

Lattice Diamond Tutorial



October 2018

Copyright

Copyright © 2017 Lattice Semiconductor Corporation. All rights reserved. This document may not, in whole or part, be reproduced, modified, distributed, or publicly displayed without prior written consent from Lattice Semiconductor Corporation (“Lattice”).

Trademarks

All Lattice trademarks are as listed at www.latticesemi.com/legal. Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. All other trademarks are the property of their respective owners.

Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL LATTICE OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF LATTICE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Lattice may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. Lattice makes no commitment to update this documentation. Lattice reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. Lattice recommends its customers obtain the latest version of the relevant information to establish that the information being relied upon is current and before ordering any products.

Type Conventions Used in This Document

Convention	Meaning or Use
Bold	Items in the user interface that you select or click. Text that you type into the user interface.
<i><Italic></i>	Variables in commands, code syntax, and path names.
Ctrl+L	Press the two keys at the same time.
<code>Courier</code>	Code examples. Messages, reports, and prompts from the software.
<code>...</code>	Omitted material in a line of code.
<code>.</code> <code>.</code> <code>.</code>	Omitted lines in code and report examples.
[]	Optional items in syntax descriptions. In bus specifications, the brackets are required.
()	Grouped items in syntax descriptions.
{ }	Repeatable items in syntax descriptions.
	A choice between items in syntax descriptions.

Contents

Lattice Diamond Tutorial	7
Learning Objectives	7
Time to Complete This Tutorial	8
System Requirements	8
Accessing Online Help and Diamond User Guide	9
About the Tutorial Design	9
About the Tutorial Data Flow	9
Task 1: Create a New Lattice Diamond Project	11
Task 2: Create an IPexpress Module	17
Task 3: Verify Functionality with Simulation	20
Task 4: Inspect Strategy Settings	22
Task 5: Examine Resources	24
Task 6: Run Synthesis Process	27
Task 7: Set Timing and Location Assignments	28
Task 8: Running Place and Route	33
Task 9: Examine Post Place and Route Results	36
Task 10: Adjust Static Timing Constraints and Review Results	40
Task 11: Comparing Multiple Place and Route Runs	43
Task 12: Analyze Power Consumption	44
Task 13: Run Export Utility Programs	47
Task 14: Download a Bitstream to an FPGA	47
Task 15: Convert a File Using Deployment Tool	49
Task 16: Use Reveal Inserter to Add On-chip Debug Logic	53
Setting Up the Trigger Units	55
Setting Up the Trigger Expressions	56
Inserting the Debug Logic	57
Generating a Bitstream and Programming the FPGA	59

Task 17: Use Reveal Logic Analyzer to Perform Logic Analysis	60
Creating a New Reveal Logic Analyzer Project	60
Running Logic Analyzer	63
Summary of Accomplishments	64
Recommended References	65

Lattice Diamond Tutorial

The next generation design tool for FPGA design, Lattice Diamond™, is designed to address the needs of high-density FPGA designers.

This tutorial leads you through all the basic steps of designing and implementing a mixed VHDL, Verilog, and EDIF design targeted to the LatticeECP3 device family. It shows you how to use several processes, tools, and reports from the Lattice Diamond software to import sources, run design analysis, view design hierarchy, and inspect strategy settings. The tutorial then proceeds to step through the processes of adding and editing a strategy, specifying the synthesis requirements, examining the device resources, setting timing and location assignments, and editing preferences to configure the filter to implement the design to the target device.

Learning Objectives

When you have completed this tutorial, you should be able to do the following:

- ▶ Create a new Lattice Diamond project
- ▶ Create an IPexpress module
- ▶ Verify functionality with simulation
- ▶ Inspect strategy settings
- ▶ Examine resources
- ▶ Run synthesis processes
- ▶ Set timing and location assignments
- ▶ Run place and route
- ▶ Examine post place and route results
- ▶ Adjust static timing constraints and review results

- ▶ Compare multiple place and route runs
- ▶ Analyze power consumption
- ▶ Run export utility programs
- ▶ Download a bitstream to an FPGA
- ▶ Convert a file using Deployment Tool
- ▶ Use the Reveal Inserter to add on-chip debug logic
- ▶ Use the Reveal Logic Analyzer to perform logic analysis

Time to Complete This Tutorial

The time to complete this tutorial is about 90 minutes.

System Requirements

The following software is required to complete the tutorial:

- ▶ Lattice Diamond software
- ▶ (Optional) LatticeECP3 Versa Development Kit

Note

The subscription version of Diamond software supports LatticeECP3 devices.

LatticeECP3 device support is disabled in the free version of Diamond software.

Users of the free version of Diamond software should request the “Diamond Free License - For Versa Kit Only.” This license enables the user to select the LatticeECP3 LFE3-35EA device, which is required to perform the tutorial.

Many of the tasks in this tutorial can be performed without an actual LatticeECP3 Versa Development Kit, but the low-cost LatticeECP3 Versa Development Kit is recommended to perform all of the tasks in this tutorial.

- ▶ To download Diamond software, go to:
<http://www.latticesemi.com/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond.aspx>
You must have a Lattice web account to access this web page.
 - ▶ To obtain a “Diamond Free License - For Versa Kit Only,” go to:
<http://www.latticesemi.com/Support/Licensing/DiamondAndiCEcube2SoftwareLicensing/Versa>. You must have a Lattice web account to access this web page.
 - ▶ To purchase a low-cost LatticeECP3 Versa Development Kit, go to:
<http://www.latticesemi.com/Products/DevelopmentBoardsAndKits/LatticeECP3VersaDevelopmentKit.aspx>. You must have a Lattice web account to access this web page.
-

Accessing Online Help and Diamond User Guide

You can find online help information on any tool included in the tutorial at any time by choosing **Help > Lattice Diamond Help**.

Another excellent resource is the *Diamond User Guide*, available from the start page of Lattice Diamond online help.

