Actel® Libero™Integrated Design Environment v2.3
Structural Schematic Flow Design Tutorial

Table of Contents

1

Design Flow in Libero™ IDE v2.3

This chapter describes the basic design flow for creating designs using the Libero IDE software.

Step 1 - Design Creation

Your first step is to plan your design and enter it as either HDL (VHDL or Verilog), structural schematic, or mixed-mode (schematic and RTL).

Step 2 - Design Verification

After you have defined your design, you must verify that it functions the way you intended. After creating a test bench using WaveFormer Lite use the ModelSim for Actel VHDL or Verilog simulator to perform functional simulation on your schematic or HDL design.

Step 3 - Design Synthesis/EDIF Generation

A design must be synthesized if the design was created using VHDL or Verilog. Use Synplify or Synplify Lite from Synplicity to generate your EDIF netlist. You can re-verify your design "post-synthesis" using the VHDL or Verilog ModelSim for Actel simulator used in step 2. While all RTL code must be synthesized, pure schematic designs are automatically "netlisted" out via the Libero IDE tools to create a structural VHDL or structural Verilog netlist.

Step 4 - Design Implementation

After you have functionally verified that your design works, the next step is to implement the design using the Actel Designer software. The Designer software automatically places and routes the design and returns timing information. Use the tools that come with Designer to further optimize your design. Use Timer to perform static timing analysis on your design, ChipEdit to customize your I/O macro placement, PinEdit for I/O customization, SmartPower for power analysis, and Netlist Viewer to view your netlist.

Step 5 - Timing Simulation

After you are done with design Implementation, you can verify that your design meets timing specifications. After creating a test bench using WaveFormer Lite, use the ModelSim for Actel VHDL or Verilog simulator to perform timing simulation.

Step 6 - Device Programming

Once you have completed your design, and you are satisfied with the timing simulation, create your programming file. Depending upon your device family, you need to generate a Fuse, Bitstream, or STAPL programming file.

2

Schematic Flow in Libero™ IDE v2.3

Example 1 demonstrates a structural schematic design flow in Libero IDE. You should review each of the steps to become familiar with the schematic design flows and also work through them.

Example 1

A two input multiplexor in Figure 1 has been implemented in the Libero IDE schematic entry tool using symbols from the Actel "eX" symbol library. Using Actel's Libero IDE tool suite, you will complete the following:

- Create a Libero IDE project
- Use Libero's schematic editor to enter the two input mux schematic
- Functionally simulate in the ModelSim for Actel HDL simulator
- Implement the design with Actel's Designer software
- Timing simulation with back annotated timing in the ModelSim for Actel HDL simulator

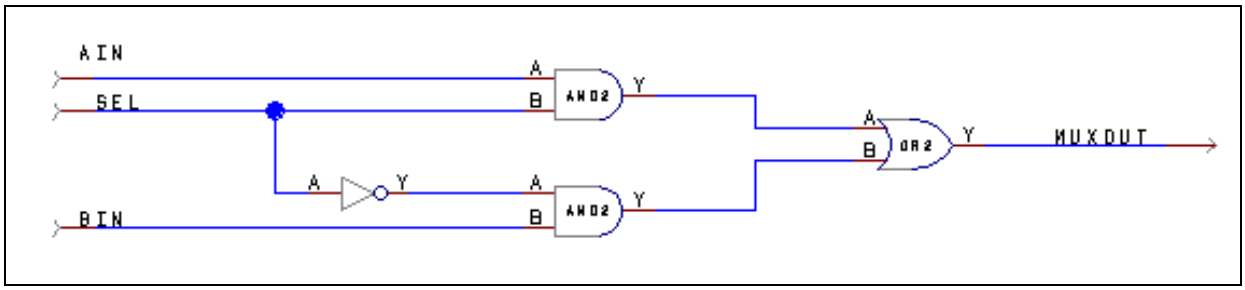

Figure 1 Libero IDE Schematic (two input mux)

Step 1 – Creating A Libero IDE Project

1. Start Libero by double-clicking on the Actel Libero IDE icon.
2. From the File menu click *New.* The New Project dialog box appears, as shown in Figure 2. Enter the following in the New Project dialog box:

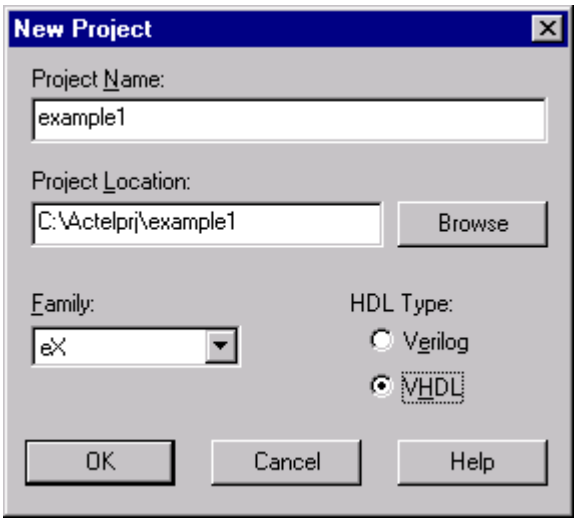

Figure 2. Libero's New Project dialog box

- **Family:** Select eX from the Family drop-down list box
- **HDL:** Select VHDL or Verilog as the HDL
- **Location:** Specify a location in the Location field, or select a location by clicking Browse and browsing the project directory
- **Project Name:** Enter example1 in the Project Name field

3. Click ***OK.*** The project "example 1" is created and opened in Libero.

Step 2 - Design Creation - Entering a Schematic

Using Libero IDE, create the two input mux schematic.

To create the two input mux schematic:

1. From the File menu, click *New* again or double-click on *ViewDraw* in the process window. The New File dialog box is displayed Figure 3).

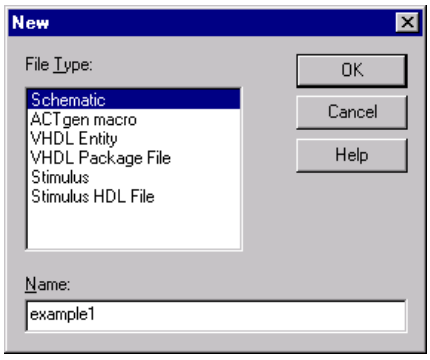

Figure 3. Libero's New File dialog box

Select Schematic in the New dialog box and specify the name. Click OK. ViewDraw opens in a separate window.

2. From ViewDraw, choose *Add* from the Component menu. The Add Component dialog box appears as in Figure 4.

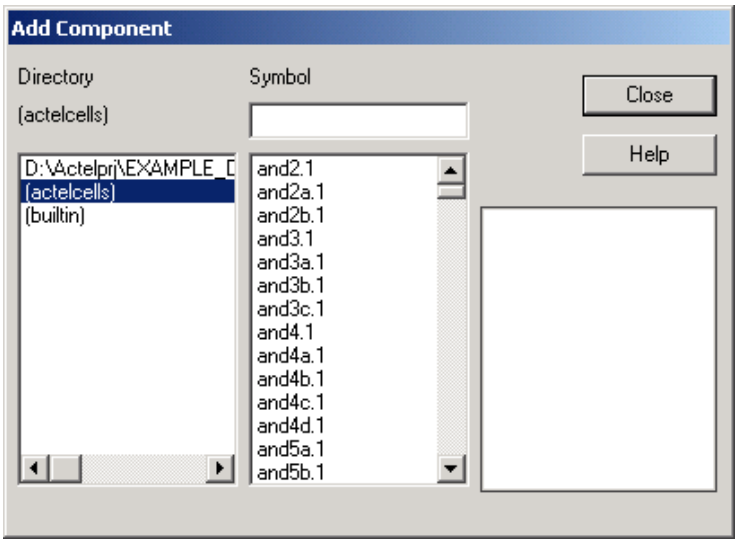

Figure 4. ViewDraw's Add Component dialog box

3. For basic cell, click on "actelcells." A list of cells will appear in the middle column.

4. Begin creating the schematic shown in Figure 1 by placing the components onto your schematic.

   The symbols you need for this project are *AND2, INV, OR2, INBUF, OUTBUF* and the hierarchical connectors for the I/O's. The hierarchical connectors are located in the "built-in" directory while the rest of the cells can be found in the "actelcells" directory. Select and drag each component from the list or viewer screen, and drop it in place on the schematic sheet. For more information, refer to the Viewdraw Online help.

   Note: INBUF and OUTBUF are required in schematic designs to define I/O. (Please refer to the Actel Macro Guide for more information on other types of I/O's.) If you don't wish to place I/O's in the schematic manually, Libero IDE can insert them automatically via the **"Optimize and Insert Pads"** feature which invokes Synplicity. It is important to note that Synplicity will not synthesize the ports correctly in the netlist unless you have added hierarchical connectors to your design. Therefore, add the hierarchical connectors.

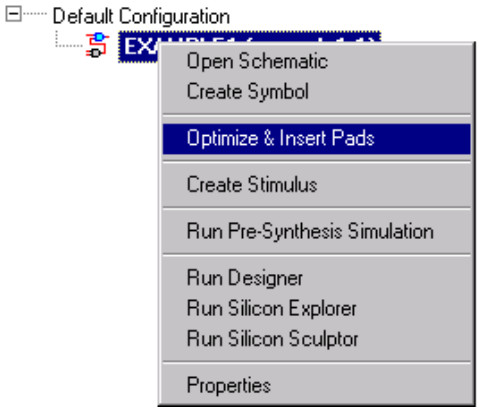

Figure 5 Invoking "Optimize and Insert Pads"

5. **Connect the components with Net and/or Bus.** From the Add menu, click Net/Bus or use toolbar. For more Net/Bus connection procedures, refer to Viewdraw Online help.

6. **Add I/O names to your design.** Double click in net/bus of the schematic drawing to bring up the Net Properties window (Figure 6).

   In the drop-down edit field, type in the name you want to assign the I/O. Click *OK*. The name you enter will appear on the schematic. The following names should be assigned, as shown in the schematic in Figure 1 on page 4.
   - **Inputs:** AIN, BIN, SEL
   - **Outputs:** MUXOUT

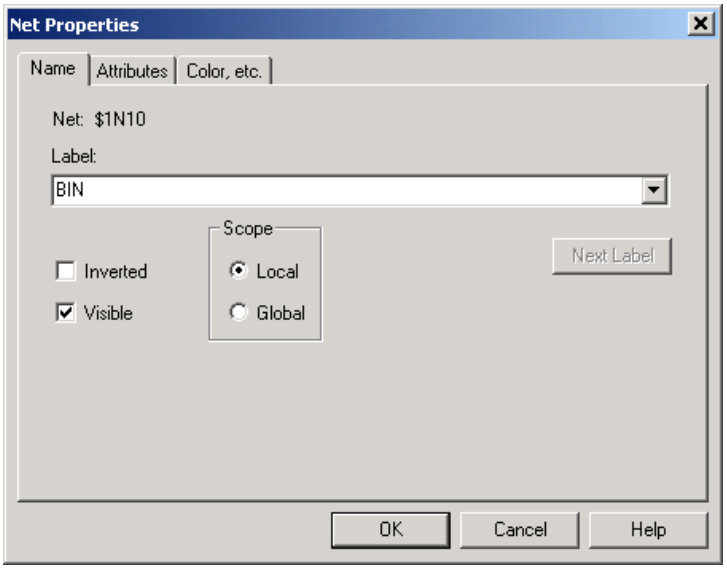

Figure 6. Viewdraw's Net Properties dialog box.

7. From the File menu, click *Save+Check* to save and to show any errors or warnings in your schematic. Correct any errors that are indicated.

8. From the File menu, click *Exit* to exit the schematic capture. The schematic is saved as example1.

   Note: If example1 does not appear in bold in the design hierarchy browser then right click it and select *Set As Root*.

9. From the File Manager tab within the Libero IDE, you also have the option of performing some additional checks to your schematic by right clicking the schematic and selecting *Check Schematic*..

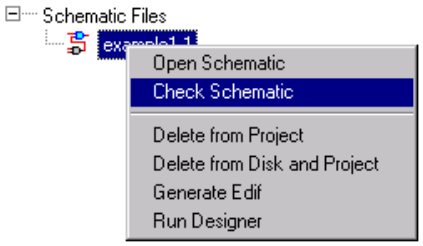

Figure 7 Invoking "Check Schematic" from Libero

Step 3 - Design Verification - Functional Simulation

When you open ModelSim for Actel within the Libero IDE for the first time in the design flow, a structural netlist is created that you can simulate. The ModelSim for Actel simulator uses pre-compiled libraries that are installed from the Libero CD.

The steps to simulate the design are:

- Create a testbench using WaveFormer Lite or a text editor
- Simulate the design with the ModelSim for Actel simulator

Creating stimulus with WaveFormer Lite

Follow the steps below to create stimulus using WaveFormer Lite.

1. From the Libero IDE Design Hierarchy tab, double click the left mouse button on the WaveFormer Lite Stimulus icon.
2. The Graphical Test Bench Generator will display the signals you assigned to your design in the schematic (see Figure 8).

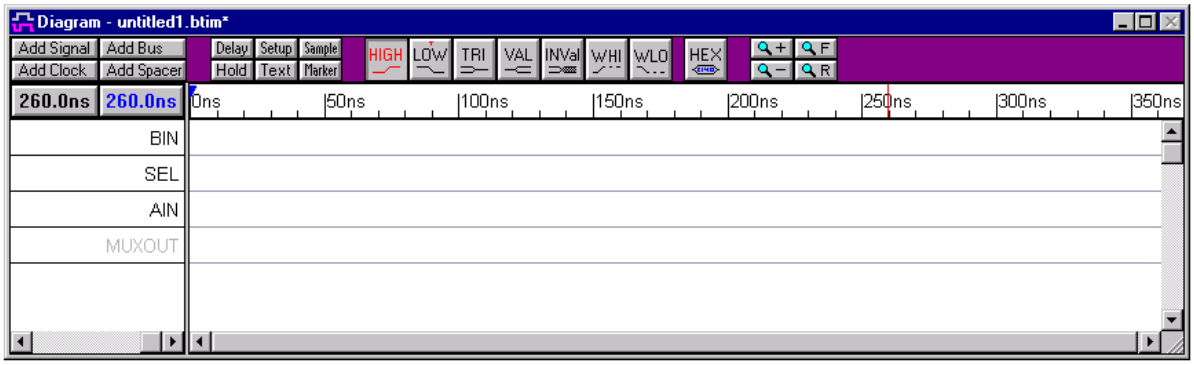

Figure 8 - WaveFormer Lite Diagram window

3. The WaveFormer Lite Diagram editor controls for zooming, adding signals and segment levels are shown in Figure 9.

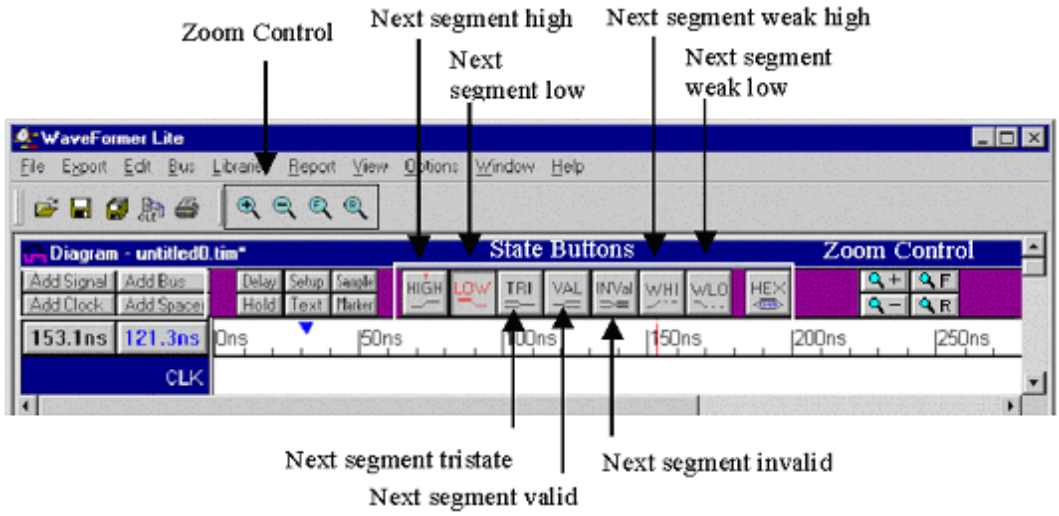

Figure 9- WaveFormer Lite Diagram editor controls

Draw a waveform in the Timing Diagram Editor for a signal:

1. Select the state of the signal you wish to draw by clicking on one of the state values in the toolbar. To draw a '1' value for a signal, select the HIGH button. Place the mouse cursor inside the diagram window in the same row as the signal name. Then, position the mouse cursor at the time value you want the selected state to end. Left click the mouse to draw a waveform from the end of the previously existing signal (if there is any) to the mouse cursor position.

2. Move the mouse to the right and left click again to draw another segment. Click on a state button to activate it. The state buttons automatically toggle between the two most recently activated states. This allows you to draw alternating High and Low values easily without having to select the toolbar state each time. Using the steps above, create stimulus for AIN, BIN and SEL as shown in Figure 10.

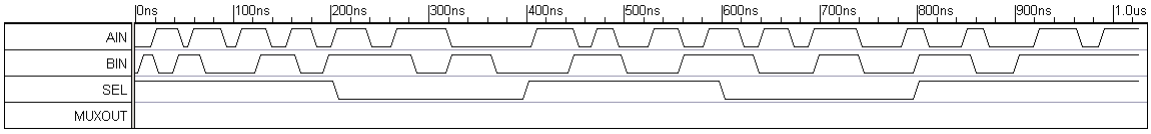

Figure 10 - Timing diagram for example1

3. After creating the all waveforms successfully, select Save As from the File menu to save the waveforms. The Save As dialog box is displayed (Figure 11). Enter example1.btim as the file name and click Save.

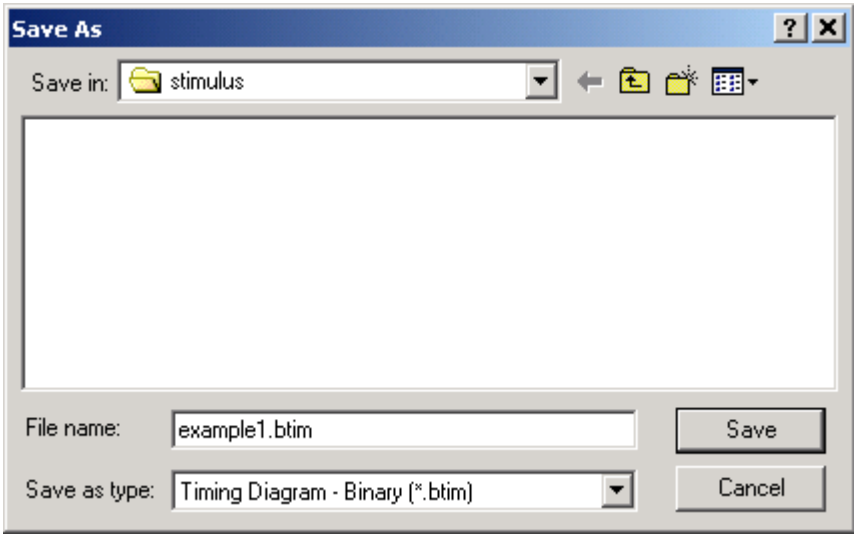

Figure 11 - Save As dialog box

4. After saving the timing diagram file, click the Export in the Wave Former Lite menu. From the Export menu, click *Export Timing Diagram As*. Save the stimulus file as either VHDL or Verilog (according to the netlist format you exported from Designer).

   For VHDL, select "VHDL Wait with Top Level Testbench" in files of type and enter "example1.vhd" for the file name (Figure 12). Click Save to generate the testbench. For Verilog, select "Verilog with Top Level Testbench" in files of type and enter "example1.v" for the file name (Figure 13). Click Save to generate the testbench.

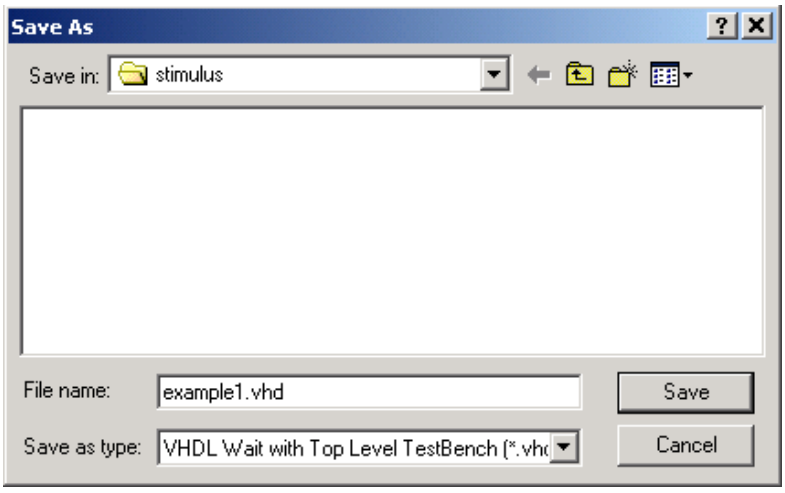

Figure 12 - Exporting a VHDL testbench

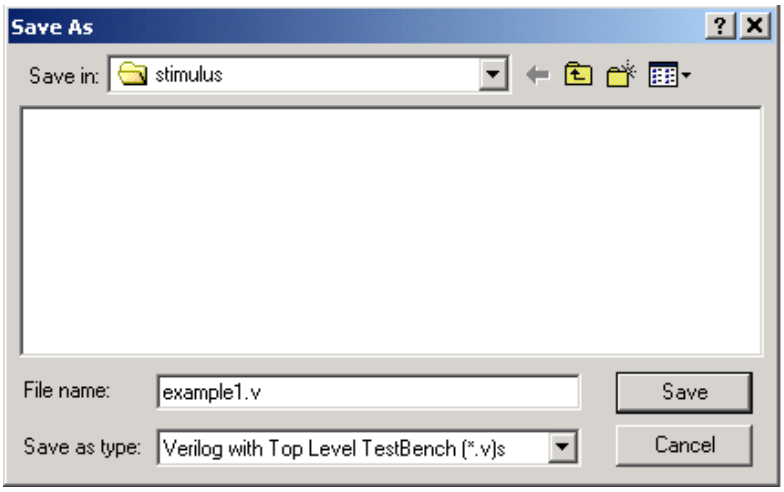

Figure 13 - Exporting a Verilog testbench

5. The testbench with a component declaration and instantiation inside can be viewed in the WaveFormer Lite Report window.

6. Exit WaveFormer Lite (File > Exit).

7. The stimulus files are visible on the Libero IDE File Manager tab (Figure 14).

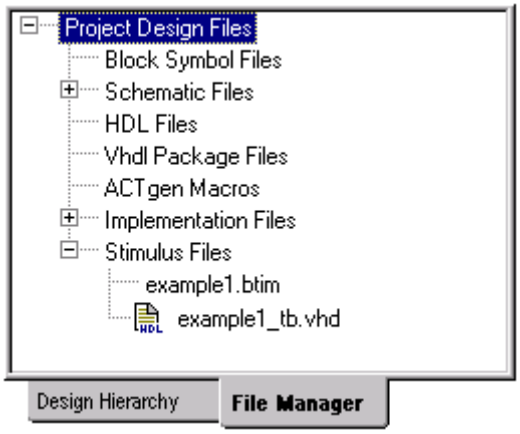

Figure 14 - Stimulus files in Libero IDE

Your design is ready to simulate with ModelSim for Actel.

Simulating with the ModelSim for Actel simulator:

1. Invoke the ModelSim for Actel simulator by double clicking the **ModelSim for Actel Simulation** button in the Libero IDE Process window or by selecting example1 in the Libero IDE Design Hierarchy tab then right clicking and selecting *Run Pre-Synthesis Simulation* (Figure 15). Note: if you ran *Optimize and Insert Pads* to insert IO's into your design, then you will choose *Run Post-Synthesis Simulation* from the list of options.

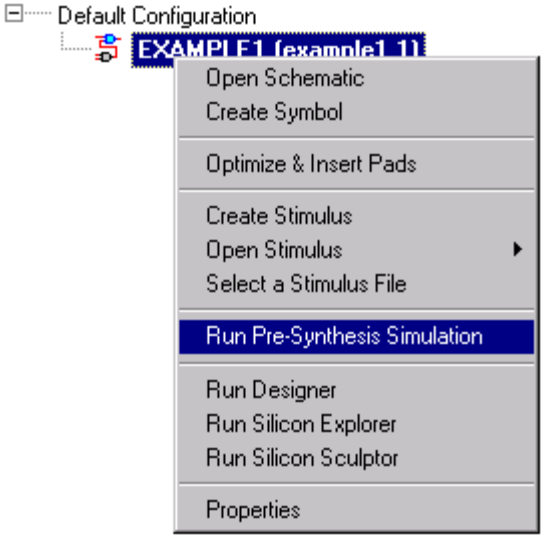

Figure 15 - Invoking the ModelSim for Actel Simulator for example1

2. Libero IDE will prompt you to associate a stimulus file. Select the testbench you just created using Waveformer Lite and add it to the "Associated Files" column.

3. The ModelSim for Actel VHDL or Verilog Simulator will open and compile the source files (Figure 16).

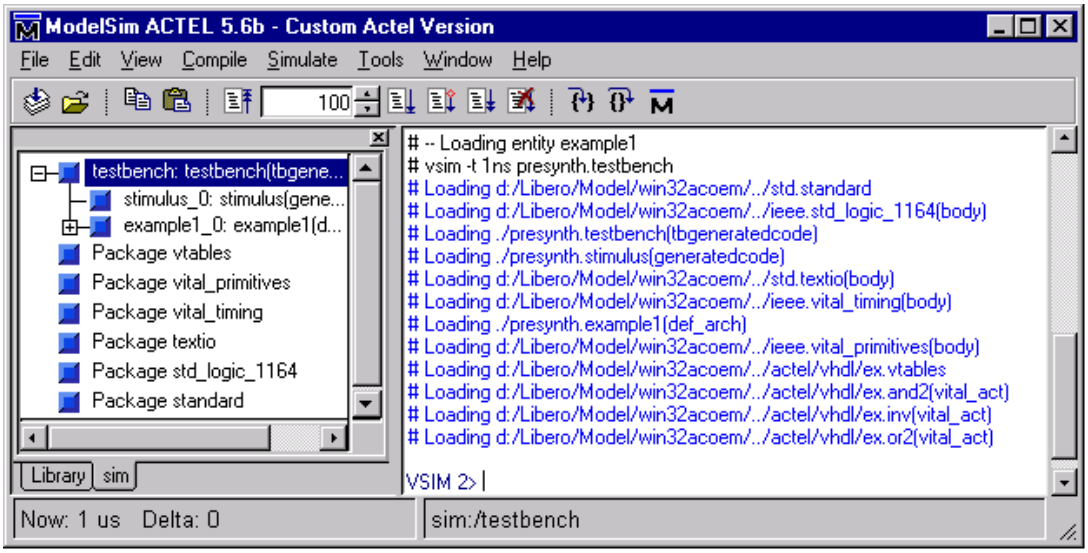

Figure 16 - ModelSim for Actel main Window after compiling

4. When the compilation completes, the simulator will run for 1 us and a Wave window will open to display the simulation results. (Figure 17).

5. Scroll in the wave window to verify the multiplexor works correctly. Use the zoom buttons to zoom in and out as necessary (Figure 18).

6. Exit the simulator by selecting Quit from the File menu in the ModelSim for Actel main window.

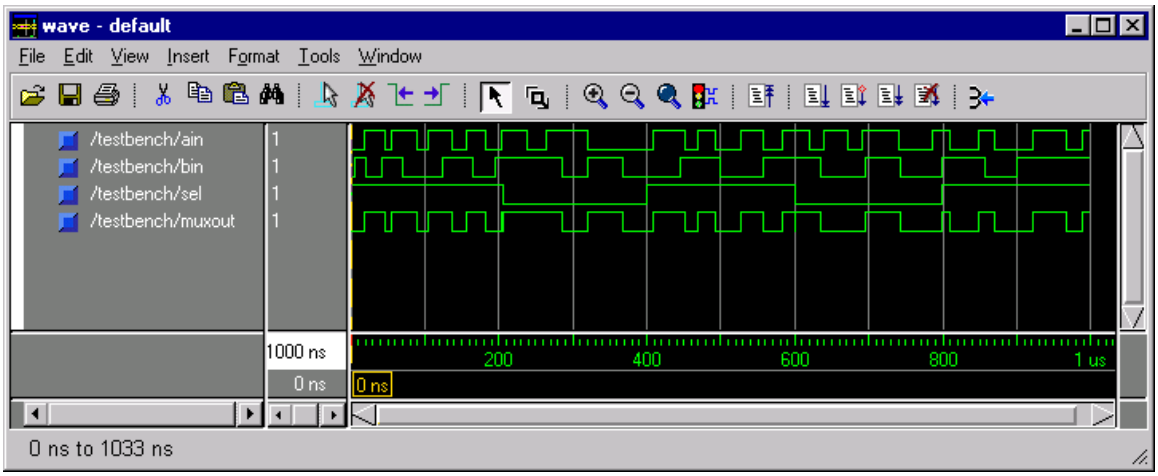

Figure 17 - ModelSim for Actel Wave Window for example1 design

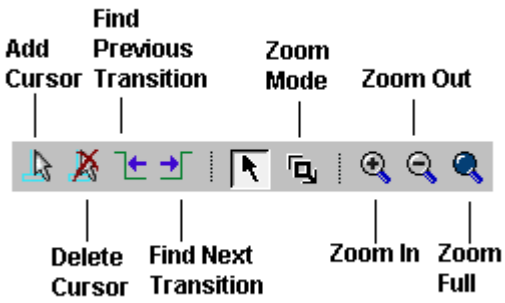

Figure 18 - ModelSim for Actel zoom controls

Step 4 - Design Implementation - Place and Route

Implement the design using Actel's Designer software.

1. **Run Designer.** Select the Design Hierarchy tab in Libero IDE. Right-click on example1 and select *Run Designer*. Actel's Designer application will open and your design file will be read in.
2. **Compile your design.** From the Tools menu, click *Compile*. A series of menus will query you on device type, device package, speed grade, voltage, and operating conditions. Select **eX64** in the Die window and **49CS** in the Package window. Click on **Next** and **Finish** to complete the remaining steps. Once you have finished the setup, Designer will compile your design and show you the utilization of the selected device. Also note that the Compile box in Designer will turn green indicating that compile has successfully completed.
3. **(Optional) Designer User Tools.** Once you have successfully compiled your design, you can assign pins with PinEdit, view pre-layout static timing analysis with Timer, set timing constraints in Timer, and use ChipEdit to assign modules. You can use each of these functions by left clicking on the flow tree. For more information on these functions please refer to the Designer User's Guide and online help.
   For this example, we will make no changes to the design in this step.
4. **Lay Out your design.** From Designer, click on Layout. In the Layout dialog box click OK to accept standard layout. The Layout box in Designer will turn green indicating that layout has successfully completed.
5. **Back-Annotate your design.** From Designer, click on Back-Annotate, or select *Back-Annotate* from the Tools menu. Choose SDF as CAE type and appropriate simulation language. Click OK. Designer will export example1_ba.sdf file in this case. Ensure that **VHDL** (or Verilog) and **Export Netlist** are checked.
6. **Click on *Fuse* in the Designer flow tree if you wish to create a programming file for your design.** This step can be performed later after you are satisfied with the back-annotated timing simulation.

7. **Save and Close Designer.** From the File menu, click *Exit*. Select Yes to save the design before closing Designer. Designer saves all of the design information in a *.adb file. The file example1.adb appears under the Implementation Files of the File Manager tab in Libero IDE (Figure 19). You can re-open this file with by right clicking and selecting **Open in Designer** to launch the Designer application.

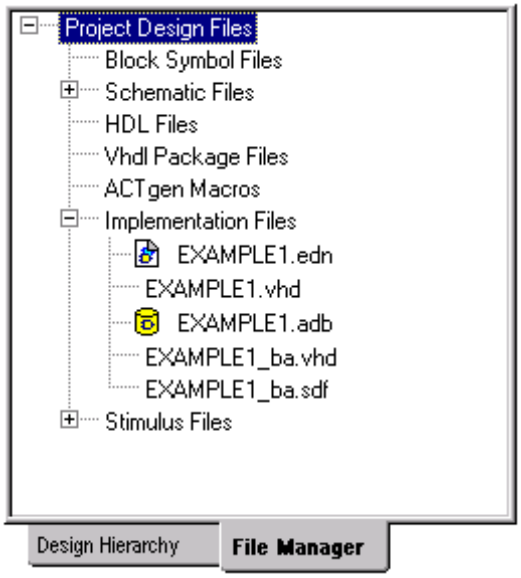

Figure 19 - Designer files in the Libero IDE File Manager tab

Step 5 – Back Annotated Timing Simulation with the ModelSim for Actel HDL Simulator

To perform timing simulation with back annotated timing in the ModelSim for Actel HDL simulator:
Invoke the ModelSim for Actel simulator by Double clicking the **ModelSim for Actel Simulation** button in the Libero IDE Process window or by selecting example1 in the Libero IDE Design Hierarchy tab then right clicking and selecting *Run Post-layout Simulation*.

1. The ModelSim for Actel VHDL (or Verilog) Simulator will open and compile the source files.
2. When the compilation completes, the simulator will run for 1 us and a Wave window will open to display the simulation results.
3. Scroll in the wave window to verify the multiplexer works correctly. Use the cursor and zoom buttons to zoom in and out and measure timing delays.
4. Exit the simulator by selecting Quit from the File menu in the ModelSim for Actel main window.
5. Close the Libero IDE by selecting **File > Exit**.

End of tutorial design

APPENDIX - Waveformer Lite Tutorial

Creating stimulus with WaveFormer Lite:

Follow the steps below to create stimulus for using WaveFormer Lite.

1. From the Libero IDE Design Hierarchy tab, double click the left mouse button on the WaveFormer Lite Stimulus icon.
2. The Graphical Test Bench Generator will display the signals you assigned to your design in the schematic.

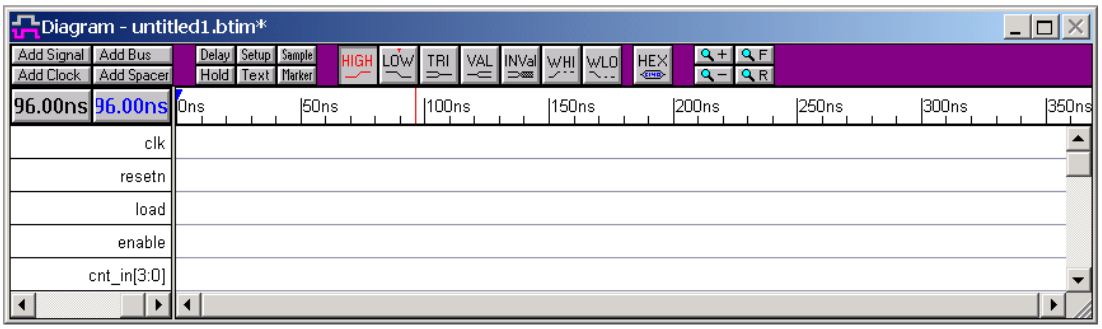

Figure A1 - WaveFormer Lite Diagram window for example2

To draw a waveform in the Timing Diagram Editor for a Clock signal:

1. Create a clock by right clicking CLK and then selecting *Signal(s)* <-> *Clock(s)* to convert CLK from signal to clock (ie. periodic) format. (Figure A2).
2. Double-click CLK again to bring up the Signal Properties menu, and then select the *Clock Properties* sub-menu. Define the clock frequency.
3. Specify the Frequency or Period and a Duty Cycle to generate clock signals. Click **OK** to create the clock waveform.

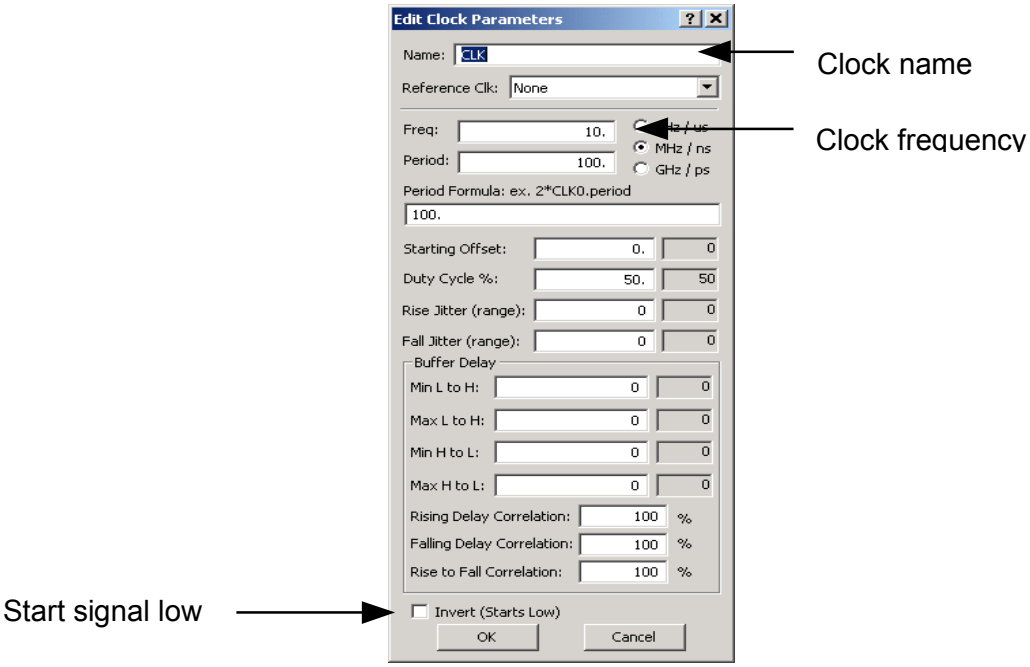

Figure A2 - Edit Clock Parameters window

To draw a waveform in the Timing Diagram Editor for a bus signal:

1. Select the state (value) of the signal you wish to draw by clicking on one of the state values in the toolbar. For example, to draw a valid bus value, select the VAL button. The selected state button turns red (see Figure A3).

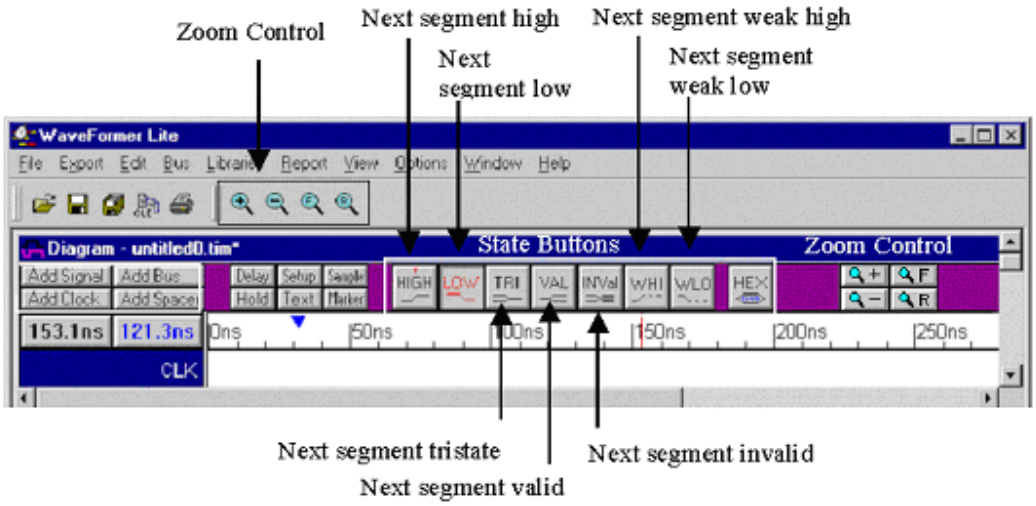

Figure A3 - WaveFormer Lite Diagram editor controls

2. Place the mouse cursor inside the diagram window in the same row as the bus name and position the mouse cursor at the time value where you want the selected state to end. Left click the mouse to draw a waveform from the end of the previously existing signal (if there is any) to the mouse cursor position.

3. Move the mouse to the right and click again to draw another segment. Click on a bus state button (TRI, VAL or INVal) to activate it. The state buttons automatically toggle between the two most recently activated states. The waveform to be drawn next will be determined by the button that was selected (highlighted red). Double-click on a button to prevent it from toggling.

   When you draw signals using the mouse, the signal edges are automatically aligned to the closest edge grid time. To fine tune and make minor adjustments to signal transition times, position the cursor over the transition until it changes to a bolded bi-directional arrow. Left click and drag the transition edge either left or right in time.

To draw a waveform in the Timing Diagram Editor for a regular signal:

4. Select the state of the signal you wish to draw by clicking on one of the state values in the toolbar. To draw a '1' value for a signal, select the HIGH button. Place the mouse cursor inside the diagram window in the same row as the signal name. Then, position the mouse cursor at the time value where you want the selected state to end. Left click the mouse to draw a waveform from the end of the previously existing signal (if there is any) to the mouse cursor position.

5. Move the mouse to the right and left click again to draw another segment. Click on a state button to activate it. The state buttons automatically toggle between the two most recently activated states. This allows you to draw alternating High and Low values easily without having to select the toolbar state each time.

Copying Waveforms

You can copy sections of waveforms and paste those sections either onto (overwrite) or into (insert) any signal in the diagram. To copy and paste waveform sections:

1. Select the names of the signals that you want to copy. If no signals are selected, the Block Copy command will select all the signals in the diagram.
2. Choose the Edit > Block Copy Waveforms menu option. This opens the Block Copy Waveforms dialog box (Figure A4), with the selected signals displayed in the Change Waveform Destination list box.
3. In the dialog, enter the values that define the copy and paste:

   - Choose either Time or Clock cycle for the base units of the dialog. If you are copying just signals (no clocks) then time is the default base unit of the dialog. If you are copying part of a clock then it is best to choose a clock cycles base unit and choose the copied clock as the reference clock. If you select time when copying clocks, the (end_time - start_time) must equal an integral number of clock periods, and the place_at time must be at the same clock period offset as the start_time.
   - Start and End define the times of the block copy.
   - Place At is the time that at which the block will be pasted.
   - The Insert and Overwrite radio buttons determine whether the paste block will be inserted into the existing waveforms or overwrite those waveforms.
   - The list box at the bottom of the dialog determines which signal the copied waveforms will be pasted into. To change this mapping:
     i) Select a line in the list box. This places the destination signal into the drop-down list box on top of the list box.
     ii) Choose another signal from the drop-down list box. Each destination signal can be used only once per copy.
     iii) Click OK to complete the copy and paste operation.

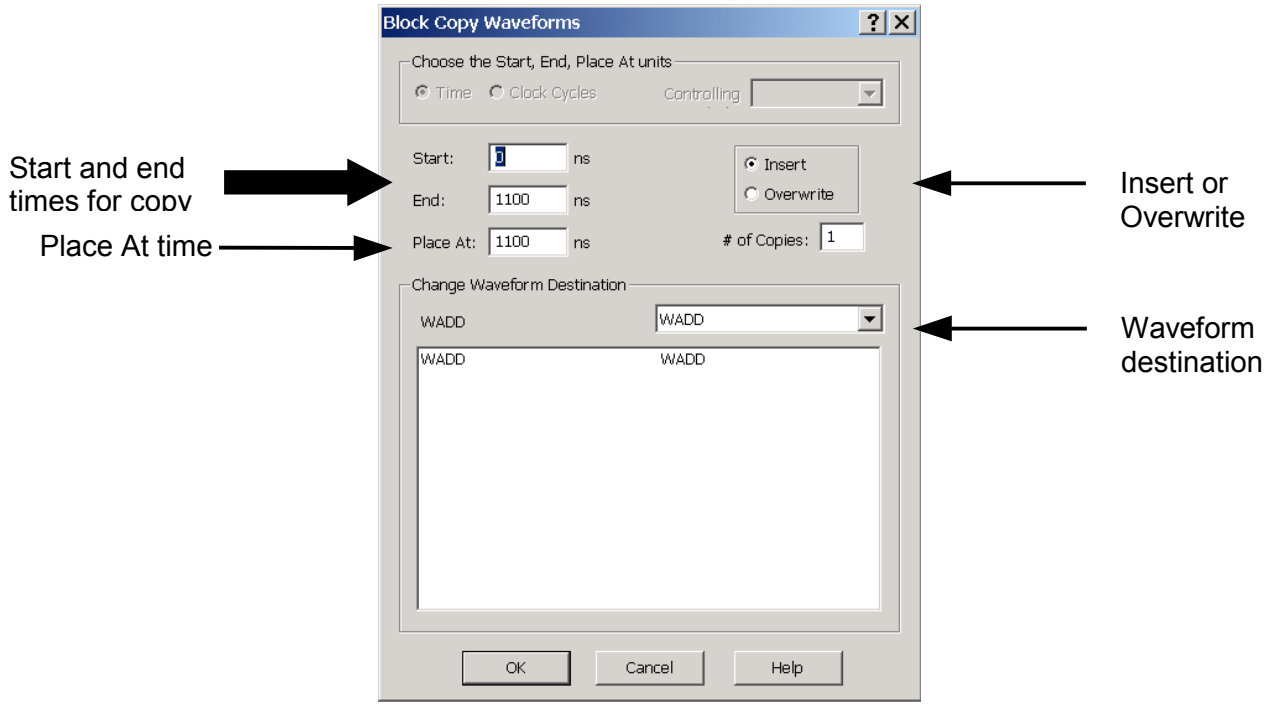

Figure A4 - Block Copy Waveforms dialog box