following is an example testbench:

```
'timescale 1ns/100ps
module test;
//Inputs and outputs declaration
wire ...........
reg ..................
//Instantiate the top module of your design in the test module
<top_module> <instance_name> (.....Pin List...);
...........
//stimulus patterns
initial
begin
.....
end
endmodule
```

## Example Command File

You can use a command file for batch simulation. Your command file should include all command variables and Verilog switches you want to set during simulation. The following is an example command file:

```
<test_bench>.v
<design_name>.v
<verilog_switch_1> ... <verilog_switch_n>
```

The <design_name>.v variable is the Verilog top-level design that includes all design sublevels. This variable can represent a behavioral Verilog design for function simulation or gate-level Verilog design for structural or timing simulation.

The <verilog_switch_1> and <verilog_switch_n> variables represent Verilog switches that you can add to your command line during simulation. Refer to "Verilog Switches" on page 20 for information about available Verilog switches.

# 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. **Create or modify the testbench.** Make sure your testbench has a timescale definition added to it. The following is an example timescale definition:

   ```
   'timescale 1ns/100ps
   ```

   Refer to "Example Testbench" on page 15 for information about creating testbenches.

2. **Create or modify the command file.** A command file is only necessary if you are running batch simulation. Refer to "Example Command File" on page 15 for information.

3. **Simulate the design.** If your design is a Verilog HDL design, make sure that you simulate your behavioral Verilog HDL source file. Invoke the Verilog simulator by typing the following command:

   ```
   <verilog_executable> <test_bench>.v <design_name>.v -v
   $ALSDIR/lib/vlog/<act_fam>/<act_fam>.v
   ```

   If you are using the migration libraries, type the following command:

   ```
   <verilog_executable> <test_bench>.v <design_name>.v -v
   $ALSDIR/lib/vlog/<act_fam>.v -v $ALSDIR/lib/vlog/<act_fam>_mig.v
   ```

   The "-v" option is a Verilog switch that you can add to your command line during simulation. Refer to "Verilog Switches" on page 20 for information.

### *To simulate a design using a command file,*

invoke the Verilog simulator by typing the following command:

```
  <verilog_executable> -f <command_file>
```

The “-f” switch is necessary if you are using a command file to simulate a design.

# Structural Simulation

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

1. **Create or modify the testbench.** Make sure your testbench has a timescale definition added to it. The following is an example timescale definition:

   ```
   'timescale 1ns/100ps
   ```

   Refer to “Example Testbench” on page 15 for information about creating testbenches.

2. **Create or modify the command file.** A command file is only necessary if you are running batch simulation. Refer to “Example Command File” on page 15 for information.

3. **Simulate the design.** If your design is a Verilog HDL design, make sure that you simulate the structural Verilog HDL netlist that you generated using Designer or the “edn2vlog” program. Invoke the Verilog simulator by typing the following command:

   ```
   <verilog_executable> <test_bench>.v <design_name>.v -v
   $ALSDIR/lib/vlog/<act_fam>.v
   ```

   If you are using the migrations libraries, type the following command:

   ```
   <verilog_executable> <test_bench>.v <design_name>.v -v
   $ALSDIR/lib/vlog/<act_fam> -v $ALSDIR/lib/vlog/<act_fam>_mig.v
   ```

   The “-v” option is a Verilog switch that you can add to your command line during simulation. Refer to “Verilog Switches” on page 20 for information.

### *To simulate a design using a command file,*

invoke the Verilog simulator by typing the following command:

```
<verilog_executable> -f <command_file>
```

The “-f” switch is necessary if you are using a command file to simulate a design.

# Timing Simulation

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

1. **Place-and-route your design in Designer.** Refer to the *Designer User's Guide* for information about placing-and-routing a design using Designer.
2. **Extract timing information for your design from Designer.** 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. **Create or modify a testbench.** Make sure your testbench has an "$sdf_annotate" construct in it. The following is an example testbench with such construct:

```
'timescale 1ns/100ps
module test;

//Inputs and outputs declaration
wire ............
reg .................

//Instantiate the top module of your design in the test module
<top_module> <instance_name> (.....Pin List...);
..........

//stimulus patterns
initial
begin
.....
end

//Invoke SDF routine to back annotate
initial
$sdf_annotate("<design_name>.sdf",<instance_name>);
endmodule
```

The "<instance_name>" variable is the top-level instance name.

Refer to "Example Testbench" on page 15 for information about creating testbenches.

If your design contains a PLL, you must use a 1ps timescale resolution, for example, 'timescale 1ns/ps.

4. **Create or modify the command file.** A command file is only necessary if you are running batch simulation. Refer to "Example Command File" on page 15 for information about creating command files.
5. **(VCS Only) Create a PLI table.** The PLI table is a text file that contains PLI commands for VCS. The following is an example PLI table called "sdf.tab" that uses the module "test" in the testbench example in step 3.

   ```
   $sdf_annotate call=sdf_annotate_call acc+=tchk,mp,mip,prx:test+
   ```

   Refer to the VCS documentation for information about creating a PLI table.
6. **Simulate the design.** If your design is a Verilog HDL design, make sure that you simulate the structural Verilog HDL netlist that you generated using Designer or the "edn2vlog" program. Invoke the Verilog simulator by typing the following command (for VCS):

   ```
   vcs <test_bench>.v <design_name>.v -v $ALSDIR/lib/vlog/<act_fam>.v -M -
   P sdf.tab
   ```

   If you are using the migrations libraries, type the following command:

   ```
   vcs <test_bench>.v <design_name>.v -v $ALSDIR/lib/vlog/<act_fam>.v -v
   $ALSDIR/lib/vlog/<act_fam>_mig.v -M -P sdf.tab
   ```

   For other simulators:

   ```
   <verilog_executable> <test_bench>.v <design_name>.v -v
   $ALSDIR/lib/vlog/<act_fam>.v
   ```

   If you are using the migrations libraries, type the following command:

   ```
   <verilog_executable> <test_bench>.v <design_name>.v -v
   $ALSDIR/lib/vlog/<act_fam>.v -v $ALSDIR/lib/vlog/<act_fam>_mig.v
   ```

   The "-v" option is a Verilog switch that you can add to your command line during simulation. Refer to "Verilog Switches" on page 20 for information.

### *To simulate a design using a command file,*

Invoke the Verilog simulator by typing the following command:

```
<verilog_executable> -f <command_file>
```

The "-f" switch is necessary if you are using a command file to simulate a design.

# Verilog Switches

This section defines and gives usage examples of some common Verilog switches for simulators that interpret design files or compile the files on-the-fly. Refer to the documentation included with your Verilog simulation tool for additional information about using switches during simulation.

## Minus Switches

Table 4-1 defines and gives usage examples of Verilog minus switches.

Table 4-1. Minus Switches

| Switch | Definition |
|---|---|
| -s | Stop option; initiates entry into interactive mode after successful design compilation. |
| -a | Accelerated option; directs accelerated, declared elements to simulate in accelerated mode (Verilog-XL only). |
| -c | Compile only option; compiles the text code in a data file and exits the simulation mode. |
| -d | Decompile option; retargets data files into existing text files. |
| -f | Command argument file option; reads invocation command from a text file. |
| -l | Log file option. |
| -y | Library directory option; specifies a target library directory. |
| -v | Specifies the library file. |

## Plus Switches

Table 4-2 defines and gives usage examples of Verilog plus switches.

Table 4-2. Plus Switches

| Switch | Definition |
|---|---|
| +libext+ | Used with -y switch. |

Table 4-2. Plus Switches  (Continued)

| Switch | Definition |
|---|---|
| +delay_mode_path | Specifies the path delay model for simulation. |
| +delay_mode_unit | Specifies the unit delay model for simulation. |
| +delay_mode_zero | Functional simulation option; specifies the zero delay model for simulation. |
| +mindelays | Back-annotation option; selects minimum delay for simulation. |
| +maxdelays | Back-annotation option; selects maximum delay for simulation. |
| +typdelays | Back-annotation option; selects typical delay for simulation. |
| +transport_int_delays | Considers the interconnect delays as transport delays instead of inertial; needed with Axcelerator and ProASIC3/E library. |

5

# Simulation with ModelSim

This chapter describes the procedures for performing functional (behavioral and structural) and timing simulation on an Actel design using commands for the ModelSim simulator. Refer to the documentation included with your simulation tool for information about simulating a design using the graphical user interface.

## Behavioral Simulation

Use the following procedure to perform a behavioral simulation of a design using the ModelSim simulator. Type UNIX commands at the UNIX prompt. Type PC commands on the command line of the ModelSim Transcript window. The commands below are for PC. To make the commands work for UNIX, use forward slashes instead of back slashes.

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

   ```
   cd <project_dir>
   ```

3. **Create a "work" directory.** Type the following command:

   ```
   vlib work

   vmap work ./work
   ```

4. **Compile your design source and testbench file(s).** Before simulating your design, you must compile the source files and testbench. For hierarchical designs, compile the lower-level design blocks before the higher-level design blocks. Type the following commands:

   ```
   vlog <behavioral_design_file>.v
   vlog <test_bench>.v
   ```

5. **Simulate your design.** Type the following command:

   ```
   vsim <topmost_module_name>
   ```

   For example:

   ```
   vsim test_adder_behave
   ```

   The module test_adder_behave in the testbench will be simulated.

If any Actel macros are instantiated in your Verilog source, use the following command to simulate your design with the compiled Actel Verilog library.

```
vsim -L $ALSDIR\lib\vlog\mti\<act_fam> <topmost_module_name>
```

# Structural Simulation

Use the following procedure to perform a structural simulation of a design using the ModelSim simulator. Type UNIX commands at the UNIX prompt. Type PC commands on the command line of the ModelSim Transcript window. The commands below are for PC. To make the commands work for UNIX, use forward slashes instead of back slashes.

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

   ```
   cd <project_dir>
   ```

3. **Create a "work" directory.** You only need to create a work directory if you are using a different project directory than the one you used for behavioral simulation. Type the following command:

   ```
   vlib work
   ```

4. **Compile the structural netlist and testbench.** If you have not already generated a structural Verilog netlist, go to "Generating a Structural Verilog Netlist" on page 13 for the procedure. Type the following commands:

   ```
   vlog <structural_netlist>.v
   vlog <test_bench>.v
   ```

5. **Simulate your design.** Type the following commands:

   ```
   vsim -L $ALSDIR\lib\vlog\mti\<act_fam> <topmost_module_name>
   ```

   For example:

   ```
   vsim -L $ALSDIR\lib\vlog\mti\42mx test_adder_structure
   ```

   The module test_adder_structure in the testbench will be simulated using the compiled 42MX Verilog library.

# Timing Simulation

Use the following procedure to perform a timing simulation of a design using the ModelSim simulator. Type UNIX commands at the UNIX prompt. Type PC commands on the command line of the ModelSim Transcript window. The commands below are for PC. To make the commands work for UNIX, use forward slashes instead of back slashes.

1. **Place-and-route your design in Designer.** Refer to the *Designer User's Guide* for information about placing-and-routing a design using Designer.
2. **Extract timing information for your design from Designer.** Choose the Export command from the File menu or click Back Annotate. The Extract dialog is displayed. Create a <design_name>.sdf file by choosing the SDF option from the CAE pull-down menu. Click OK.
3. **Invoke the simulator (PC only).**
4. **Change the directory to your project directory.** This directory must include your Verilog design files and testbench. Type the following command:

   ```
   cd <project_dir>
   ```

5. **Create a "work" directory.** You only need to create a work directory if you are using a different project directory than the one you used for behavioral and structural simulation. Type the following command:

   ```
   vlib work
   ```

6. **Compile the structural netlist and testbench.** If you have not already generated a structural Verilog netlist, go to "Generating a Structural Verilog Netlist" on page 13 for the procedure. Type the following commands:

   ```
   vlog <structural_netlist>.v
   vlog <test_bench>.v
   ```

7. **Simulate your design using timing information contained in the SDF file.** Type the following command:

   ```
   vsim -L $ALSDIR\lib\vlog\mti\<act_fam> -sdf[max|typ|min]
   /<region>=<design name>.sdf -c <topmost_module_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 -L $ALSDIR\lib\vlog\mti\42mx -sdfmax /uut=adder.sdf -c
   ```

```
test_adder_structural
```

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

# 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

### A
Accessing
- Migration Libraries 7
- Verilog Libraries 7

Actel
- web site 27
- web-based technical support 27

Assumptions 5

### B
Back-Annotating Delays 18
Behavioral Simulation 10, 16, 23

### C
Capturing a Design 10
Command File 15
Compiling Verilog Libraries 7
Contacting Actel
- customer service 27
- electronic mail 28
- telephone 28
- web-based technical support 27

Conventions 6
- Document 6

Creating
- Command File 15
- SDF File 18–26
- Stimulus File 15

Creating Testbench 15
Customer service 27

### D
Delays 18
Design Creation/Verification 10
- Behavioral Simulation 10
- Design Capture 10
- EDIF Netlist Generation 11
- Structural Netlist Generation 11
- Structural Simulation 11
- Synthesis 11

Design Flow 10
- Programming 12
- System Verification 12

Design Implementation 11
- Place-and-Route 11
- Timing Simulation 12

Design Layout 11
Design Synthesis 11
Designer
- EDIF Option 13
- Extracting Timing Information 18, 26
- GENERIC Option 13
- Place-and-Route 11
- Software Installation Directory 5
- Structural Netlist Generation 13
- Verilog Option 11, 13

Device
- Programming 12
- Verification 12

Document
- Assumptions 5
- Organization 5

Document Conventions 6

### E
EDIF Netlist Generation 11, 13
EDIF Option 13
edn2vlog 13
Electronic mail 28
Example Testbench on page 11 16
Extracting Timing Information 18–26

F
File
  Command 15
  SDF 18
  Stimulus 15
Functional Simulation 16, 23

G
Gate-Level Netlist 11
Generating
  EDIF Netlist 11, 13
  Gate-Level Netlist 11
  Structural Netlist 11, 13
GENERIC Option 13

I
Installation Directory
  Designer 5
  Migration Libraries 7
  Verilog Libraries 7

L
Libraries
  Compiling 7
  Migration 7
  Verilog 7

M
Migration Library Installation Directory 7
Minus Switches 20

N
Netlist Generation
  EDIF 11
  Gate-Level 11
  Structural 11

O
Online Help 6

P
Place-and-Route 11
Plus Switches 20
Product Support 27–28
Product support
  customer service 27
  electronic mail 28
  technical support 27
  web site 27
Programming a Device 12

R
Required Software 7

S
Schematic Capture 10
SDF file 18–26
sdf_annotate 18
Setup Procedures
  User Setup 7
Simulation
  Behavioral 10, 16, 23
  Functional 16, 23
  Model Technology 23–26
  SilosIII 15–19
  Structural 11, 17, 24
  Synthesis-Based 10, 11
  Timing 12, 18, 25
  VCS 15–19
  Verilog-XL 15–19
Software Requirements 7

Stimulus File 15
Structural Netlist Generation 11
  Designer 13
  EDIF Option 13
  edn2vlog 13
  GENERIC Option 13
Structural Simulation 17
  Model Technology V-System Simulator 24
  MTI 24
  Verilog 11
Switches 20
  Minus 20
  Plus 20
Synthesis 11
System Requirements 7
System Verification 12
  Silicon Explorer 12

*T*

Testbench 15
Timescale Definition 16, 17
Timing Information 18–26
  SDF File 18–26
Timing Simulation 12, 18
  MTI 25

*U*

Unit Delays 10
User Setup 7

*V*

Verilog 10
  Back-Annotating Delays 18
  Behavioral Simulation 10
  Libraries 7
  Option 13
  Option in Designer 11
  Structural Simulation 11
  Switches 20
  Timing Simulation 12
Verilog Design Flow
  Design Implementation 11
  Programming 12
Verilog HDL Source Entry 10
Verilog Libraries
  Compiling 7
Verilog Library Installation Directory 7

*W*

Web-based technical support 27

***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__. • Dunlop House, Riverside Way • Camberley, Surrey GU15 3YL • United Kingdom*
*Phone +44 (0)1276.401452 • Fax +44 (0)1276.401490*

*__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*

*__Actel Hong Kong__ • 39th Floor, One Pacific Place • 88 Queensway, Admiralty Hong Kong*
*Phone +852.227.35712 Fax +852.227.35999*

Actel