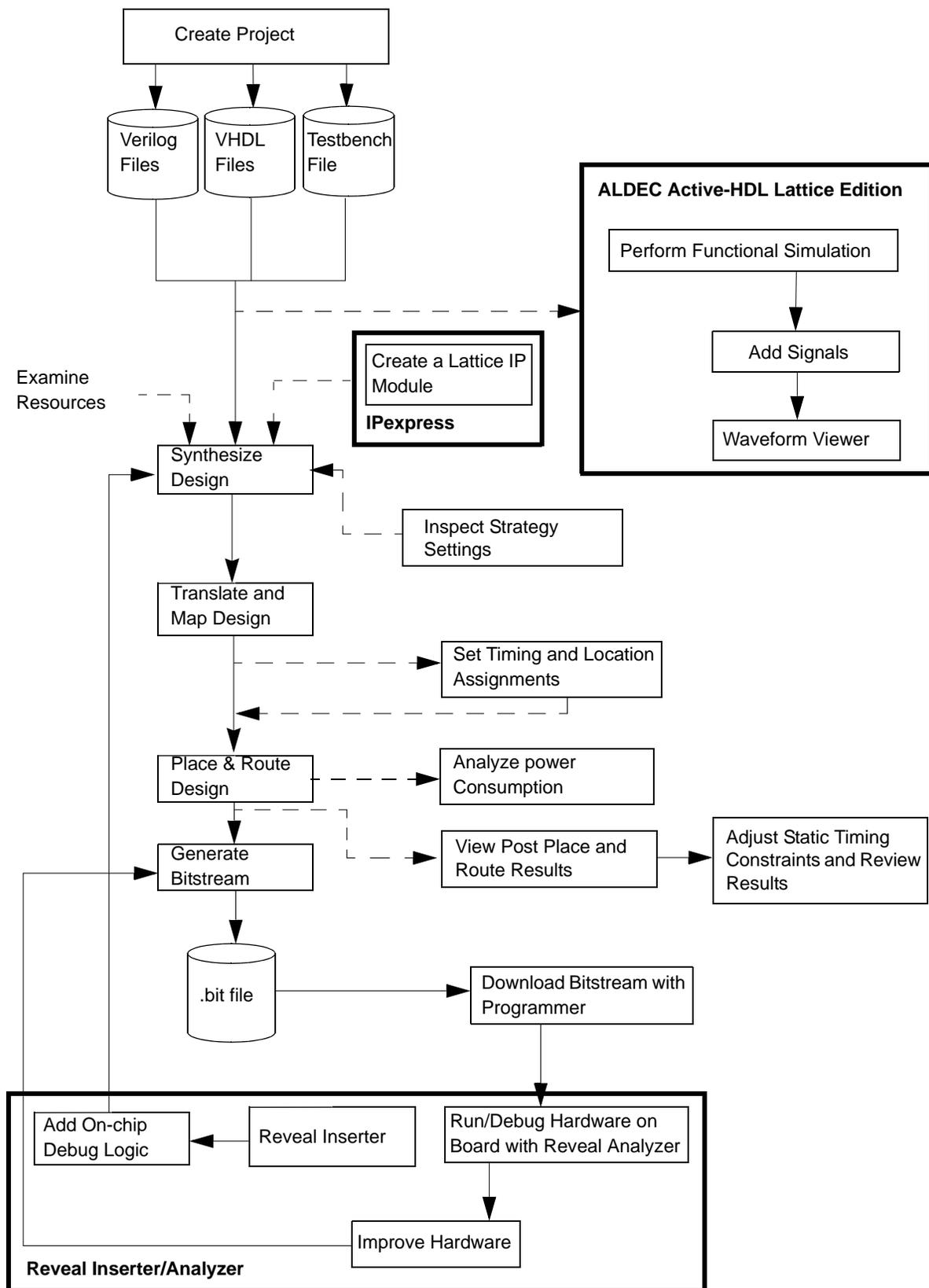
About the Tutorial Design

The design in this tutorial consists of a Verilog HDL module, two VHDL modules, and one EDIF module. The design that you create is targeted to LatticeECP3 device families.

About the Tutorial Data Flow

Figure 1 illustrates the tutorial data flow through the system. You may find it helpful to refer to this diagram as you move through the tutorial tasks.

Figure 1: Tutorial Data Flow



Task 1: Create a New Lattice Diamond Project

Projects are used to manage input files, preferences, and optimization options related to an FPGA implementation. While there are a number of tasks you can perform independent of a project, most designs start with creating a new project.

Note

Some of the screen captures in this tutorial may have been taken from a version of Lattice Diamond that differs from the one you are using. There may be slight differences in the graphical user interface (GUI), but the software functions the same.

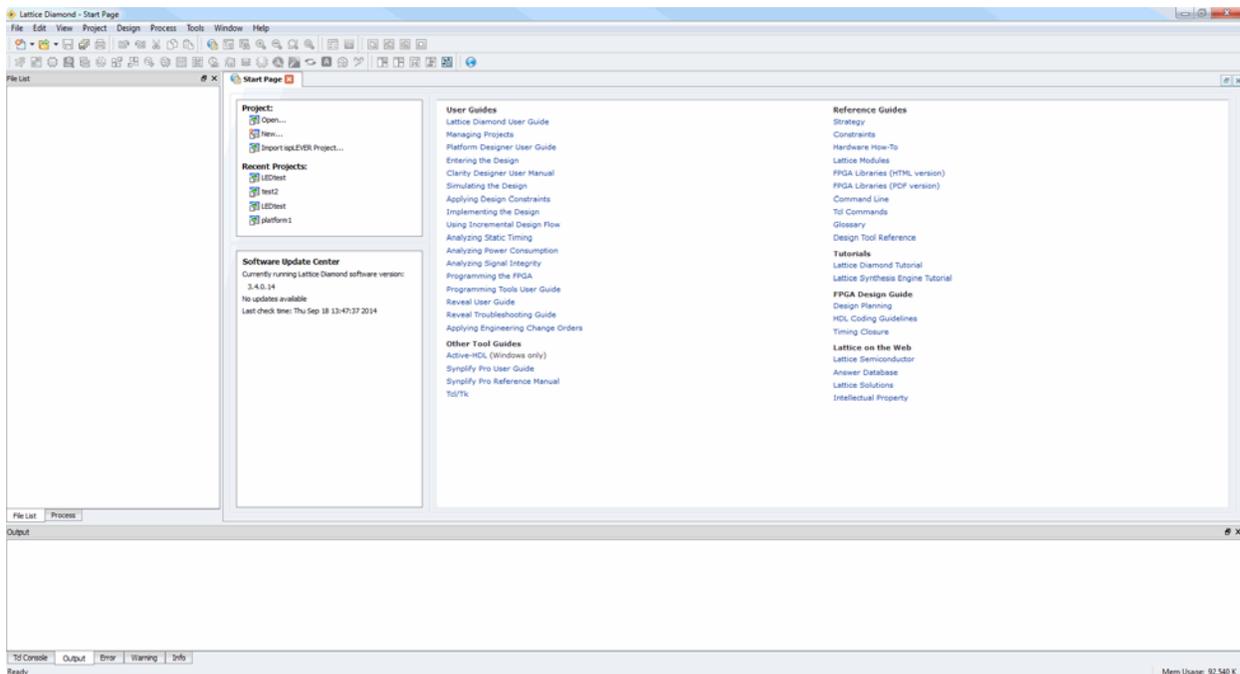
To create a new project:

1. Do one of the following depending on your operating system:
 - ▶ From your Windows desktop, choose **Start > Programs > Lattice Diamond >  Lattice Diamond**.
 - ▶ From your Linux platform shell window or C-shell window, execute:


```
<install_path>/bin/linux/diamond
```

The Lattice Diamond Design Environment appears, as shown in Figure 2.

Figure 2: Diamond Design Environment



The initial layout provides the Start Page, which provides a list of common project actions like Open to open a pre-existing project and New to run the New Project wizard. Hyperlinks in the right pane of the Start Page provide access to user guides, reference material, and online resources available from www.latticesemi.com.

For almost all questions, the place to start is Lattice Diamond's online Help. It describes the FPGA design flow using Diamond, the libraries of logic design elements, and the details of the Diamond design tools. The Help also provides easy access to many other information sources. The Help can be accessed from **Help > Lattice Diamond Help**.

2. Open a new project in one of the following ways:
 - ▶ In the Start page, under **Project**, click **New**.
 - ▶ From the Diamond main window choose **File > New >  Project**.
 - ▶ Click the down arrow in the  icon from the toolbar and then choose **Project**.

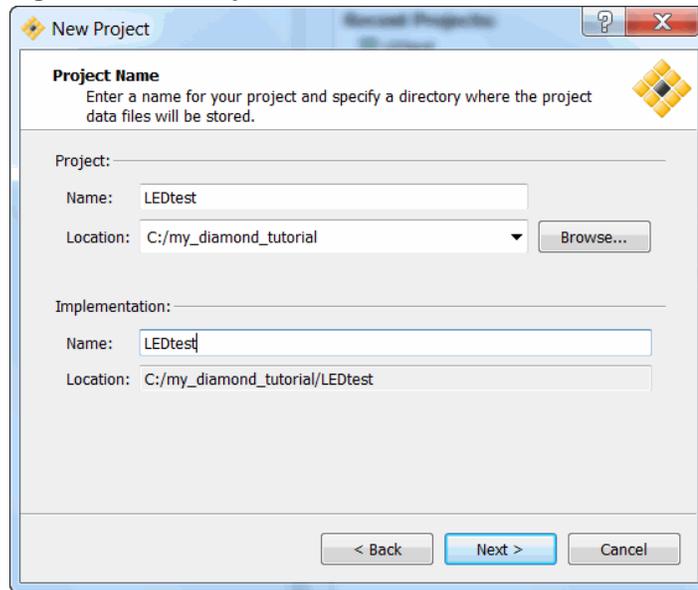
The New Project wizard opens.

3. Click **Next**.
4. Specify the project name: **LEDtest**.

Note

File names for Diamond projects and project source files must start with a letter (A-Z, a-z) and must contain only alphanumeric characters (A-Z, a-z, 0-9) and underscores (_).

5. Click **Browse**. In the Project Location dialog box, browse to where you want to store the project's files, such as **C:/my_diamond_tutorial**, as shown in Figure 3. Click **Select Folder**.
6. Specify the implementation name: **LEDtest**. The directory to store the implementation is automatically displayed in the Location box. We will talk about creating a new implementation later in this tutorial.

Figure 3: New Project Window7. Click **Next**.

The Add Source dialog box appears.

8. Click **Add Source**.

The Import File dialog box appears.

9. Navigate to the folder containing the source files, which are located in the *<diamond_install_directory>/docs/tutorial/Diamond_tutorial* directory. Select the following files in the directory:

- ▶ clockDivider.v
- ▶ count4.v
- ▶ count8.vhd
- ▶ LEDtest.v
- ▶ testbench.v
- ▶ topcount.v
- ▶ typepackage.vhd

and click **Open**.

The Add Source step of the Wizard appears with all the selected source files added.

10. Select **Copy source to implementation directory**.11. Click **Next**.

The Device Selector dialog box appears.

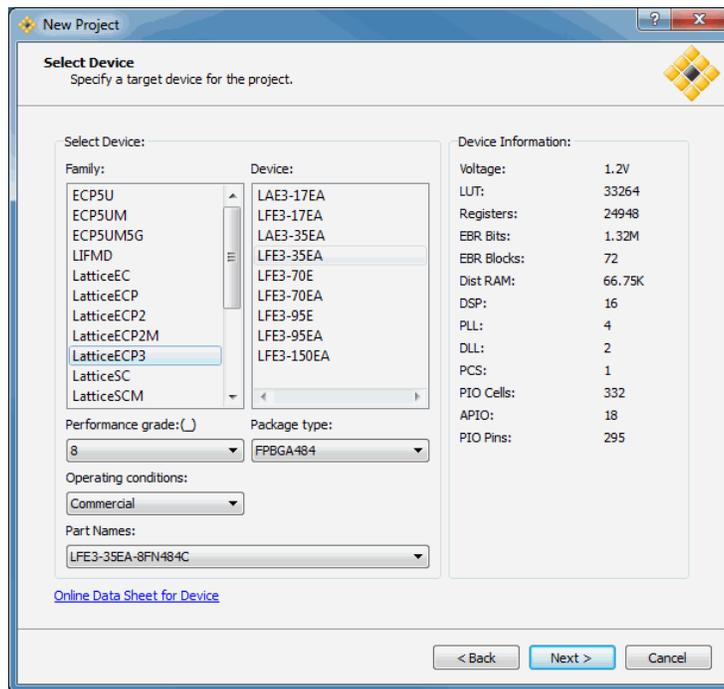
12. Select the following device options:

- ▶ Family: **LatticeECP3**
- ▶ Device: **LFE3-35EA**

- ▶ Performance Grade: **8**
- ▶ Package type: **FPBGA484**
- ▶ Operating Conditions: **Commercial**
- ▶ Part Names: **LFE3-35EA-8FN484C**

The dialog box should resemble Figure 4.

Figure 4: New Project Wizard Device Selector Dialog Box



13. Click **Next**.

The Select Synthesis Tool dialog box opens.

14. Choose **Synplify Pro**.

15. Click **Next**.

The Project Information dialog box appears. The project information includes project name, location, implementation name, device, synthesis tool, and import source.

16. Click **Finish**.

The File List and Process views are populated and the Reports view appears.

The File List view displays the components of the project. The imported VHDL, Verilog, and EDIF files appear in the Input Files folder in the File List view. The File List view organizes project files by categories: Strategies, and Implementation including Input Files, Constraint Files, Debug Files, Script Files, and Analysis Files. You may adjust file order by dragging and dropping filenames in the list. Properties of each file are

accessed by right-clicking the file and selecting Properties from the pop-up menu.

Note

You can also see Area, I/O Assistant, Quick, and Timing listed in the Strategies folder in the File List view. These are predefined strategies supplied by Lattice Semiconductor that solve particular design requirements. For details of these predefined strategies, refer to the Diamond online Help.

When you create a new project in Diamond, a logical preference file (.lpf) is automatically generated and assigned the same name as the FPGA project.

For this tutorial a logical preference file named **pin_assignments.lpf** is provided and contains all the pin assignments needed to program this design project onto the LatticeECP3 FPGA. All changes that you make to logical constraints will be saved in this file until you create a new logical preference file or add another existing one.

17. In the File List, right-click **LPF Constraint files** and select **Add > Existing File**.

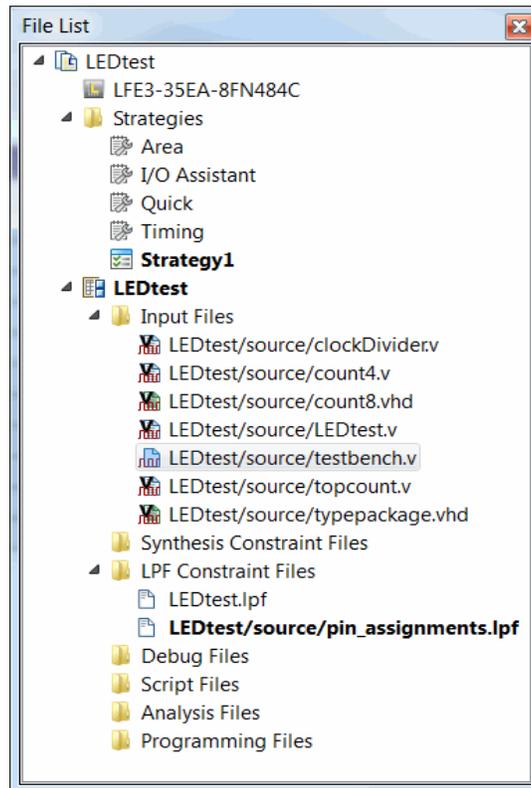
Add Existing File dialog box appears.

18. Navigate to *<diamond_install_directory>/docs/tutorial/Diamond_tutorial* and select the file **pin_assignments.lpf**. Choose **Copy file to Implementation's Source directory**, and click **Add**.

19. In the File List, right-click *pin_assignments.lpf* and choose **Set as Active Preference File**.

In the File List, right-click *testbench.v* and choose **Include for > Simulation** as shown in **Figure 5**.

20. Choose **View > Show Views > Hierarchy** to ensure that Hierarchy View will stay open for the whole session.

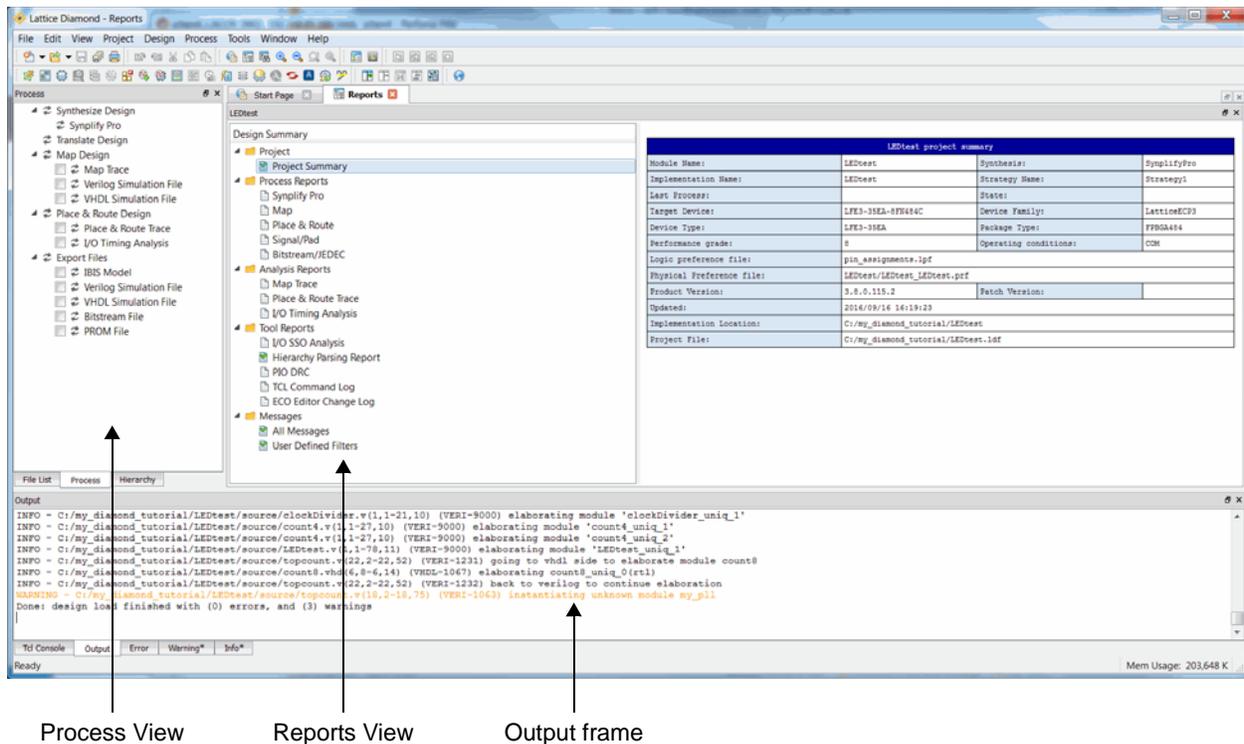
Figure 5: File List View

The Process view, shown in Figure 6, lists all the processes available, such as Synthesize Design, Translate Design, Map Design, Place & Route Design, and Export Files.

The Reports view allows you to examine and print process reports. There are two panes in the Reports view. The left pane lists the reports. The right pane displays the reports.

Log messages are displayed in the Output frame of the Diamond main window.

Figure 6: Process View and Reports View



Task 2: Create an IPexpress Module

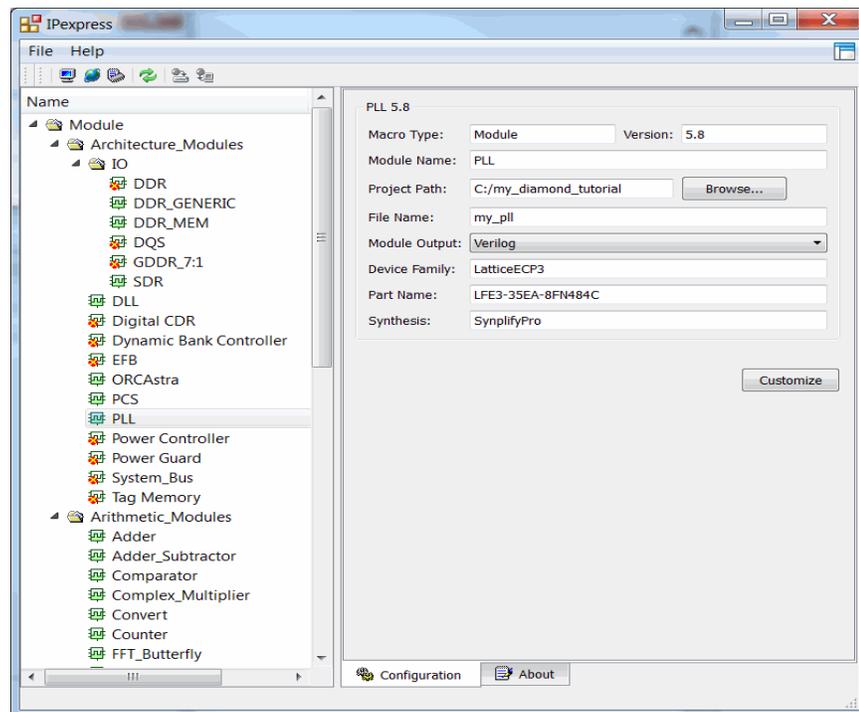
IPexpress is an easy way to use a collection of modules from Lattice Semiconductor. With IPexpress these modules can be extensively customized. They can be created as part of a specific project or as a library for multiple projects.

In this task, you will generate a phase lock loop (PLL) module to import into your design.

To Generate and Import a Module with IPexpress:

1. Choose **Tools > IPexpress**. The IPexpress tool appears.
2. Click the  icon in the upper right corner to detach the IPexpress window. An index of the available modules for the target device appears in the module tree on the left, as shown in Figure 7.

Figure 7: New Project Wizard Device Selector Dialog Box



3. In the module tree, under Module > Architecture_Modules, select **PLL**.
4. In the Configuration tab, all information is filled in from the design project except for File Name and Module Output. For this tutorial:
 - a. Enter **my_pll** as the File Name.
 - b. Enter **Verilog** as the Module Output.
 - c. Click **Customize**.

The Lattice FPGA Module -- PLL dialog box appears.

5. In the Configuration tab:
 - ▶ Select **Frequency Mode**.
 - ▶ For CLKI Frequency enter **100**.
 - ▶ For CLKOP/CLKOS (Non Bypass Mode) Desired Frequency enter **500**.
 - ▶ Select **Enable CLKOK**.
 - ▶ For CLKOK Desired Frequency enter **50**.
6. Click **Calculate**.
7. Click **Import IPX to Diamond project**.
8. Click **Generate**.

The IPexpress .ipx file is generated.
9. Click the **Generated Log** tab to view the log file, as shown in Figure 9.
10. Click **Close**.

The IPExpress .ipx file is imported into the Diamond project and appears in the File List as shown in Figure 9 on page 20.

11. Choose **File > Close Window** to close IPExpress.

Figure 8: IPExpress Generated Log Tab

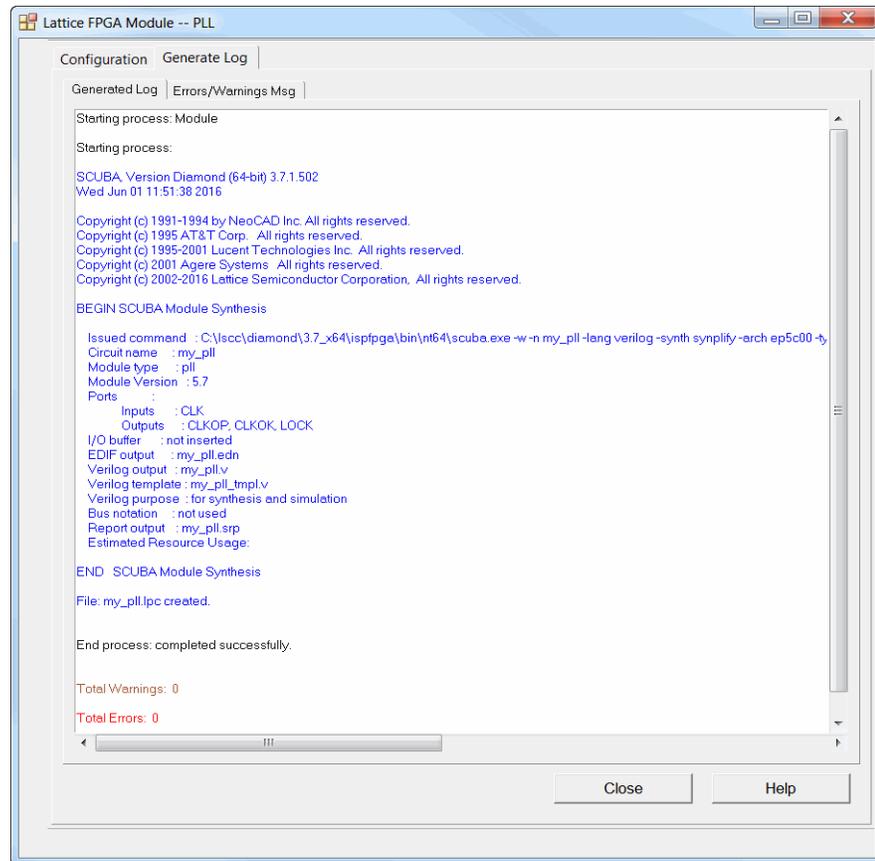
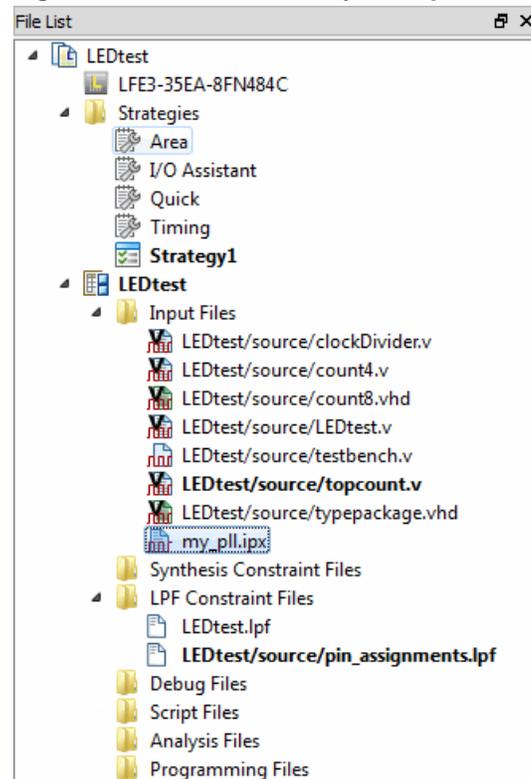


Figure 9: PLL Generated by IP Express Imported into Diamond Project

Task 3: Verify Functionality with Simulation

Diamond provides an interface to create a new simulation project file that can be imported into a standalone simulator. Diamond can export Active-HDL and ModelSim® simulation files.

Aldec® Active-HDL™ is an integrated environment designed for simulation of VHDL, Verilog/SystemVerilog, EDIF, and SystemC designs.

In this task, you will simulate the design using Active-HDL and analyze the resulting waveforms.

To simulate the design:

Make sure all the Source files are included in the simulation. In the File List pane, under **Input Files**, right-click on all of the source files except for testbench.v and select **Include For > Synthesis and Simulation**. The file testbench.v should be included for simulation only.

1. Choose **Tools > Simulation Wizard**.

The Simulation Wizard dialog box appears.

2. Click **Next**.

The Simulator Project Name dialog box appears.

3. Perform the following:
 - ▶ Specify Project name: **simulationfile**.
 - ▶ Make sure Active-HDL is selected for Simulator.
 - ▶ Click the  button to browse to where you want to store the project's file.
4. Click **Next**.
 - ▶ If you left the default selection for the project file location in the previous step, click **Yes** in the dialog box that asks whether you want to create a new folder.

The Process Stage dialog box appears.

5. Select **RTL** in the Process Stage dialog box.
6. Click **Next**.

The Add and Reorder Source dialog box appears.

7. Make sure all source files are present in the Source Files list. If the file `my_pll.v` that you created in “Task 2: Create an IPexpress Module” on page 17 is not present in the Source files list, click the Add File button , browse to your project directory, and choose the file **my_pll.v**.
8. Click **Next**.

The Parse HDL Files for Simulation dialog box appears.

9. Click **Next**.

The Summary dialog box appears. Make sure that **Run simulator**, **Add top-level signals to waveform display**, and **Run simulation** are selected.

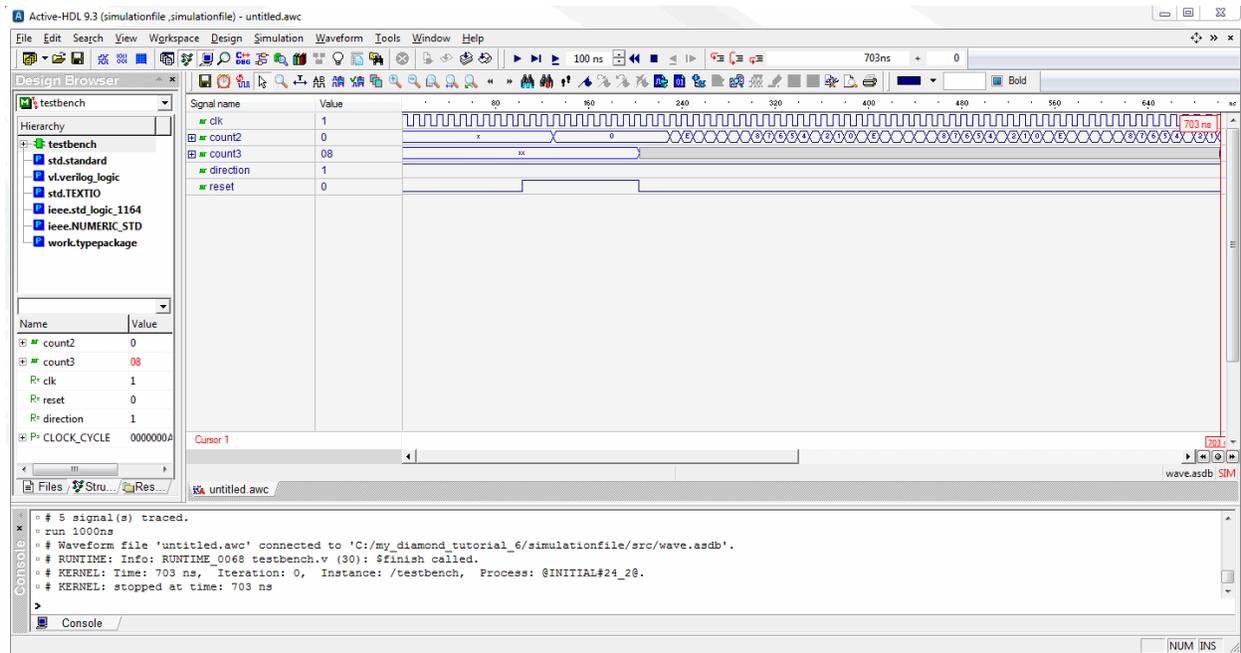
10. Click **Finish**.

The Aldec Active-HDL software is launched and the simulation starts automatically. After completing the simulation, a dialog box appears stating “Simulation has finished. There are no more vectors to simulate.”

11. Click **OK**.

The waveform appears, as shown in Figure 10.

Figure 10: Simulated Waveform

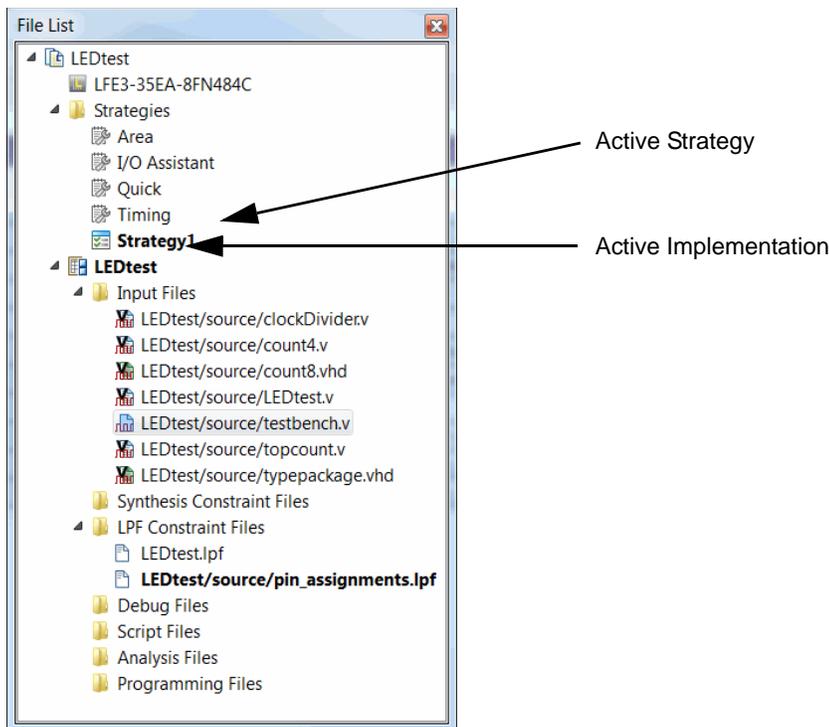


12. Choose **File > Exit** to close Active-HDL.
The Design Browser dialog box appears.
13. Click **OK**.
The Save File? dialog box appears.
14. Click **No**.

Task 4: Inspect Strategy Settings

Implementations define the design structural elements for a project, including source code, constraint files, and debug insertion. Implementation contains all source files, constraint files, debug files, scripts, and analysis files. Source can mix VHDL, Verilog, & EDIF. Files can be referenced or included in the implementation. Referenced files can be shared between implementations.

A strategy is a collection of settings for controlling the different stages of the implementation process (synthesis, map, place & route, and so on). Strategies can control whether the design is optimized for area or speed, how long place and route takes, and many other factors. Diamond provides a default strategy, which may be a good collection to start with, and some variations that you can try. You can modify Strategy1, as shown in Figure 11, and create other strategies to experiment with or to use in different circumstances. Predefined strategies can also be cloned and then modified.

Figure 11: Implementations and Strategies

To adjust synthesis settings:

1. From the File List view, double-click **Strategy1**.

The Strategies - Strategy1 dialog box, shown in Figure 12, appears.

2. Click **Synthesize Design > Synplify Pro**.

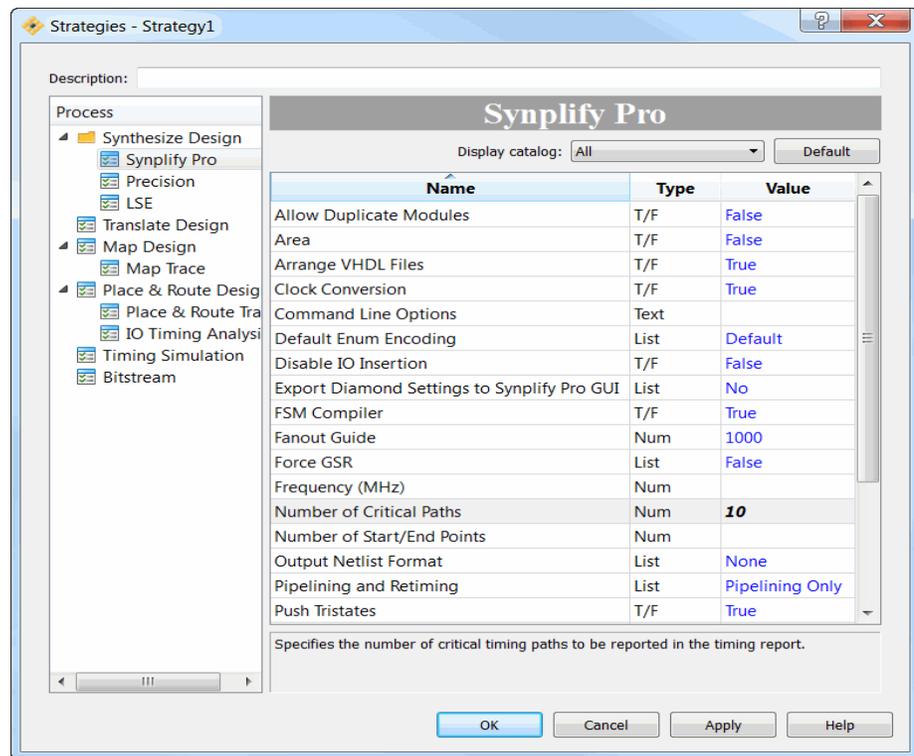
A set of default global synthesis timing constraints and optimization settings appears in the panel. Synplicity Synplify Pro® settings are displayed as the default in the dialog box.

For information on FPGA Design Constraints File (.fdc) usage in Synplify, see the *Synplify and Synplify Pro for Lattice Reference Manual* in the Synplify Pro for Lattice installation directory.

3. Specify the following setting for Synplify Pro, as shown in Figure 12:

Number of Critical Paths: **10**

Figure 12: Strategies -- Strategy 1

**Note**

When each strategy is selected, descriptive text appears in the lower panel of the dialog box. Default values in the strategies dialog box are shown in blue while changed values are shown in black.

3. Click **OK**. Global synthesis options are now set for the design.

Task 5: Examine Resources

Diamond provides visualization tools to help you understand and document the physical resources of the target device and the utilization of resources. You can browse and locate device features independent of the project's source files. After synthesis, you can view the calculated resource utilization.

To browse device resources:

1. Choose **Tools > Device View**.

The Device view appears.

2. Click the Detach Tool icon  in the upper right corner of the Device view tab to make it a separate window.

An index of the physical resources of the target device appears.

3. Expand the **Device** folder.

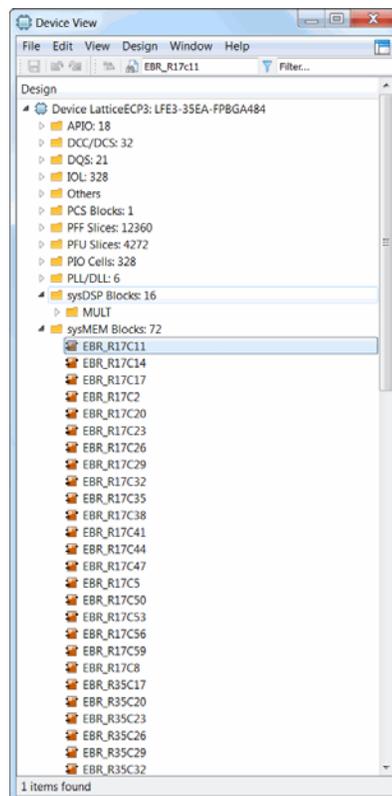
Several folders organized by feature type appear.

4. Expand the **sysDSP Blocks** and **sysMEM Blocks** folders.
5. Type **EBR_R17C11** (Embedded Block RAM Row 17, Column 11) into the Find entry box at the top of the Device View.
6. Press **Enter**.

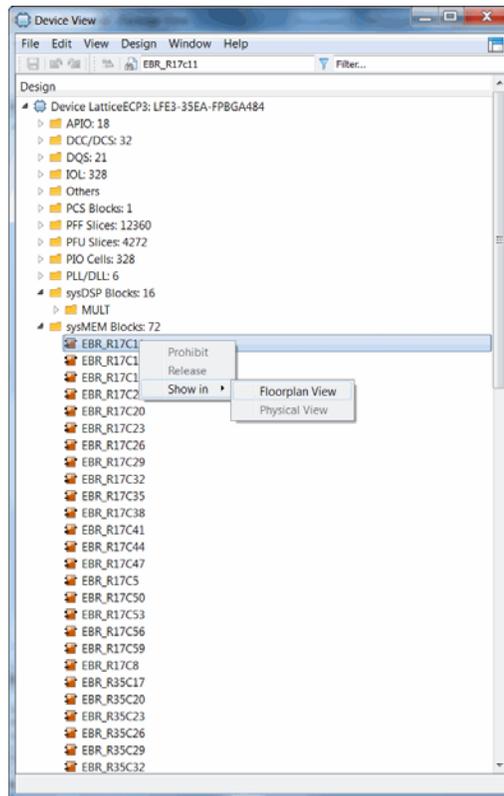
The EBR design symbol is highlighted, as shown in Figure 13.

LatticeECP3 devices contain one or more rows of sysMEM EBR blocks. EBRs are large, fast, dedicated 18Kbit memory blocks. Each sysMEM block can be configured in a variety of depths and widths as RAM and ROM functions.

Figure 13: Device View

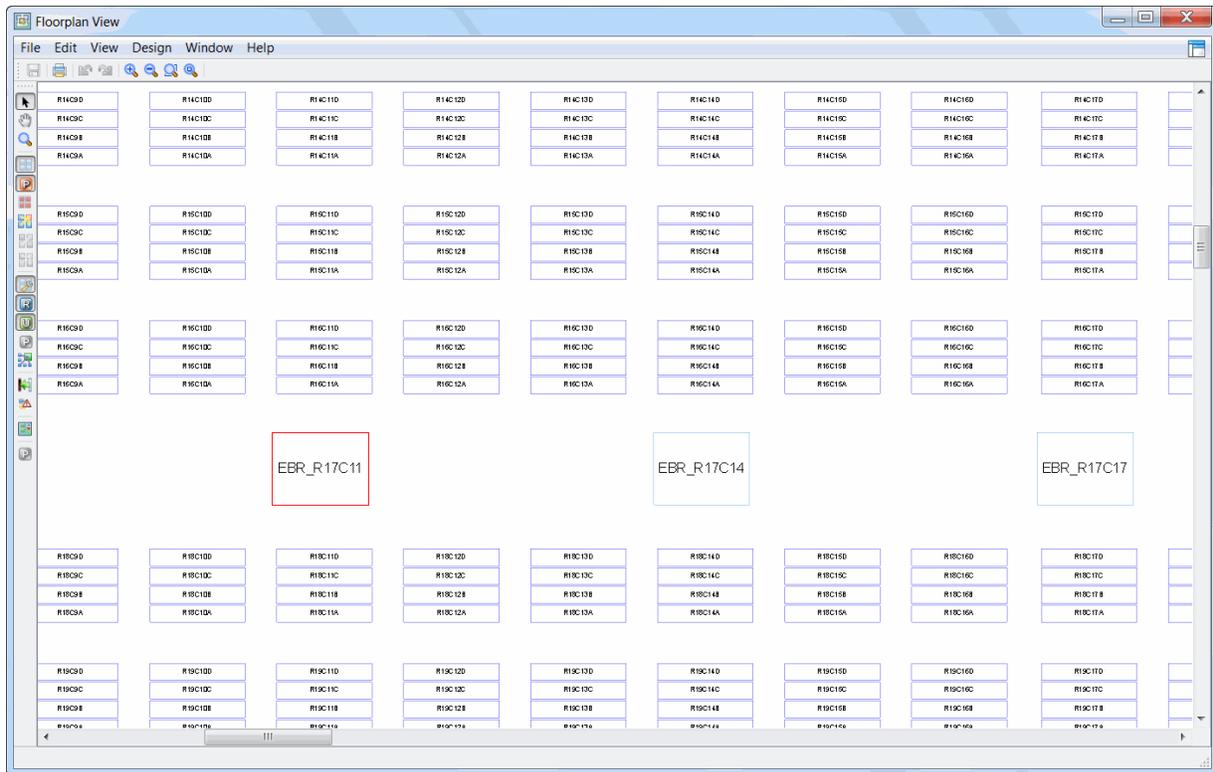


7. Right-click **EBR_R17C11** and choose **Show in > Floorplan View**, as shown in Figure 14.

Figure 14: Device View with Cross Probing

Floorplan View, shown in Figure 15, provides a large-component layout of your design. It displays user constraints from the logical preference file (.lpf) and placement and routing information.

Figure 15: Floorplan View



8. Close the Floorplan View and the Device View.

Task 6: Run Synthesis Process

Synthesis is the process of translating a register-transfer-level design into a process-specific, gate-level netlist that is optimized for Lattice Semiconductor FPGAs. Diamond can be used with almost any synthesis tool. Diamond comes with two tools fully integrated: Synopsys Synplify Pro for Lattice and Lattice Synthesis Engine (LSE). “Fully integrated” means that you can set options and run synthesis entirely from within Diamond.

You will be using Synopsys Synplify Pro for Lattice to synthesize your design for the LatticeECP3 FPGA. If you are designing for MachXO, MachXO2, or Platform Manager, you can use Lattice Synthesis Engine or another third-party synthesis tool instead of Synplify Pro for Lattice. To change the synthesis tool, from the Diamond main window, choose **Project > Active Implementation > Select Synthesis Tool**.

To synthesize the design and examine resource utilization:

1. From the Process View, double-click **Synthesize Design**.

When finished, check the icon next to Synthesize Design in the Process frame. A green check mark 🟢 indicates success; a yellow triangle ⚠️ indicates success with warnings; a red X ❌ indicates failure.

- When the synthesis process is complete, choose **View > Show Views > Hierarchy**. After you've synthesized your design and loaded this view for the first time, its name changes to Hierarchy – Post Synthesis Resources.
- Select the **Hierarchy – Post Synthesis Resources** tab, as shown in Figure 16.

Figure 16: Post Synthesis Resources

Unit	File	LUT4	PFU Registers	IO Registers	Carry Cells	Instantiated
topcount	.../LEDtest/source/topcount.v	83(8)	85(1)	27(27)	30(0)	51(37)
my_pll_uniq_1(my_pll_inst)	...diamond_tutorial/my_pll.v	0(0)	0(0)	0(0)	0(0)	2(2)
LEDtest_uniq_1(my_LEDtest)	.../LEDtest/source/LEDtest.v	15(15)	0(0)	0(0)	0(0)	1(1)
count8_uniq_0(counter1)	...LEDtest/source/count8.vhd	0(0)	16(16)	0(0)	5(5)	2(2)
count4_uniq_2(counter3)	...ial/LEDtest/source/count4.v	1(1)	12(12)	0(0)	3(3)	2(2)
count4_uniq_1(counter2)	...ial/LEDtest/source/count4.v	4(4)	8(8)	0(0)	0(0)	3(3)
clockDivider_uniq_1(clockDivider_inst)	...Dtest/source/clockDividerv	55(55)	48(48)	0(0)	22(22)	4(4)

The Post-Synthesis Hierarchy View displays the number of logical resources within each level of the design.

In the Hierarchy table shown in Figure 16, topcount is the top module displaying the resource utilization.

- ