# Axcelerator Starter Kit

*User's Guide v1.1*


Actel®

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



Printed in the United States of America

Part Number: 50200037-1/02.05

Release: February 2005



## Trademarks

# Table of Contents

# Introduction

Thank you for purchasing the Actel Axcelerator (AX) Starter Kit.

This guide provides you with the information you need to easily evaluate Axcelerator devices.

## Document Organization

The Axcelerator Starter Kit contains the following chapters:

Chapter 1 – "Setup and Self-Test" describes how to setup the Axcelerator Evaluation Board and how to perform a self test.

Chapter 2 – "Hardware Description" describes the components on the Axcelerator Evaluation Board.

Chapter 3 – "Tutorial" contains a nine-step process that illustrates a basic Verilog design.

Appendix A – "Board Connections" contains a table listing the board connections.

Appendix B – "P160 Connections" contains a table listing the P160 connections.

Appendix C – "Board Schematic" shows the schematic of the Axcelerator Evaluation Board.

Appendix D – "Product Support" describes our support services.

## Document Assumptions

This user's guide assumes the following:

- You intend to use Actel Libero® Integrated Design Environment (IDE) software.
- You have installed and are familiar with Actel Libero IDE software.
- You are familiar with the Verilog hardware description language.
- You are familiar with UNIX workstations and operating systems or PCs and Windows® operating systems.

# 1

# Setup and Self-Test

This chapter describes how to set up the Axcelerator Evaluation Board and how to perform a self-test. Figure 1-1 shows an AX Starter Kit example design specification.

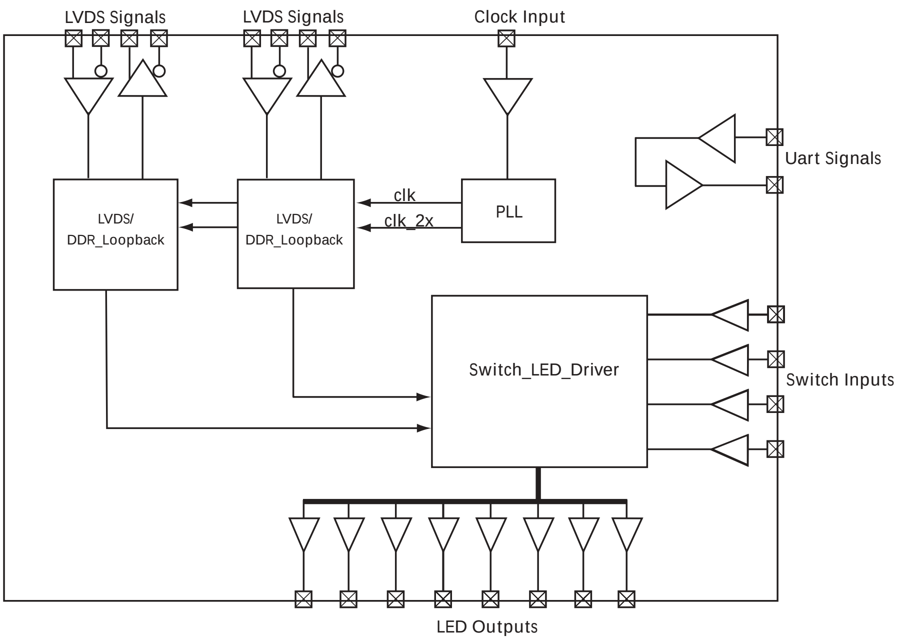


Figure 1-1. Design Specification

## LVDS DDR Loopback Circuit

The AX Starter Kit example design contains a self-checking serial loopback circuit. This circuit uses the AX chip's Double Data Rate (DDR) functionality to transmit and receive data at 80 MHz. This is double the 40 MHz clocking rate of the rest of the system. When functioning, the circuit periodically sends a ten-bit test pattern (**0101110011**) at 80 Megabits per second. See the enclosed Verilog files for an example of using DDR and Low Voltage Differential Signaling (LVDS) I/Os. If a cable is connected from the LVDS outputs to the LVDS inputs, the circuit runs and an LED begins blinking to verify that the data is being transmitted and received correctly. If the I/Os are not connected or if there is another problem, the LED will not light up. There are two identical LVDS/DDR loopback circuits in the example design. One displays its status on LED[8] and is wired to the

SMA connectors on the board. The other is wired only to jumpers, and its status is displayed on LED[7].

# Switches and LEDs Circuit

Table 1-1 will help in understanding the behavioral relationships between each of the switches and indications on the various LEDS.

Table 1-1. Switches and LEDs

| Switch | Description |
|---|---|
| SWITCH[1] | Asynchronously turns on LEDs[8:2] while held. Sets LEDs[6:2] to **01010** for a short period when released. |
| SWITCH[2] | Asynchronously turns off all LEDs[8:2] while held. Sets LEDs[6:2] to **10101** for a short period when released. |
| SWITCH[3] | Sets LEDs[6:2] to begin a standard binary counting pattern. Any other switch overrides the counting mode. (The behavior of SWITCH[3] is the default behavior if you do not press any switches.) |
| SWITCH[4] | Reset. Asynchronous reset for all logic. Also turns all LEDs[8:1] on while held. |

# Using and Testing the Example Design

This section describes using and testing the example design (AXEV_TOPPLACE_REV2_SEP28.pdf).

### *To verify that the design/board is programmed correctly, powered-up, and running:*

1. **Set up loopback on both LVDS (TX, RX) signal pairs via shunts placed on JP36 and JP37.**

   On header **JP36**:

   - Using a shunt, connect RX2P to TX2P; i.e., receive 2 positive to transmit 2 positive. (Connect header pin 2 to pin 1 on JP36.)
   - Using a shunt, connect RX2N to TX2N; i.e., receive 2 negative to transmit 2 negative. (Connect header pin 4 to pin 3 on header JP36.)

On header **JP37**:

- Using a shunt, connect RX1P to TX1P; i.e., receive 1 positive to transmit 1 positive. (Connect header pin 3 to pin 1 on JP37.)
- Using a shunt, connect RX1N to TX1N; i.e., receive 1 negative to transmit 1 negative. (Connect pin 4 to pin 2 on header JP37.)

2. **Using a multimeter, test the impedance between power and ground for each voltage plane.**

   The reading should be taken between the test points labelled 3.3 V, 2.5 V. 1.8 V and 1.5 V and the test points labelled ground. A high reading is to be expected for all power/ground planes. If a short is detected (indicated by a low impedance reading of less than 1 W), do not proceed further.

3. **Attach 9 V DC power to the Board at J1.**

   Plug the 9 V DC supply cable into the power plug on the back of the board.

4. **Press SW4 and verify that all LEDs go high (turn on).** Verify that LEDs[6:1] go low (turn off) when SW4 is released.

   Note: LEDs[8:7] may stay on.

5. **Press SW1 and verify that LEDs[8:2] turn on.** When you release SW1, verify that LEDs[6:2] display **01010** for a short time (**0** indicates the LED is off, **1** indicates the LED is on).
6. **Press SW2 and verify that LEDs[8:2] turn off.** When you release SW2, verify that LEDs[6:2] display **10101** for a short time.
7. **Press SW3 and verify that LEDs[6:2] begin counting in a standard binary counting sequence.**
8. **Provided loopback cables are connected to the SMA connectors, or loopback shunts are connected to the headers JP36 and JP37 (as in step 1), the LVDS loopback status may be observed.** Observe the LVDS/DDR loopback status by verifying that LED[8] and LED[7] are blinking.

Table 1-2 shows the Design Pin Description.

Table 1-2. Design Pin Description

| Name | Pin Number | Pin Type | Description |
|---|---|---|---|
| CLK_in | 76 | LVTTL Input | 40 MHz clock input to PLL. This will be used to generate 40 and 80 MHz internal clocks. |

Table 1-2. Design Pin Description (Continued)

| Name | Pin Number | Pin Type | Description |
|---|---|---|---|
| LED_out[1] | 123 | LVTTL Output | Driver for LED_1. This LED will be driven low unless the reset switch is active. In this case LED_1 will be driven high asynchronously. |
| LED_out[2] | 122 | LVTTL Output | Driver for LED_2. Asynchronously driven high by reset or SWITCH[1]. Asynchronously driven low by SWITCH[2]. If the clock circuit is functioning when the switch is released, the LED will be set as one of the following states.<br>SWITCH[1] – LEDs[6:2] display **01010** for approximately 1 second, then return to 0.<br>SWITCH[2] – LEDs[6:2] display **10101** for approximately 1 second then return to 0.<br>SWITCH[3] – LEDs[6:2] begin a counting pattern. Counting continues until any other switch is touched. |
| LED_out[3] | 121 | LVTTL Output | See LED_out[2] |
| LED_out[4] | 120 | LVTTL Output | See LED_out[2] |
| LED_out[5] | 109 | LVTTL Output | See LED_out[2] |
| LED_out[6] | 108 | LVTTL Output | See LED_out[2] |

Table 1-2. Design Pin Description (Continued)

| Name | Pin Number | Pin Type | Description |
|---|---|---|---|
| LED_out[7] | 107 | LVTTL Output | Driver for LED_7. This LED will be driven high asynchronously by either reset or SWITCH[1]. After reset the LED will remain on briefly, then act as the status light for the LVDS loopback circuit 2. If the loopback circuit is functioning properly, this LED will flash. Otherwise, this LED will be off. (This circuit is connected to test header JP36.) |
| LED_out[8] | 106 | LVTTL Output | Driver for LED_8. This LED will be driven high asynchronously by either reset or SWITCH[1]. After reset the LED will remain on briefly, then act as the status light for the LVDS loopback circuit 1. If the loopback circuit is functioning properly, this LED will flash. Otherwise, this LED will be off. (This circuit is connected to SMA connectors J3, J5, J6, J4, and test header JP37). |
| LVDS_in2[0] | 92 | LVDS Input | LVDS positive input. Connect to LVDS_out2[0] for loopback test. |
| LVDS_in2[1] | 91 | LVDS Input | LVDS negative input. Connect to LVDS_out2[1] for loopback test. |
| LVDS_out2[0] | 97 | LVDS Output | LVDS positive output. Connect to LVDS_in2[0] for loopback test. |
| LVDS_out2[1] | 96 | LVDS Output | LVDS negative output. Connect to LVDS_in2[1] for loopback test. |
| LVDS_in1[0] | 82 | LVDS | LVDS positive input. Connect to LVDS_out1[0] for loopback test. |
| LVDS_in1[1] | 81 | Input | LVDS negative input. Connect to LVDS_out1[1] for loopback test. |

Table 1-2. Design Pin Description (Continued)

| Name | Pin Number | Pin Type | Description |
|---|---|---|---|
| LVDS_out1[0] | 88 | LVDS Input | LVDS positive output. Connect to LVDS_in1[0] for loopback test. |
| LVDS_out1[1] | 87 | LVDS Output | LVDS negative output. Connect to LVDS_in1[1] for loopback test. |
| SWITCH[1] | 115 | LVTTL Input | Asynchronously turns on all LEDs while held. Sets LEDs[2:6] to **101010** for a short period when released. |
| SWITCH[2] | 114 | LVTTL Input | Asynchronously turns off all LEDs while held. Sets LEDs[2:6] to **010101** for a short period when released. |
| SWITCH[3] | 111 | LVTTL Input | Sets LEDs[2:6] into a counting pattern. Any other switch will override the counting mode. |
| SWITCH[4] | 110 | LVTTL Input | Reset – asynchronous reset for all logic. Also drives all LEDs high while held. |
| UartRX | 3 | LVTTL Input | Uart input signal. Note that the example design does not actually contain a Uart. The UartRX and UartTX signals are connected to each other. This will echo any incoming characters back to any terminal that is connected. |
| UartTX | 4 | LVTTL Output | Uart output signal. See UartRX for more info. |

2

# Hardware Description

This chapter describes the components of the evaluation board. See Figure 1-1 on page 7 for a drawing of the evaluation board.

## Axcelerator Evaluation Board Components

The Axcelerator evaluation board consists of the following:

- Wall mount power supply connector with switch and LED indicator
- Jumpers for I/O voltages of 1.5 V, 2.5 V or 3.3 V I/O on a per bank basis
- Multiple clock options
- Four switches that provide input to the device
- Eight LEDs driven by outputs from the device
- Jumpers that allow disconnection of all external circuitry from the FPGA
- Seven headers for easy access to device I/O signals
- Internal and external LVDS connections with loopback capability

For more information, refer to the following appendices:

- Appendix A – “Board Connections” on page 59
- Appendix B – “P160 Connections” on page 69
- Appendix C – “Board Schematic” on page 73

### Power Supply

The Axcelerator evaluation board has the following power supply *requirements* and features:

- Wall mount power supply
- 9 V, minimum 500 mA supply with 2.1 mm P5 type female connector (Note that the AX Starter Kit is supplied with a 9 V, 2 A supply as standard.)
- Digikey part number T413-P5P-ND for US
- Digikey Part Number T408-P5P-ND for Europe
- The power is controlled by an On/Off switch
- LED D9 indicates the presence of a working wall mount supply
- LED D12 indicates the presence of a working 3.3 V source

- LED D11 indicates the presence of a working 2.5 V source
- LED D13 indicates the presence of a working 1.5 V source
- JP1 - JP8, one jumper per bank, can be used to select 1.5, 2.5, or 3.3 volts for the Device I/O Voltage, on a per bank basis

# Clock Circuits

The evaluation board has multiple clock sources, which are described below.

## External Clock

An external clock can be provided for the board using the SMA connector J7, which connects the external clock to pin 192 of the device. Pin 192 is a global HCLK input pin.

## 40 MHz Oscillator

The 40 MHz oscillator is connected to J13, pin 19 (bank 5 header) on the board. J13, pin 20 connects to pin 76 of the device. Pin 76 is a global CLK input pin. If you want to use pin 76 for a different clock signal, disconnect the jumper across J13, pins 19/20.

## Multiplier Clock

An ICS512 PLL clock multiplier is provided on the board. The clock multiplier uses Y1, a Pletronics 25.0 MHz crystal, as its external clock source. The multiplier output clock frequency is determined by the jumper settings of JP25, as described in Table 2-1 below.

Table 2-1. JP25 Pins and Clock Frequency

| JP25 pins 1/2 | JP25 pins 3/4 | Clock Frequency |
|---|---|---|
| Closed | Closed | 100 MHz |
| Closed | Open | 133.33 MHz |
| Open | Closed | 62.5 MHz |
| Open | Open | 50 MHz |

The clock multiplier output is connected to J13, pin 21 (bank 5 header). J13, pin 22 connects to pin 77 of the device. Pin 77 is a global CLK input pin. If you want to use pin 77 for a different clock signal, disconnect the jumper across J13, pins 21/22.

### Clock Socket

A clock socket, Y2, is provided on the board. The output of this socket is connected to J13, pin 17 (bank 5 header). J13, pin 18 connects to pin 70 of the device. Pin 70 is a global CLK input pin. If you want to use pin 70 for a different clock signal, disconnect the jumper across J13, pins 17/18.

### Manual Clock

The manual clock button (SW6) lights D14, the pulse-generated LED, when activated, and generates a pulse. The manual clock is connected to J13, pin 15 (bank 5 header). J13, pin 16 connects to pin 71 of the device. Pin 71 is a global CLK input pin. If you want to use pin 71 for a different clock signal, disconnect the jumper across J13, pins 15/16.

## LVDS Connections

Both internal and external LVDS is available with loopback capability. Pads for SMA connectors are provided for external LVDS connections. Jumper JP37 allows for loopback of external LVDS connections, and Jumper JP36 allows for loopback of internal LVDS connections. The LVDS connections are detailed in Table 2-2.

Table 2-2. LVDS Connections

| Source | Signal | Jumper/Pin | Device Connection |
|---|---|---|---|
| SMA J3 | Receive P | JP38 – Pin 3 | 82 |
| SMA J5 | Receive N | JP38 – Pin 4 | 81 |
| SMA J6 | Transmit P | JP38 – Pin 1 | 88 |
| SMA J4 | Transmit N | JP38 – Pin 2 | 87 |
| Device | Receive P | JP36 – Pin 1 | 92 |
| Device | Receive N | JP36 – Pin 3 | 91 |
| Device | Transmit P | JP36 – Pin 2 | 97 |
| Device | Transmit N | JP36 – Pin 4 | 96 |

# LED Device Connections

Eight LEDs are connected to the device via jumpers. If the jumpers are in place, the device I/O can drive any of the LEDs. The LEDs change based on the following output:

- A **1** on the output of the device lights the LED.
- A **0** on the output of the device switches off the LED.
- An unprogrammed or tristated output may show a faintly lit LED.

Table 2-3 lists the LED/device connection.

Note: If you want to use the device I/O for other purposes, remove the jumpers.

Table 2-3. LED Device Connection

| LED | Device Connection |
|---|---|
| D1 | 109 |
| D2 | 108 |
| D3 | 107 |
| D4 | 106 |
| D5 | 103 |
| D6 | 102 |
| D7 | 101 |
| D8 | 100 |

# Switch Device Connections

Four switches are connected to the device via jumpers. If the jumpers are in place, the device I/O can be driven by the switches.

- Pressing the switch drives a **1** into the device. The **1** continues to drive while you hold the switch.
- Releasing the switch drives a **0** into the device.

Table 2-4 on page 17 lists the switch/device connections.

Note: If you want to use the device I/O for other purposes, remove the jumpers.

Table 2-4. Switch/Device Connection

| Switch | Device Connection |
|---|---|
| SW1 | 115 |
| SW2 | 114 |
| SW3 | 111 |
| SW4 | 110 |

# Headers

Seven headers are included on the board, allowing easy access to the AX250 I/Os. Table 2-5 lists the header connections.

Table 2-5. Header Connections

| Pin | BANK 0/1 | BANK2 | BANK3 | BANK4 | BANK5 | BANK6 | BANK7 |
|---|---|---|---|---|---|---|---|
| 1 | $V_{CC1B0}$ | 3.3 V | 3.3 V | 3.3 V | 3.3 V | 3.3 V | 3.3 V |
| 2 | 3.3 V | 3.3 V | 3.3 V | 3.3 V | 3.3 V | 3.3 V | 3.3 V |
| 3 | $V_{CCIB1}$ | $V_{CCIB2}$ | $V_{CCIB3}$ | $V_{CCIB4}$ | $V_{CCIB5}$ | $V_{CCIB6}$ | $V_{CCIB7}$ |
| 4 | 1.5 V | 1.5 V | 1.5 V | 1.5 V | 1.5 V | 1.5 V | 1.5 V |
| 5 | P199 | P154 | P129 | LED_1 | P54 | P27 | UART_SIN |
| 6 | P198 | P153 | P128 | LED1 | P55 | P28 | UART_SOUT |
| 7 | P197 | P152 | P127 | LED_2 | P56 | P29 | P5 |
| 8 | P191 | P151 | P126 | LED2 | P57 | P30 | P6 |
| 9 | P186 | P148 | P123 | LED_3 | P60 | P33 | P7 |
| 10 | P185 | P147 | P122 | LED3 | P61 | P34 | P10 |
| 11 | P181 | P146 | P121 | LED_4 | P62 | P35 | P11 |

Table 2-5. Header Connections (Continued)

| Pin | BANK 0/1 | BANK2 | BANK3 | BANK4 | BANK5 | BANK6 | BANK7 |
|---|---|---|---|---|---|---|---|
| 12 | P180 | P145 | P120 | LED4 | P66 | P36 | P12 |
| 13 | P175 | P144 | P117 | LED_5 | P67 | P37 | P13 |
| 14 | P174 | P141 | P116 | LED5 | NC | P40 | P16 |
| 15 | P171 | P140 | USER_1 | LED_6 | MANUAL_CLOCK | P41 | P17 |
| 16 | P170 | P139 | USER1 | LED6 | MANUAL_CLK_IN | P42 | P18 |
| 17 | P166 | P138 | USER_2 | LED_7 | CLK_SOCKET | P43 | P19 |
| 18 | P165 | P137 | USER2 | LED7 | CLK_SOCKET_I | P44 | P22 |
| 19 | P162 | P136 | USER_3 | LED_8 | OSC_CLK | P47 | P23 |
| 20 | P161 | P135 | USER3 | LED8 | OSC_CLK_IN | P48 | P24 |
| 21 | P160 | P134 | USER_4 | NC | MULT_CLK | P49 | P25 |
| 22 | P159 | P133 | USER4 | NC | MULT_CLK_IN | P50 | NC |
| 23 | GND | GND | GND | GND | GND | GND | GND |
| 24 | GND | GND | GND | GND | GND | GND | GND |

# 3 Tutorial

This tutorial illustrates a basic Verilog design for an Axcelerator evaluation board. The design targets the Actel Axcelerator family. This tutorial contains the following nine high-level steps:


- "Step 1 – Create a New Project"
- "Step 2 – Add ACTgen Components"
- "Step 3 – Perform a Pre-Synthesis Simulation"
- "Step 4 – Synthesize the Design in Synplify®"
- "Step 5 – Perform a Post-Synthesis Simulation"
- "Step 6 – Implement the Design Using Designer"
- "Step 7 – Perform a Timing Simulation with Back-Annotated Timing"
- "Step 8 – Generate the Programming File"
- "Step 9 – Program the Device"


In this tutorial, you will create a self-checking serial loopback system operating at 40 MHz. The AX device will transmit and receive data at 80 Megabits per second using its DDR functionality. A 10-bit test pattern (**0101110011**) will transmit and receive using the AX device's LVDS I/Os.

## Step 1 – Create a New Project

In this step you will use the Libero IDE HDL editor to enter the Actel Verilog design.

### *To create the Verilog project:*

1. **Start Libero IDE by double-clicking the Actel Libero IDE icon on your desktop.**

2. **From the File menu, choose *New Project*.** The New Project Wizard dialog box is displayed, as shown in Figure 3-1.

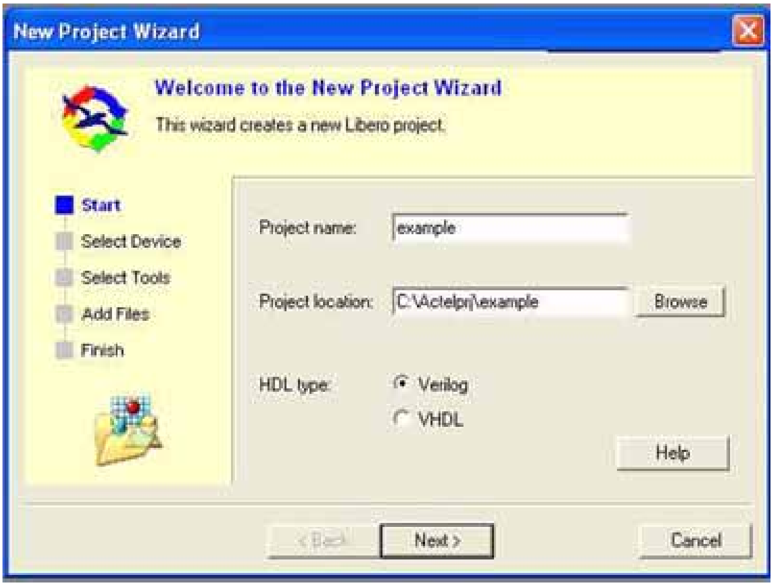


Figure 3-1. Project Wizard Welcome

3. **Type *example* in the Project name field.**
4. **In the Project location field, click *Browse* to navigate to *C:\Actelprj*.**
5. **Type *example* in the Project location field after C:\Actelprj.**
6. **Select the *Verilog* radio button in the HDL type field.**
7. **Click *Next*.**

8. **Select the *Axcelerator* family, *AX250* die, and *208PQFP* package.** See Figure 3-2.


Figure 3-2. Family, Die, and Package

9. **Click *Finish*.**

   The software creates the project **example** and opens it in the Libero IDE.

10. **From the File menu, choose *New*.**

The **New** dialog box opens, as shown in Figure 3-3.

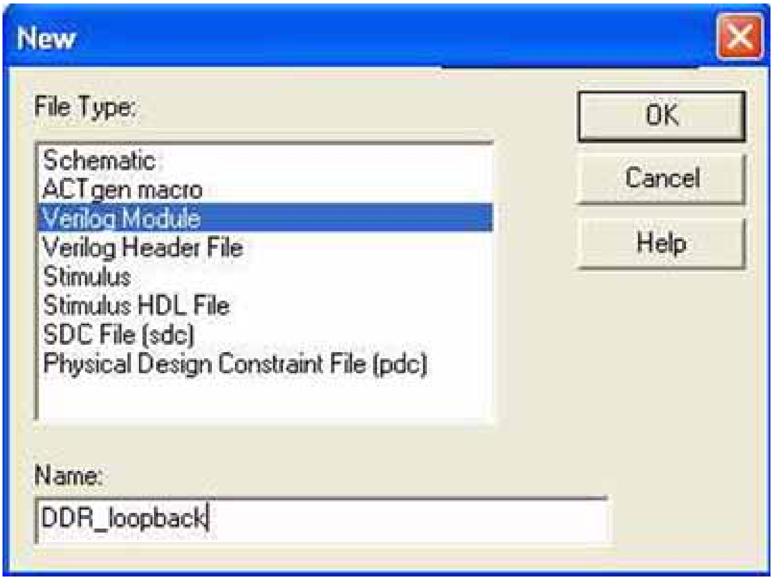


Figure 3-3. New Dialog Box

11. **Select *Verilog Module* in the File Type field.**
12. **Type *DDR_loopback* in the Name field.**
13. **Click *OK*.**

    The **HDL Editor** opens.

14. **Enter the DDR_loopback file, described in Appendix A** ("Board Connections" on page 59). If the document is open in an electronic form, cut and paste it from this document.
15. **From the File menu, choose *Save*.**

The design file **DDR_loopback** appears on the Design Hierarchy tabbed page in Libero IDE. The file name **DDR_loopback** is listed under HDL files on the File Manager tabbed page in the Libero IDE, as shown in Figure 3-4 and Figure 3-5.

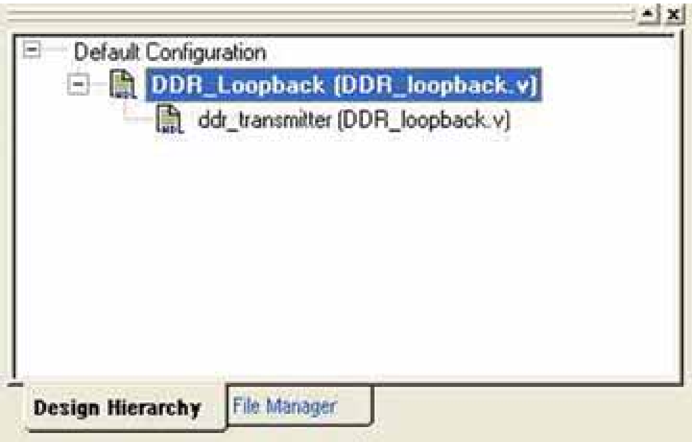


Figure 3-4. Design Hierarchy Tab

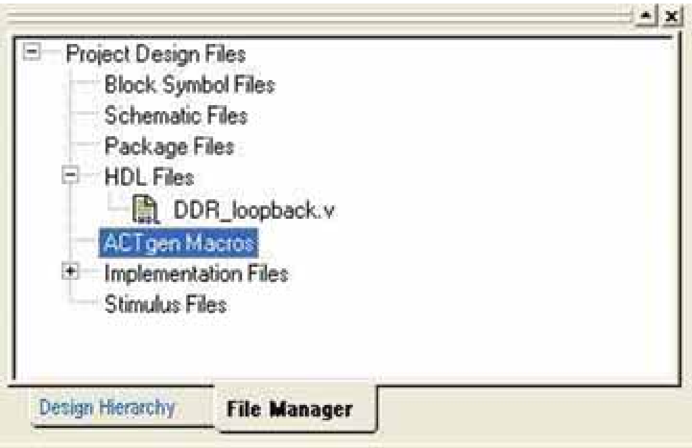


Figure 3-5. File Manager Tab

16. Click the *File Manager* tab.
17. Right-click on *DDR_loopback.v* and select *check HDL*.

This checks the syntax of the **DDR_loopback.v** file. Before moving to the next section, check the code for errors and make modifications if necessary.

18. **Repeat steps 8-13 for *Switch_LED_driver.v* and *top.v* files.**

# Step 2 – Add ACTgen Components

In this step, you learn how to add ACTgen components.

### *To add ACTgen components:*

1. **Click on the *ACTgen* icon in the Design Flow window.**

   The **New** dialog box opens, as shown in Figure 3-6. The ACTgen macro file type is selected.

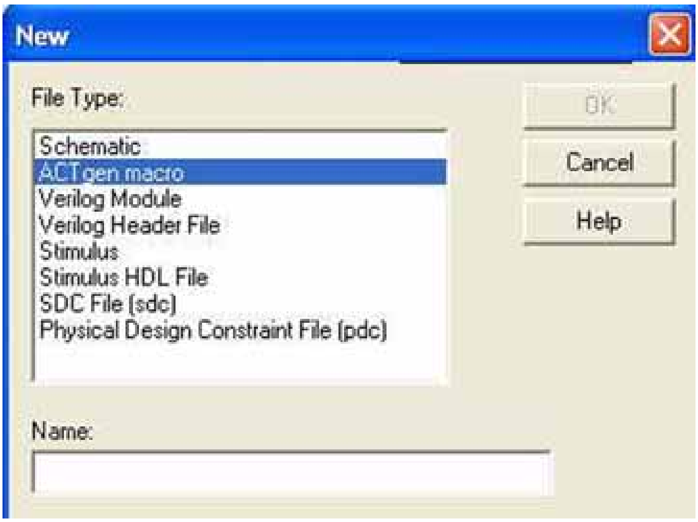


Figure 3-6. ACTgen Macro New Dialog Box

2. **Enter *pll_ax3* in the Name field and click *OK*.**

   The ACTgen GUI opens.

3. **Select *PLL* macros in the ACTgen GUI, as shown in Figure 3-7.**

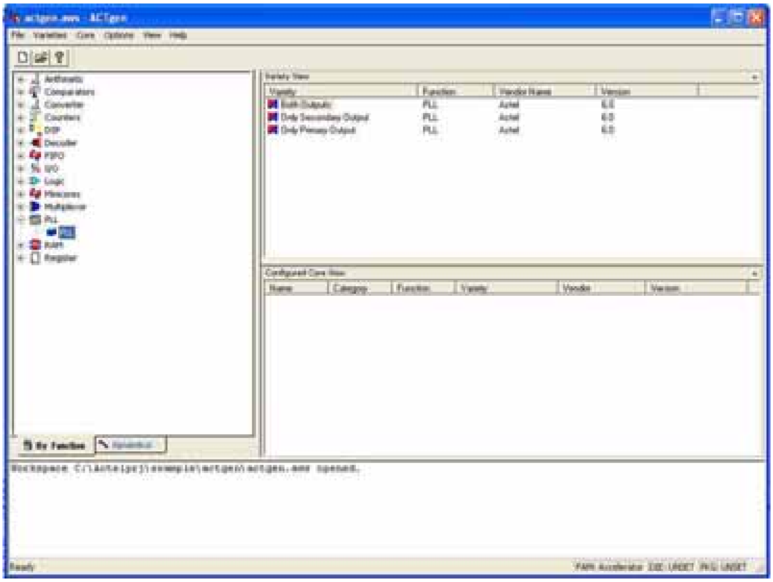

Figure 3-7. PLL Macros

4. **Select the *Both Outputs* option in the Variety View window.**

   The ACTgen Macro Builder – PLL dialog box opens. Enter the following values, as shown in Figure 3-8 on page 26:

   - REF Clock: **40 MHz**
   - CLK1: **40 MHz**
   - CLK2: **80 MHz**

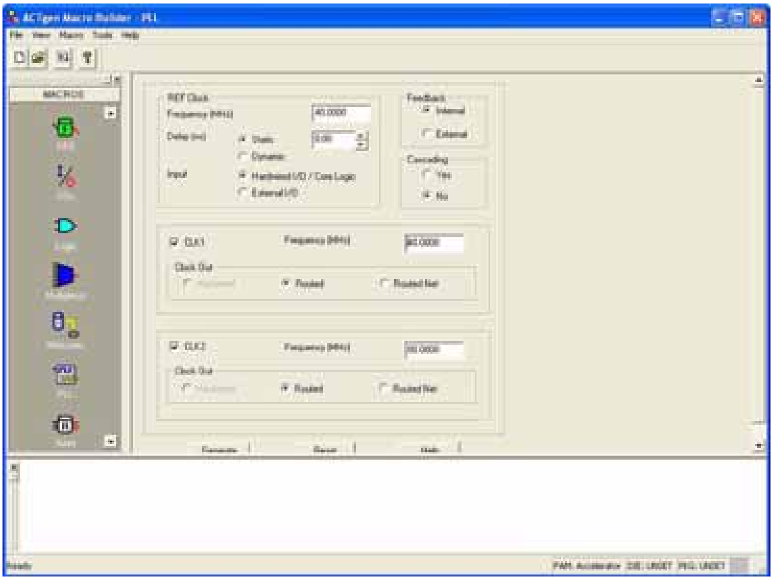


Figure 3-8. ACTgen Macro Builder - PLL

5. **Click the *Save* button.**

   The **Save As** dialog box opens.

6. **Verify that all the information is correct and click *Save*.**

The **pll_ax3** file appears in the Design Hierarchy window, as shown in Figure 3-9.


Figure 3-9. Design Hierarchy, pll_ax3

# Step 3 – Perform a Pre-Synthesis Simulation

The next step is to simulate the RTL description of the design. First, you will use SynaptiCAD WaveFormer Lite™ AE to create a stimulus for the design, and then you will generate a test bench for the design.

## Creating a Stimulus Using WaveFormer Lite

WaveFormer Lite generates VHDL test benches from drawn waveforms. Listed below are the three basic steps for creating test benches using WaveFormer Lite and the Actel Libero IDE software:

- Importing Signal Information
- Drawing and Copying Waveforms
- Exporting the Test Bench

### Importing Signal Information

Follow the steps below to launch WaveFormer Lite and import the signal information into it.

#### *To launch WaveFormer Lite and import signal information:*

1. **Before you start WaveFormer Lite, make sure that the AX_starterkit block is at the top level (it will be in a bold font style on the Design Hierarchy tabbed page).** If the **AX_starterkit** is not at the top level, right-click the file and select **set as root**.

2. **Click the *WaveFormer Lite* icon in the Libero IDE, or right-click the *AX_starterkit* file on the Design Hierarchy tabbed page and select *Create Stimulus*.**

   WaveFormer Lite launches and the port signals appear in the diagram window, as shown in Figure 3-10.

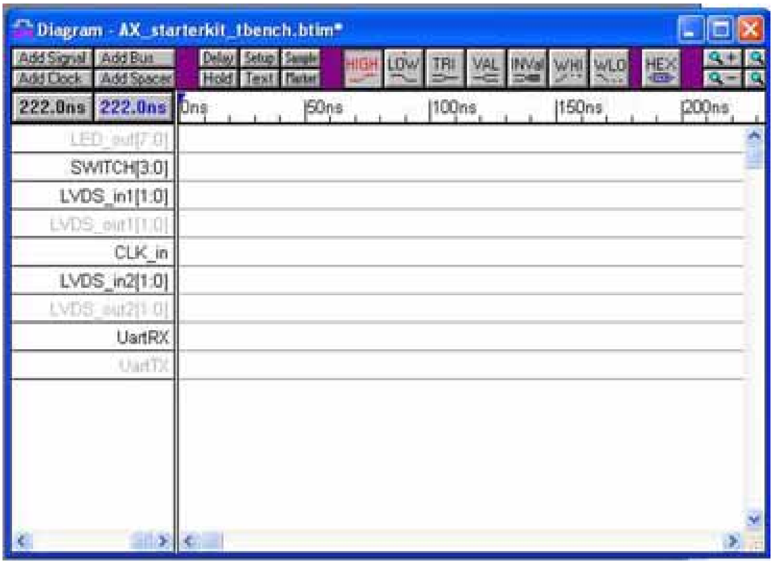


Figure 3-10. WaveFormer Lite with Port Signals

As shown in Figure 3-10, the AX_starterkit design contains the following signals:

- LED_out output signals
- SWITCH input signals
- LVDS_in1 input signals
- LVDS_out1 output signals
- CLK_in input signals
- LVDS_in2 input signals
- LVDS_out2 output signals
- UartRX input signals
- UartTX output signals

## Drawing Waveforms

The state buttons are the buttons labeled with pictures of waveforms: HIGH, LOW, TRIstate, VALid, INValid, WHI (weak high), and WLO (weak low), as shown in Figure 3-11.


Figure 3-11. State Buttons

When a state button is activated, it is pushed in and colored red. The active state is the type of waveform that is drawn next. Click a state button to activate it.

The state buttons automatically toggle between the two most recently activated states. The state with the small red '1' above the name will be the toggle state. The initially activated state is HIGH and the initial toggle state is LOW in Figure 3-11.

Signal edges are automatically aligned to the closest edge grid when you draw signals using the mouse. Control the edge grid with the **Options** > **Grid Setting** menu items.

### *To draw a waveform:*

1. **Select the *High* state and place the mouse cursor inside the diagram window at the same vertical row as the signal name.**
2. **Click with the left mouse button.**

   This draws a waveform from the end of the signal to the mouse cursor. The red state button on the button bar determines the type of waveform drawn. The cursor shape also mirrors the red state button.
3. **Move the mouse to the right and click again to draw another segment.**

## Copying Waveforms

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

### *To copy and paste waveform sections:*

1. **Select the names of the required signals. If you do not select a signal, the *Block Copy* command selects all the signals in the diagram.**

2. From the Edit menu, select *Block Copy Waveforms*. This opens the **Block Copy Waveforms** dialog box with the selected signals displayed in the **Change Waveform Destination** list box, as shown in Figure 3-12.

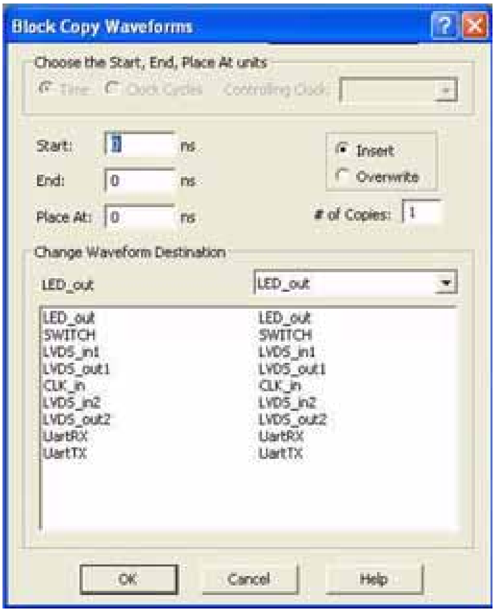


Figure 3-12. Block Copy Waveform Dialog Box

3. In the *Block Copy Waveforms* dialog box, enter the values that define copy and paste (*Start*, *End*, and *Place At*).
4. Select either *Time* or *Clock Cycles* for the basic units of the dialog.
5. When entering values, remember the following:
   - When copying only signals (no clocks), time is the default base unit of the dialog.
   - When copying part of a clock, it is best to choose **clock cycles** as the base unit; choose the copied clock as the reference clock.
   - If time is selected when copying clocks, the **End** time minus **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 at which the block will be pasted.

- The **Insert** and **Overwrite** radio buttons determine whether the paste block is inserted into the existing waveforms or overwrites 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:

- Select a line in the list box.

  This places the destination signal in the drop-down list box on top of the list box.
- Select another signal from the drop-down list box.

  Each destination signal can be used only once per copy.

6. **Click OK to complete the copy and paste operation.**

## Creating Clocks

The signal CLK_in is a clock signal. Instead of drawing the entire clock waveform, Waveformer Lite allows you to create the clock signals with the GUI.

### *To create the CLK_in with the GUI:*

1. **Left-click the *CLK_in* signal name once to select it.**
2. **Right-click the *CLK_in* signal name.**

   A shortcut menu appears.

3. Click *Signal(s) <-> Clock(s)* (see Figure 3-13). A clock signal appears.

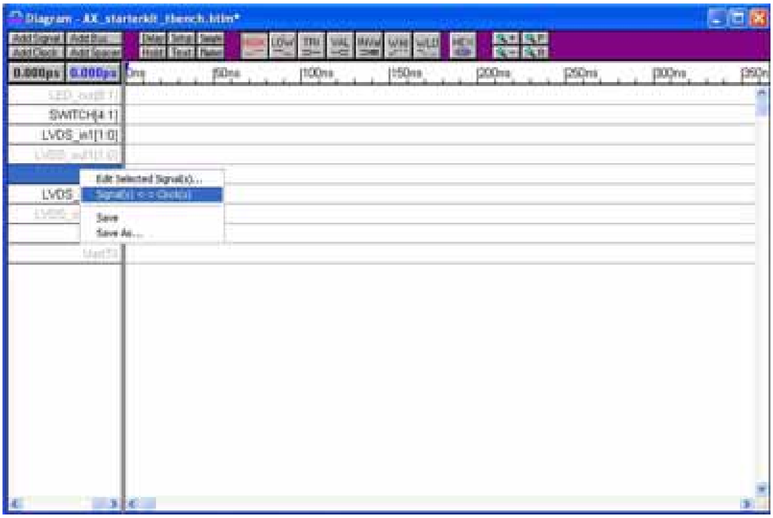


Figure 3-13. Edit Selected Signals

4. Double-click (left mouse button) the *CLK_in* signal.

The Signal Properties window appears. See Figure 3-14.

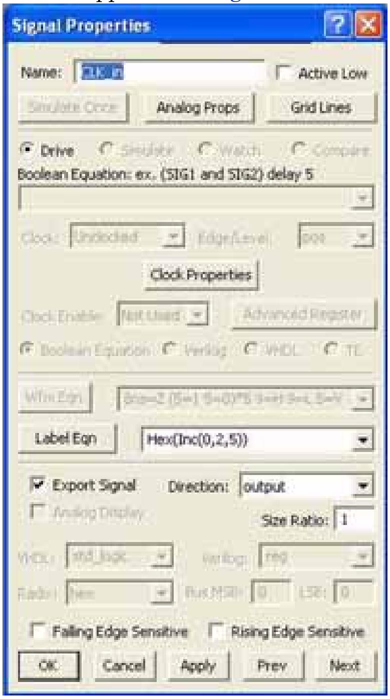


Figure 3-14. Signal Properties

5. **Click the *Clock Properties* button.**

The Edit Clock Parameters window appears. See Figure 3-15.

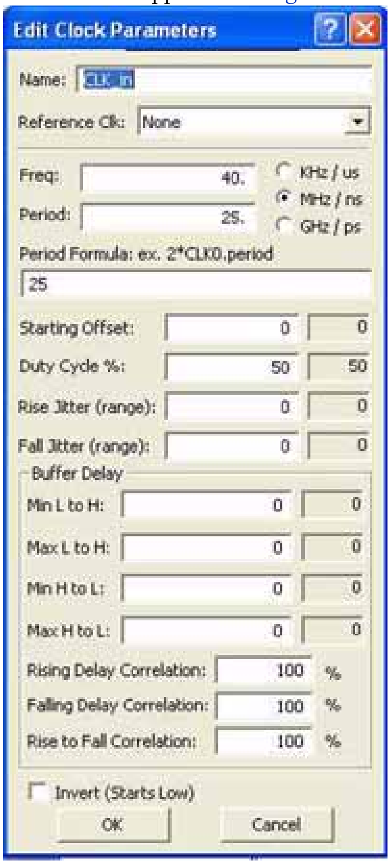


Figure 3-15. Edit Clock Parameters Dialog Box

6. Enter *40 MHz* as the frequency.
7. Click *OK*.
8. Enter the remaining parameters.

9. Following the instructions on the previous pages, create the input signals for input SWITCH[4:1], as described in Table 3-1 to create the waveform shown in Figure 3-16.

Table 3-1. SWITCH[4:1] Input Signals

| | 0 - 1 µs | 1 - 1.5 µs | 1.5 µs - 11 µs |
|---|---|---|---|
| SWITCH[4:1] | Low | 8 HEX | Low |

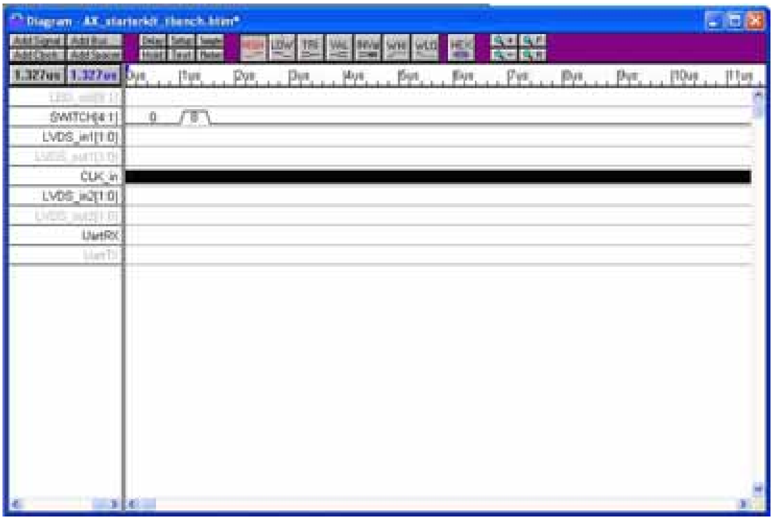

Figure 3-16. Waveform

## Saving Waveforms and Exporting Test Bench

Follow the instructions below to save a waveform.

### *To save a waveform:*

1. After successfully creating the waveform, select *Save As* from the File menu.
2. In the *Save As* dialog box, enter *AX_startkit_tbench.btim* as the file name and click *Save*.
3. From the Export menu, choose *Export Timing Diagram As*.

4. **Select** ***VHDL w/Top Level Test Bench (*.vhd)*** **in** ***Save as type*** **and enter** ***AX_starterkit_tbench.vhd*** **for the file name, as shown in Figure 3-17.**

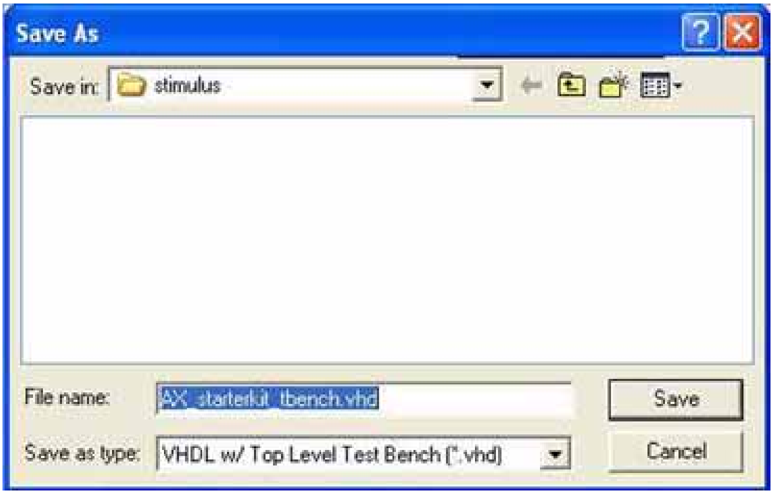


Figure 3-17. Save As Dialog Box

The **WaveFormer Lite Report** window displays the VHDL test bench with a component declaration and instantiation inside.

5. **Exit WaveFormer Lite (*File* > *Exit*).**

The Libero IDE File Manager tabbed page displays the stimulus files and the design is ready to simulate under Model*Sim*®.

## Creating a Test Bench Using HDL Editor

You can create a test bench using the HDL editor.

### *To create a stimulus file with the HDL Editor:*

1. **From the File menu, choose** ***New.***

This opens the **New File** dialog box.

2. **Select** ***Stimulus HDL file*** **from the File Type list, enter** ***AX_startkit_tbench*** **for the name, and click** ***OK.***

The file opens in the HDL Editor.

3. **Create the VHDL test bench and save it.**

Note: Use **test bench** as the entity name and follow Libero naming conventions. See the *Libero User's Guide* for more information.

## Pre-Synthesis Simulation

After the test bench is generated, use Model*Sim* to perform a pre-synthesis simulation.

### *To perform a pre-synthesis simulation:*

1. **Select a stimulus file. Right-click** ***AX_starterkit*** **on the Libero IDE Design Hierarchy tabbed page and select** ***Select Stimulus*****, as shown in Figure 3-18.**

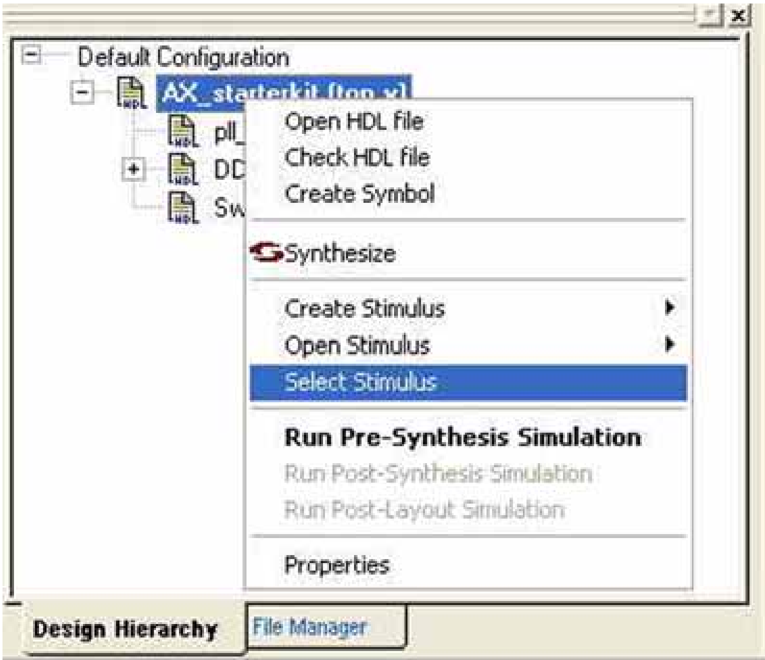


Figure 3-18. Select Stimulus

The **Select Stimulus** dialog box appears as shown in Figure 3-19.

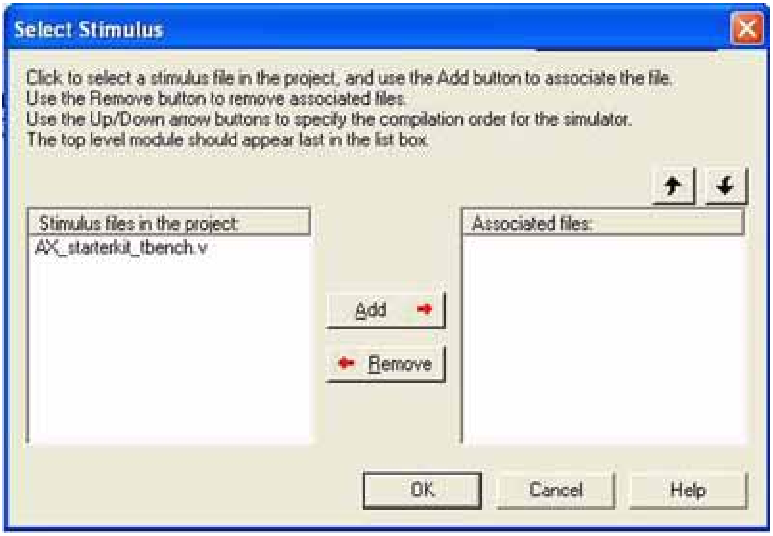


Figure 3-19. Select Stimulus Dialog Box

2. **Select** ***AX_starterkit_tbench.v*** **in the** ***Stimulus files in the project*** **window, and click** ***Add*** **to add the file to the** ***Associated files*** **list.** See Figure 3-20.


Figure 3-20. Selecting AX_starterkit_tbench.v

3. **Click** ***OK.***

A check mark appears next to WaveFormer Lite in the Process windows, as shown in Figure 3-21.


Figure 3-21. Current Project State Dialog Box

4. Double-click the Model*Sim Simulation* icon in the Libero IDE Process window, or right-click *AX_starterkit* on the Libero IDE Design Hierarchy tabbed page and select *Run Pre-Synthesis Simulation*, as shown in Figure 3-22.

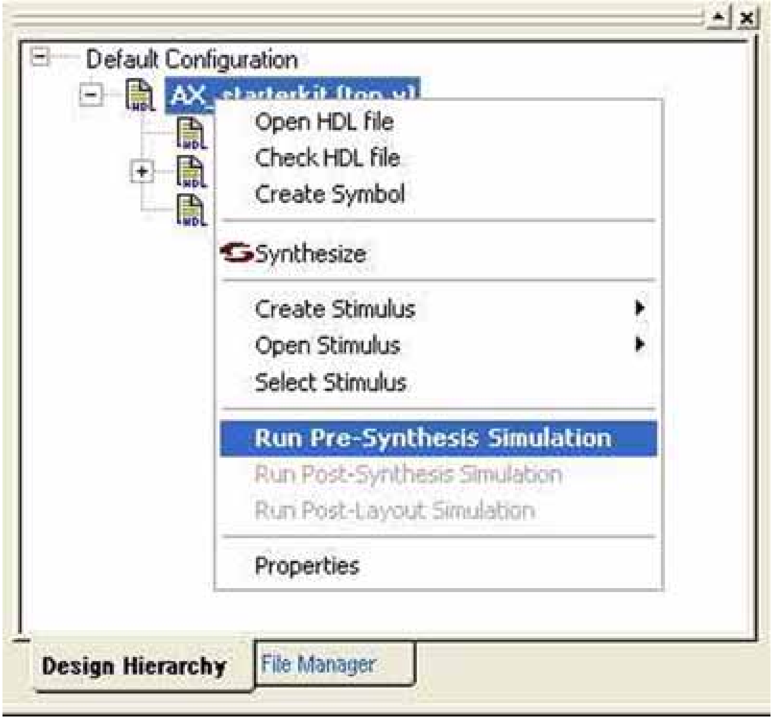


Figure 3-22. Run Pre-Synthesis Simulation

The Model*Sim* simulator opens and compiles the source files, as shown in Figure 3-23.

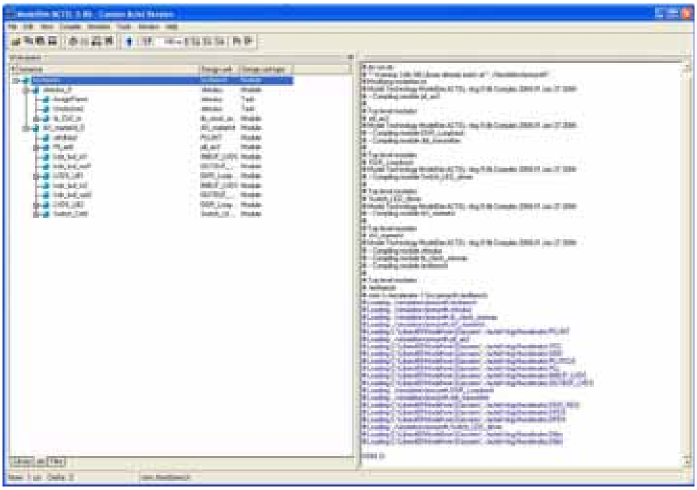

Figure 3-23. Opening and Compiling Source Files

After compilation, the simulator simulates for the default time period of 1000 ns. A wave window opens to display the simulation results, as shown in Figure 3-24.

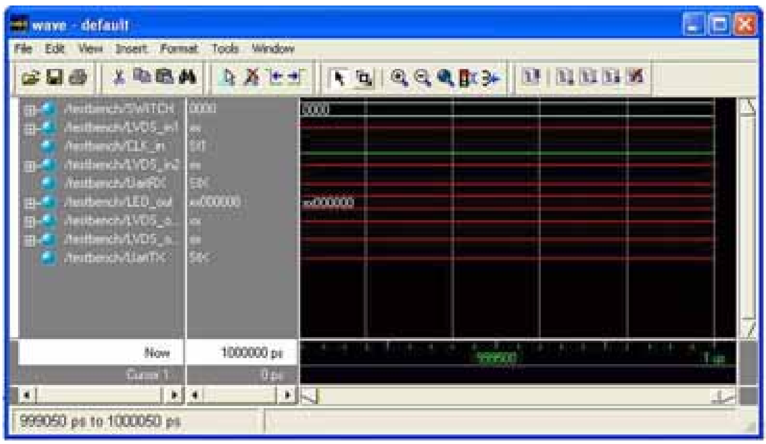


Figure 3-24. Wave Default

5. **For this simulation, there are no meaningful results in the first microsecond. Run the simulation for another nine microseconds by typing *run 9uS* at the Model*Sim* command prompt and pressing *Enter* (Figure 3-25).**


Figure 3-25. Running Simulation

Once the run command is complete, the wave window updates with the extended simulation results. Scroll through the wave window to verify that the design functions properly. See Figure 3-26. Use the **Zoom Full** button to see the entire simulation result. See Figure 3-27.

Figure 3-26. Wave Window


Figure 3-27. Zoom Full Button

6. **In the Model*Sim* window, select *Quit* from the File menu.**

# Step 4 – Synthesize the Design in Synplify®

In this step you generate an EDIF netlist by synthesizing the design in Synplify. For HDL designs, Libero IDE launches and loads the Synplicity® Synplify synthesizer with the appropriate design files.

### *To create an EDIF netlist for the design using Synplify:*

1. **In the Libero IDE, double-click the *Synplify Synthesis* icon in the Libero IDE process window or right-click the *AX_starterkit* file on the Design Hierarchy tabbed page and select *Synthesize*.**

This launches the Synplify Synthesis tool with the appropriate design files, as shown in Figure 3-28.

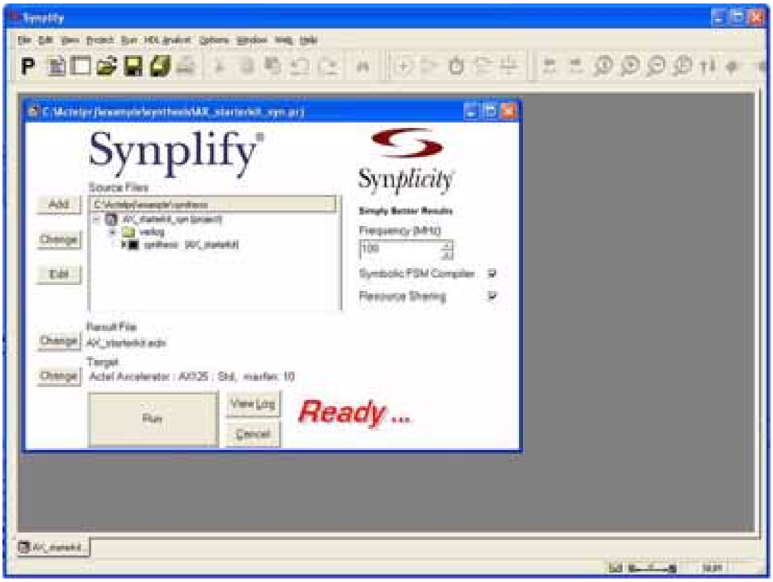


Figure 3-28. Synplify Synthesis Tool

**2. From the Project menu, choose *Implementation Options*.**

The **Options for Implementation** dialog box opens, as shown in Figure 3-29.

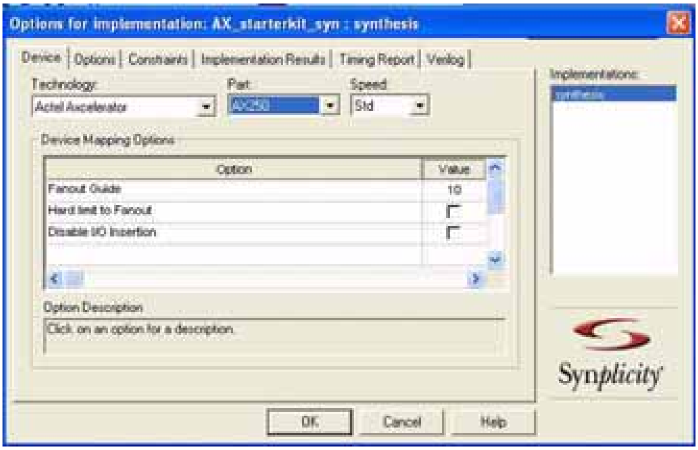


Figure 3-29. Options for Implementation Dialog Box

3. **Set the following options in the dialog box:**
   - Technology: **Actel Axcelerator** (set by Libero IDE)
   - Part: **AX250**
   - Speed: **Std** (default)
   - Fanout Guide: **10** (default)
   - Hard Limit to Fanout: **Off** (default)
   - Disable I/O Insertion: **Off** (default)
4. **Accept the default values for each of the other tabbed pages in the Options for Implementation dialog box and click *OK*.**
5. **From the Synplify main window, click *Run*.**

   Synplify compiles and synthesizes the design into a netlist called **AX_starterkit.edn**. The resulting **AX_starterkit.edn** file is then automatically translated by Libero into a Verilog netlist called **AX_starterkit.v**. The resultant EDIF and Verilog files are displayed under the **Implementation** files on the **File Manager** tabbed page of Libero IDE.

   Note: If any errors appear after clicking the **Run** button, edit the files using the **Synplify** editor. To edit the file, double-click the file name in the **Synplify** window. Any changes you make here are saved to the original design file in Libero IDE.

6. **From the File menu, choose *Exit* to close Synplify. Click *Yes* to save any settings made to the *AX_starterkit.prj* in Synplify.**

# Step 5 – Perform a Post-Synthesis Simulation

The next step is to simulate the Verilog netlist of the AX_starterkit design using the test bench created in the "Creating a Stimulus Using WaveFormer Lite" section.

### *To simulate the Verilog netlist of the AX_starterkit design:*

1. **Select the Model*Sim Simulation* icon in the Libero IDE Process window, or right-click the *AX_starterkit* file in the Design Hierarchy tabbed page and select *Run Post-Synthesis Simulation*.**

   This launches the **Model*Sim* Simulator,** which compiles the source file and test bench. Once the compilation completes, the simulator runs for 1000 ns and a wave window opens to display the simulation results.

2. **Using the *run 9us* command, extend the simulation to 10 μs.**
3. **Scroll through the wave window to verify that the andgate works correctly.**

# Step 6 – Implement the Design Using Designer

After creating and testing the design, the next phase is implementing the design using the Actel Designer software.

### *To implement the design using the Actel Designer software:*

1. **Double-click the Designer *Place-and-Route* icon in the Libero IDE Process window, or right-click *AX_starterkit* on the Design Hierarchy tabbed page of the Designer Explorer window, and select *Run Designer*.**

The Actel Designer application opens and the design files are read in, as shown in Figure 3-30.

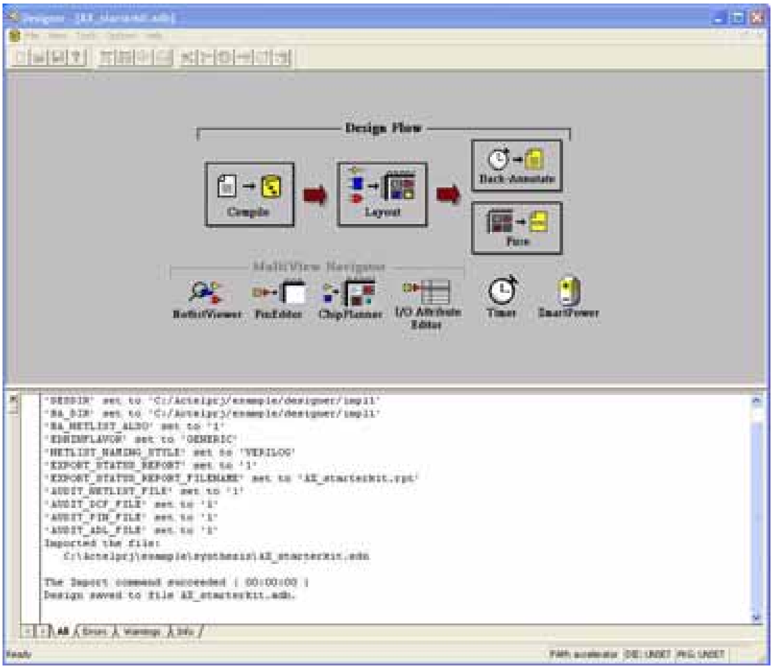


Figure 3-30. Designer Application

2. **Compile the design.**

The **Device Selection Wizard** window appears, as shown in Figure 3-31.

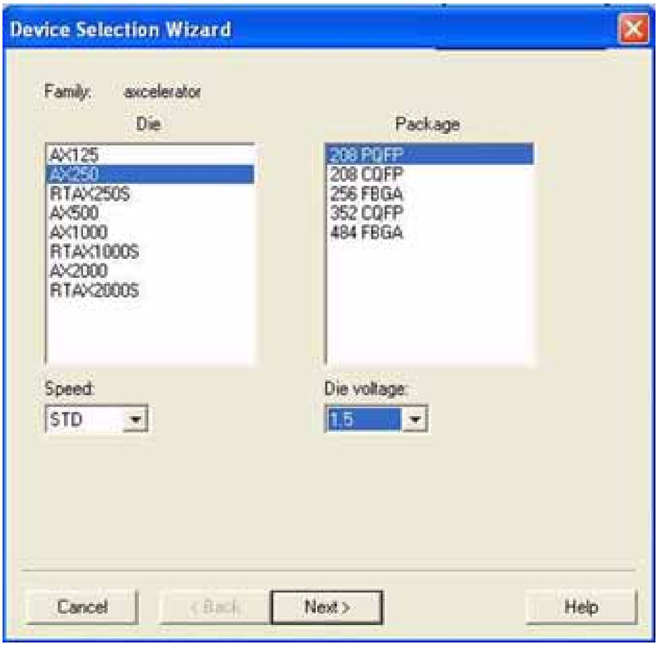


Figure 3-31. Device Selection Wizard

3. **Select *AX250* in the Die field, *208 PQFP* in the Package field, and change the Speed to *STD*. Accept the default Die voltage and click *Next*.**
4. **Accept the default settings in the remaining fields and click *Finish*.**
5. **Double-click the *Compile* icon.**

   Designer compiles the design and shows the utilization of the selected device. The **Compile** icon in Designer turns green to indicate that the compile has successfully completed.

## Assigning Pins

Once the design compiles successfully, use the PinEditor tool to perform drag-and-drop placement of pins and fix pin locations for subsequent place-and-route runs.

1. **Click the PinEditor user tool.**

This opens the **PinEditor** window inside the **MultiView Navigator** window, as shown in Figure 3-32.

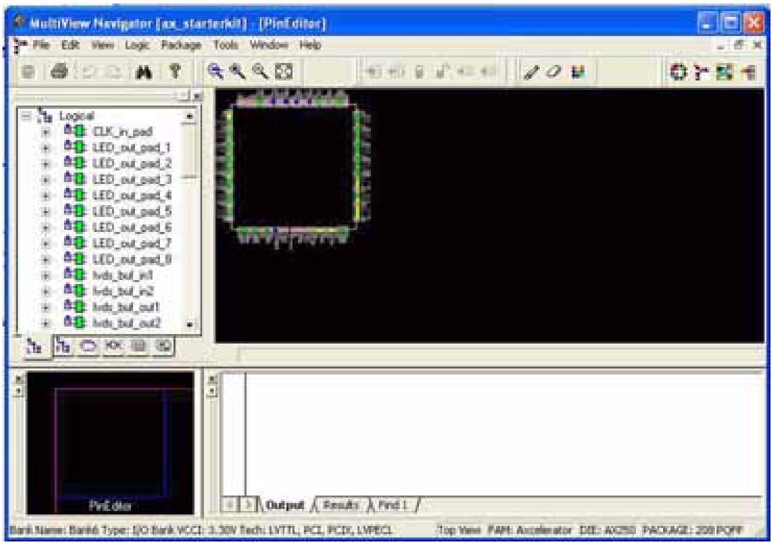


Figure 3-32. Pin Editor Window

2. **Select the *port* tab in the MultiView Navigator window, as shown in Figure 3-33.**

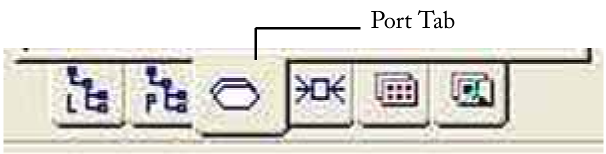


Figure 3-33. Port Tab

3. Expand the ports by selecting the "+" sign on the Ports tabbed page, as shown in Figure 3-34.

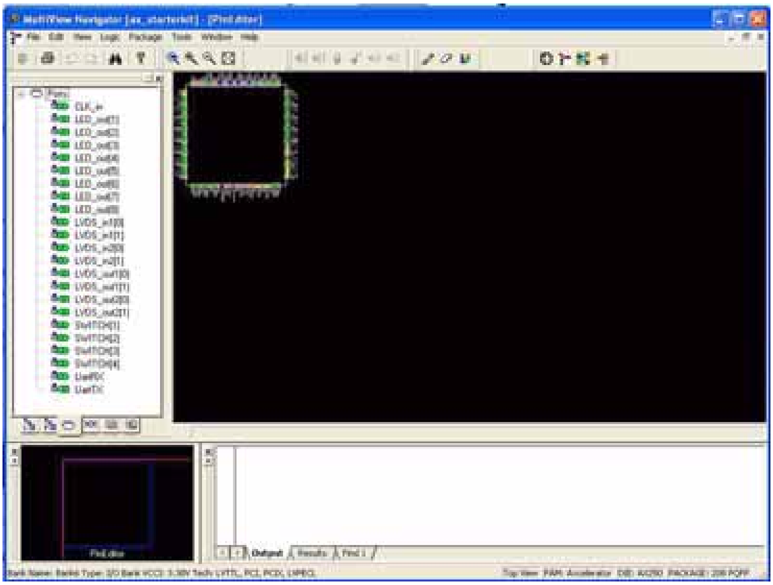

Figure 3-34. Expanding Ports

4. Drag the ports to the pin location.

5. **Assign pins according to Table 3-2.**

Table 3-2. Port/Pin Table

| Port | Pin | Port | Pin |
|---|---|---|---|
| CLK_in | 76 | LVDS_in2[0] | 92 |
| LED_out[1] | 123 | LVDS_in2[1] | 91 |
| LED_out[2] | 122 | LVDS_out[0] | 88 |
| LED_out[3] | 121 | LVDS_out[1] | 87 |
| LED_out[4] | 120 | LVDS_out2[0] | 97 |
| LED_out[5] | 109 | LVDS_out2[1] | 96 |
| LED_out[6] | 108 | SWITCH[1] | 115 |
| LED_out[7] | 107 | SWITCH[2] | 114 |
| LED_out[8] | 106 | SWITCH[3] | 111 |
| LVDS_in1[0] | 82 | SWITCH[4] | 110 |
| LVDS_in1[1] | 81 | UartRX | 3 |
| | | UartTX | 4 |

Note: For the LVDS I/Os, selecting either the p-channel or n-channel I/O will result in selecting both I/Os.

6. **Once a pin number is assigned to all of the signals, select *Commit* from the File menu.**

## I/O Bank Configuration

There are LVDS I/Os in this example design. The I/O bank to which they are assigned must be configured for LVDS.

### To configure the I/O Banks:

1. Select *I/O Bank Settings* from the PinEditor Tools menu, as shown in Figure 3-35.


Figure 3-35. I/O Bank Settings

2. The I/O Bank Settings dialog box appears, as shown in Figure 3-36.

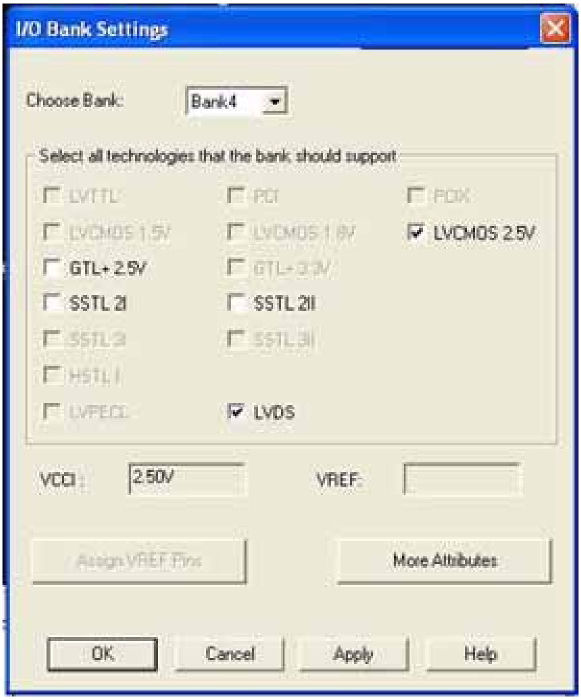


Figure 3-36. I/O Bank Settings Dialog Box

3. **Choose Bank4, and select LVDS.**

   Note: Any I/O standards that are not compatible with LVDS are grayed out.

4. **Accept the default settings for the other banks, and click OK to return to the main PinEditor GUI.**

5. **From the File menu, choose *Commit* to save the changes and close the PinEditor window.**

## Optional Designer Tools

After successfully compiling the design, you can use the Designer Tools to view a pre-layout static timing analysis with Timer, set time constraints in Timer, and use ChipEdit to assign modules. You can access each of these functions by clicking the appropriate icon.

Note: For more information on these functions, refer to the *Designer User's Guide* and online help. For this tutorial, no changes are made to the design.

## Lay Out the Design

Follow the instructions below to lay out the design.

### *To lay out the design:*

1. **From Designer, click the *Layout* icon.**

This opens the **Layout Options** dialog box, shown in Figure 3-37.

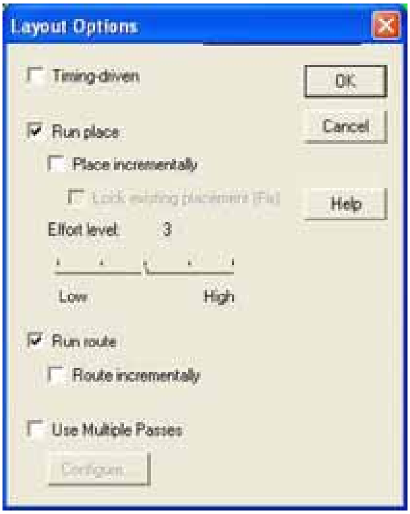


Figure 3-37. Layout Options

2. **Click *OK* to accept the default layout options.**

   This runs the place-and-route process on the design. The Layout icon in designer turns green to indicate that the layout has been successfully completed.

## Back-Annotate the Design

Use the following instructions to back-annotate the design.

### *To back-annotate the design:*

1. **From Designer, click the *Back-Annotate* icon.**

This opens the Back-Annotate dialog box, shown in Figure 3-38.

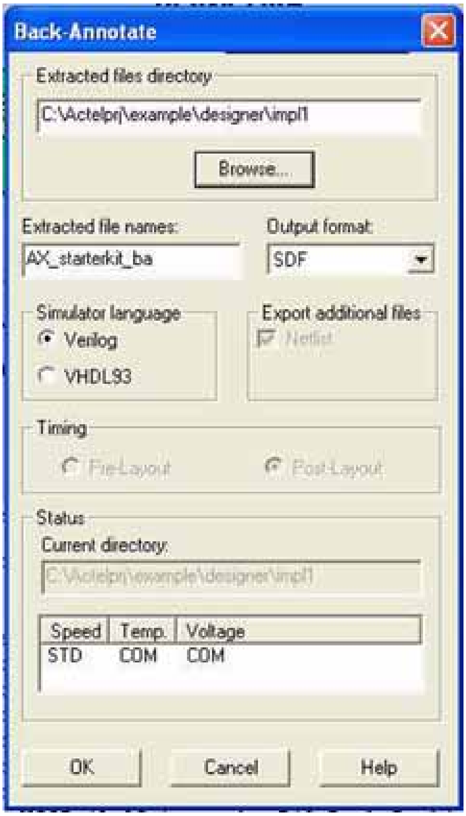


Figure 3-38. Back-Annotate Dialog Box

2. **Accept the default settings and click *OK*.**

   The **Back Annotate** icon turns green.

3. **From the File menu, choose *Exit*.**

4. **Click *Yes* to save the design before closing Designer.**

   Designer saves all the design information in an ADB file.

   Note: The file **AX_starterkit.adb** appears under the **Designer Files** on the **File Manager** tabbed page in Libero IDE. To reopen the file in Designer, right-click the file and select **Open**.

# Step 7 – Perform a Timing Simulation with Back-Annotated Timing

After completing the place-and-route and back-annotation of the design, perform a timing simulation with the Model*Sim* HDL simulator.

### *To perform a timing simulation:*

1. **Right-click the Model*Sim Simulator* icon in the Libero IDE Process window, or right-click the *AX_starterkit* file on the Design Hierarchy tabbed page and select *Run Post-Layout Simulation*.**

   This launches the **Model*Sim* Simulator,** which compiles the source file and test bench.

   Once the compilation completes, the simulator runs for 1000 ns and a wave window opens to display the simulation results.

2. **Using the *run 9us* command, extend the simulation to 10 µs.**
3. **Scroll through the wave window to verify that the andgate works correctly.**

# Step 8 – Generate the Programming File

This step generates the necessary file to program the Axcelerator device.

### *To generate a programming file:*

1. **Right-click the *AX_starterkit* file on the Design Hierarchy tabbed page to open Designer.**
2. **Left click the *Fuse* icon in Designer.**

This opens the **Generate Programming Files** dialog box, as shown in Figure 3-39.


Figure 3-39. Generate Programming Files

3. **Accept the default settings and file name and click *OK*.**

On successful completion, the FUSE icon turns green. The programming file name **AX_startkit.afm** is saved to the Libero IDE, appearing on the **File Manager** tabbed page under **Implementation** files.

# Step 9 – Program the Device

Two options are available for programming an Axcelerator device.

### *To use Actel Online Prototyping Service:*

1. **Locate the coupon in the *AX-EVAL-KIT* that allows you to submit five prototype designs to the Actel Online Prototyping Service (OPS).**
2. **Access the Actel OPS at www.actel.com/ops.**

3. **Submit designs as Actel programming (AFM) files.** You may also use this service for a fee for subsequent designs.

### *To use Silicon Sculptor Software:*

1. **Use the Designer software to generate an *AFM* file for the part.**
2. **See *Axcelerator Adapter Modules* on the Actel website.** Ensure you have the appropriate programming adapter module.
3. **Use the Silicon Sculptor software to program the part once.** Obtain the most recent version of Silicon Sculptor software at the Actel website. You should use the most recent version of Silicon Sculptor software with AFM files generated by the latest version of the Designer software.
4. **Contact Actel if you need to obtain additional parts.**

   Note: Both AX250-PQ208 and AX500-PQ208 parts will work with the evaluation board included with the starter kit.

# A

# Board Connections

Table A-1 lists the board connections for the AX250 device.

Table A-1. AX250 Board Connections

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 1 | $V_{CCDA}$ | | |
| 2 | $V_{CCA}$ | | |
| 3 | UART_SIN | Bank 7 - Pin 5 | JX1 - Pin B8 |
| 4 | UART_SOUT | Bank 7 - Pin 6 | JX1 - Pin B9 |
| 5 | I/O | Bank 7 - Pin 7 | JX1 - Pin A9 |
| 6 | I/O | Bank 7 - Pin 8 | JX1 - Pin B10 |
| 7 | I/O | Bank 7 - Pin 9 | JX1 - Pin B11 |
| 8 | $V_{CCIB7}$ | Bank 7 - Pin 3 | |
| 9 | GND | | |
| 10 | I/O | Bank 7 - Pin 10 | JX1 - Pin A11 |
| 11 | I/O | Bank 7 - Pin 11 | JX1 - Pin B12 |
| 12 | I/O | Bank 7 - Pin 12 | JX1 - Pin B13 |
| 13 | I/O | Bank 7 - Pin 13 | JX1 - Pin A13 |
| 14 | $V_{CCA}$ | | |
| 15 | GND | | |
| 16 | I/O | Bank 7 - Pin 14 | JX1 - Pin B14 |
| 17 | I/O | Bank 7 - Pin 15 | JX1 - Pin B15 |
| 18 | I/O | Bank 7 - Pin 16 | JX1 - Pin A15 |
| 19 | I/O | Bank 7 - Pin 17 | JX1 - Pin B16 |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 20 | $V_{CCIB7}$ | Bank 7 - Pin 3 | |
| 21 | GND | | |
| 22 | I/O | Bank 7 - Pin 18 | JX1 - Pin B17 |
| 23 | I/O | Bank 7 - Pin 19 | JX1 - Pin A17 |
| 24 | I/O | Bank 7 - Pin 20 | JX1 - Pin B18 |
| 25 | I/O | Bank 7 - Pin 21 | JX1 - Pin B19 |
| 26 | $V_{CCDA}$ | | |
| 27 | I/O | Bank 6 - Pin 5 | JX1 - Pin A19 |
| 28 | I/O | Bank 6- Pin 6 | JX1 - Pin B20 |
| 29 | I/O | Bank 6 - Pin 7 | JX1 - Pin B21 |
| 30 | I/O | Bank 6 - Pin 8 | JX1 - Pin A21 |
| 31 | $V_{CCIB6}$ | Bank 6 - Pin 3 | |
| 32 | GND | | |
| 33 | I/O | Bank 6- Pin 9 | JX1 - Pin B22 |
| 34 | I/O | Bank 6 - Pin 10 | JX1 - Pin B23 |
| 35 | I/O | Bank 6 - Pin 11 | JX1 - Pin A23 |
| 36 | I/O | Bank 6 - Pin 12 | JX1 - Pin B24 |
| 37 | I/O | Bank 6 - Pin 13 | JX1 - Pin B25 |
| 38 | $V_{CCA}$ | | |
| 39 | GND | | |
| 40 | I/O | Bank 6 - Pin 14 | JX1 - Pin A25 |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 41 | I/O | Bank 6 - Pin 15 | JX1 - Pin B26 |
| 42 | I/O | Bank 6 - Pin 16 | JX1 - Pin B27 |
| 43 | I/O | Bank 6 - Pin 17 | JX1 - Pin A27 |
| 44 | I/O | Bank 6 - Pin 18 | JX1 - Pin B28 |
| 45 | $V_{CCIB6}$ | Bank 6- Pin 3 | |
| 46 | GND | | |
| 47 | I/O | Bank 6 - Pin 19 | JX1 - Pin B29 |
| 48 | I/O | Bank 6 - Pin 20 | JX1 - Pin A29 |
| 49 | I/O | Bank 6 - Pin 21 | JX1 - Pin B30 |
| 50 | I/O | Bank 6 - Pin 22 | JX1 - Pin B31 |
| 51 | GND | | |
| 52 | $V_{CCA}$ | | |
| 53 | $V_{CCDA}$ | | |
| 54 | I/O | Bank 5 - Pin 5 | JX1 - Pin A31 |
| 55 | I/O | Bank 5 - Pin 6 | JX1 - Pin B32 |
| 56 | I/O | Bank 5 - Pin 7 | JX1 - Pin B33 |
| 57 | I/O | Bank 5 - Pin 8 | JX1 - Pin A33 |
| 58 | $V_{CCIB5}$ | Bank 5 - Pin 3 | |
| 59 | GND | | |
| 60 | I/O | Bank 5 - Pin 9 | JX1 - Pin B34 |
| 61 | I/O | Bank 5 - Pin 10 | JX1 - Pin B35 |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 62 | I/O | Bank 5 - Pin 11 | JX1 - Pin A35 |
| 63 | $V_{CCDA}$ | | |
| 64 | $V_{CCA}$ | | |
| 65 | GND | | |
| 66 | I/O | Bank 5 - Pin 12 | JX1 - Pin B36 |
| 67 | I/O | Bank 5 - Pin 13 | JX1 - Pin B37 |
| 68 | $V_{CCIB5}$ | Bank 5 - Pin 3 | |
| 69 | GND | | |
| 70 | CLK_SOCKET_IN | Bank 5 - Pin 18 | JX1 - Pin A39 |
| 71 | MANUAL_CLK_IN | Bank 5 - Pin 16 | JX1 - Pin A37 |
| 72 | $V_{CCPLH}$ | | |
| 73 | $V_{COMPPLH}$ | | |
| 74 | $V_{CCPLG}$ | | |
| 75 | $V_{COMPPLG}$ | | |
| 76 | OSC_CLK_IN | Bank 5 - Pin 20 | |
| 77 | MULT_CLK_IN | Bank 5 - Pin 22 | |
| 78 | $V_{CCDA}$ | | |
| 79 | PRD | | |
| 80 | PRC | | |
| 81 | RXDATA1N | | |
| 82 | RXDATA1P | | |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 83 | $V_{CCPLF}$ | | |
| 84 | $V_{COMPPLF}$ | | |
| 85 | $V_{CCPLE}$ | | |
| 86 | $V_{COMPPLE}$ | | |
| 87 | TXDATA1N | | |
| 88 | TXDATA1P | | |
| 89 | $V_{CCIB4}$ | Bank 4- Pin 3 | |
| 90 | GND | | |
| 91 | RXDATA2N | | |
| 92 | RXDATA2P | | |
| 93 | $V_{CCA}$ | | |
| 94 | GND | | |
| 95 | $V_{CCDA}$ | | |
| 96 | TXDATA2N | | |
| 97 | TXDATA2P | | |
| 98 | $V_{CCIB4}$ | Bank 4 - Pin 3 | |
| 99 | GND | | |
| 100 | LED_8 | Bank 4 - Pin 19 | JX2 - Pin A40 |
| 101 | LED_7 | Bank 4 - Pin 17 | JX2 - Pin A39 |
| 102 | LED_6 | Bank 4 - Pin 15 | JX2 - Pin A38 |
| 103 | LED_5 | Bank 4 - Pin 13 | JX2 - Pin A37 |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 104 | GND | | |
| 105 | $V_{CCDA}$ | | |
| 106 | LED_4 | Bank 4 - Pin 11 | JX2 - Pin B36 |
| 107 | LED_3 | Bank 4 - Pin 9 | JX2 - Pin A36 |
| 108 | LED_2 | Bank 4 - Pin 7 | JX2 - Pin A35 |
| 109 | LED_1 | Bank 4 - Pin 5 | JX2 - Pin B34 |
| 110 | USER_4 | Bank 3 - Pin 21 | JX2 - Pin A34 |
| 111 | USER_3 | Bank 3 - Pin 19 | JX2 - Pin A33 |
| 112 | $V_{CCIB3}$ | Bank 3 - Pin 3 | |
| 113 | GND | | |
| 114 | USER_2 | Bank 3 - Pin 17 | JX2 - Pin B32 |
| 115 | USER_1 | Bank 3 - Pin 15 | JX2 - Pin A32 |
| 116 | I/O | Bank 3 - Pin 14 | JX2 - Pin A31 |
| 117 | I/O | Bank 3 - Pin 13 | JX2 - Pin B30 |
| 118 | $V_{CCA}$ | | |
| 119 | GND | | |
| 120 | I/O | Bank 3 - Pin 12 | JX2 - Pin A30 |
| 121 | I/O | Bank 3 - Pin 11 | JX2 - Pin A29 |
| 122 | I/O | Bank 3 - Pin 10 | JX2 - Pin B28 |
| 123 | I/O | Bank 3 - Pin 9 | JX2 - Pin A28 |
| 124 | $V_{CCIB3}$ | Bank 3 - Pin 3 | |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 125 | GND | | |
| 126 | I/O | Bank 3 - Pin 8 | JX2 - Pin A27 |
| 127 | I/O | Bank 3 - Pin 7 | JX2 - Pin B26 |
| 128 | I/O | Bank 3 - Pin 6 | JX2 - Pin A26 |
| 129 | I/O | Bank 3 - Pin 5 | JX2 - Pin A25 |
| 130 | $V_{CCDA}$ | | |
| 131 | I/O | Bank 2 - Pin 22 | JX2 - Pin B24 |
| 132 | I/O | Bank 2 - Pin 21 | JX2 - Pin A24 |
| 133 | I/O | Bank 2 - Pin 20 | JX2 - Pin A23 |
| 134 | I/O | Bank 2 - Pin 19 | JX2 - Pin B22 |
| 135 | $V_{CCIB2}$ | Bank 2 - Pin 3 | |
| 136 | GND | | |
| 137 | I/O | Bank 2 - Pin 18 | JX2 - Pin A22 |
| 138 | I/O | Bank 2 - Pin 17 | JX2 - Pin A21 |
| 139 | I/O | Bank 2- Pin 16 | JX2 - Pin B20 |
| 140 | I/O | Bank 2 - Pin 15 | JX2 - Pin A20 |
| 141 | I/O | Bank 2 - Pin 14 | JX2 - Pin A19 |
| 142 | $V_{CCA}$ | | |
| 143 | GND | | |
| 144 | I/O | Bank 2 - Pin 13 | JX2 - Pin B18 |
| 145 | I/O | Bank 2 - Pin 12 | JX2 - Pin A18 |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 146 | I/O | Bank 2 - Pin 11 | JX2 - Pin A17 |
| 147 | I/O | Bank 2 - Pin 10 | JX2 - Pin B16 |
| 148 | I/O | Bank 2 - Pin 9 | JX2 - Pin A16 |
| 149 | $V_{CCIB2}$ | Bank 2 - Pin 3 | |
| 150 | GND | | |
| 151 | I/O | Bank 2 - Pin 8 | JX2 - Pin A15 |
| 152 | I/O | Bank 2 - Pin 7 | JX2 - Pin B14 |
| 153 | I/O | Bank 2 - Pin 6 | JX2 - Pin A14 |
| 154 | I/O | Bank 2 - Pin 5 | JX2 - Pin A13 |
| 155 | GND | | |
| 156 | $V_{CCA}$ | | |
| 157 | $V_{CCDA}$ | | |
| 158 | GND | | |
| 159 | I/O | Bank 0/1 - Pin 22 | JX2 - Pin B12 |
| 160 | I/O | Bank 0/1 - Pin 21 | JX2 - Pin A12 |
| 161 | I/O | Bank 0/1 - Pin 20 | JX2 - Pin A11 |
| 162 | I/O | Bank 0/1 - Pin 19 | JX2 - Pin B10 |
| 163 | $V_{CCIB1}$ | Bank 0/1 - Pin 3 | |
| 164 | GND | | |
| 165 | I/O | Bank 0/1 - Pin 18 | JX2 - Pin A10 |
| 166 | I/O | Bank 0/1 - Pin 17 | JX2 - Pin A9 |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 167 | $V_{CCDA}$ | | |
| 168 | $V_{CCA}$ | | |
| 169 | GND | | |
| 170 | I/O | Bank 0/1 - Pin 16 | JX2 - Pin B8 |
| 171 | I/O | Bank 0/1 - Pin 15 | JX2 - Pin A8 |
| 172 | $V_{CCIB1}$ | Bank 0/1 - Pin 1 | |
| 173 | GDN | | |
| 174 | I/O | Bank 0/1 - Pin 14 | JX2 - Pin A7 |
| 175 | I/O | Bank 0/1 - Pin 13 | JX2 - Pin B6 |
| 176 | $V_{CCPLD}$ | | |
| 177 | $V_{COMPPLD}$ | | |
| 178 | $V_{CCPLC}$ | | |
| 179 | $V_{COMPPLC}$ | | |
| 180 | I/O | Bank 0/1 - Pin 12 | JX2 - Pin A6 |
| 181 | I/O | Bank 0/1 - Pin 11 | JX2 - Pin A5 |
| 182 | $V_{CCDA}$ | | |
| 183 | PRB | | |
| 184 | PRA | | |
| 185 | I/O | Bank 0/1 - Pin 10 | JX2 - Pin B4 |
| 186 | I/O | Bank 0/1 - Pin 9 | JX2 - Pin A4 |
| 187 | $V_{CCPLB}$ | | |

Table A-1. AX250 Board Connections (Continued)

| AX250 Pin Number | Board Connect | Header Connect (I/O only) | P160 Connect (I/O only) |
|---|---|---|---|
| 188 | $V_{COMPPLB}$ | | |
| 189 | $V_{CCPLA}$ | | |
| 190 | $V_{COMPPLA}$ | | |
| 191 | I/O | Bank 0/1 - Pin 8 | JX2 - Pin A3 |
| 192 | HCLK | | |
| 193 | $V_{CCIB0}$ | | |
| 194 | GND | | |
| 195 | $V_{CCA}$ | | |
| 196 | GND | | |
| 197 | I/O | Bank 0/1 - Pin 7 | JX2 - Pin B36 |
| 198 | I/O | Bank 0/1 - Pin 6 | JX2 - Pin A2 |
| 199 | I/O | Bank 0/1 - Pin 5 | JX2 - Pin A1 |
| 200 | $V_{CCIB0}$ | | |
| 201 | GND | | |
| 202 | $V_{CCDA}$ | | |
| 203 | TDO | | |
| 204 | TDI | | |
| 205 | TCK | | |
| 206 | TMS | | |
| 207 | TRST | | |
| 208 | GND/LP | | |

B

# P160 Connections

Table B-1 lists the P160 connections.

Table B-1. P160 Connections

| JX1<br>P160 Left Header | Board Connect | JX2<br>P160 Right Header | Board Connect |
|---|---|---|---|
| A1 | NC | A1 | P199 |
| A2 | GND | A2 | P198 |
| A3 | NC | A3 | P191 |
| A4 | 5.0 V | A4 | P186 |
| A5 | NC | A5 | P181 |
| A6 | GND | A6 | P180 |
| A7 | NC | A7 | P174 |
| A8 | 3.3 V | A8 | P171 |
| A9 | P5 | A9 | P166 |
| A10 | GND | A10 | P165 |
| A11 | P10 | A11 | P161 |
| A12 | 2.5 V | A12 | P160 |
| A13 | P13 | A13 | P154 |
| A14 | GND | A14 | P153 |
| A15 | P18 | A15 | P151 |
| A16 | 5.0 V | A16 | P148 |
| A17 | P23 | A17 | P146 |
| A18 | GND | A18 | P145 |
| A19 | P27 | A19 | P141 |

Table B-1. P160 Connections (Continued)

| JX1<br>P160 Left Header | Board Connect | JX2<br>P160 Right Header | Board Connect |
|---|---|---|---|
| A20 | 3.3 V | A20 | P140 |
| A21 | P30 | A21 | P138 |
| A22 | GND | A22 | P137 |
| A23 | P35 | A23 | P133 |
| A24 | 2.5 V | A24 | P132 |
| A25 | P40 | A25 | P129 |
| A26 | GND | A26 | P128 |
| A27 | P43 | A27 | P126 |
| A28 | 5.0 V | A28 | P123 |
| A29 | P48 | A29 | P121 |
| A30 | GND | A30 | P120 |
| A31 | P54 | A31 | P116 |
| A32 | 3.3 V | A32 | USER_1 |
| A33 | P57 | A33 | USER_3 |
| A34 | GND | A34 | USER_4 |
| A35 | P62 | A35 | LED_2 |
| A36 | 2.5 V | A36 | LED_3 |
| A37 | MANUAL_CLK_IN | A37 | LED_5 |
| A38 | GND | A38 | LED_6 |
| A39 | CLK_SOCKET_IN | A39 | LED_7 |
| A40 | 5.0 V | A40 | LED_8 |

Table B-1. P160 Connections (Continued)

| JX1<br>P160 Left Header | Board Connect | JX2<br>P160 Right Header | Board Connect |
|---|---|---|---|
| B1 | NC | B1 | GND |
| B2 | NC | B2 | P197 |
| B3 | NC | B3 | 5.0 V |
| B4 | NC | B4 | P185 |
| B5 | NC | B5 | GND |
| B6 | NC | B6 | P175 |
| B7 | NC | B7 | 3.3 V |
| B8 | UART_SIN | B8 | P170 |
| B9 | UART_SOUT | B9 | GND |
| B10 | P6 | B10 | P162 |
| B11 | P7 | B11 | 2.5 V |
| B12 | P11 | B12 | P159 |
| B13 | P12 | B13 | GND |
| B14 | P16 | B14 | P152 |
| A15 | P17 | A15 | 5.0 V |
| B16 | P19 | B16 | P147 |
| B17 | P22 | B17 | GND |
| B18 | P24 | B18 | P144 |
| B19 | P25 | B19 | 3.3 V |
| B20 | P28 | B20 | P139 |
| B21 | P29 | B21 | GND |

Table B-1. P160 Connections (Continued)

| JX1<br>P160 Left Header | Board Connect | JX2<br>P160 Right Header | Board Connect |
|---|---|---|---|
| B22 | P33 | B22 | P134 |
| B23 | P34 | B23 | 2.5 V |
| B24 | P36 | B24 | P131 |
| B25 | P37 | B25 | GND |
| B26 | P41 | B26 | P127 |
| B27 | P42 | B27 | 5.0 V |
| B28 | P44 | B28 | P122 |
| B29 | P47 | B29 | GND |
| B30 | P49 | B30 | P117 |
| B31 | P50 | B31 | 3.3 V |
| B32 | P55 | B32 | USER_2 |
| B33 | P56 | B33 | GND |
| B34 | P60 | B34 | LED_1 |
| B35 | P61 | B35 | 2.5 V |
| B36 | P66 | B36 | LED_4 |
| B37 | P67 | B37 | GND |
| B38 | NC | B38 | NC |
| B39 | NC | B39 | 5.0 V |
| B40 | NC | B40 | NC |

# C

# Board Schematic

For board schematics, refer to http://www.actel.com/documents/AXEV_CS_REV2_SEP28.pdf.

D

# 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 401 500**
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.actel.com/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

***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 401 450 • Fax +44 (0) 1276 401 490*

***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*** *• Suite 2114 Two Pacific Place • 88 Queensway, Admiralty Hong Kong*
*Phone +852.2185.6460 Fax +852.2185.64688*

50200037-1/02.05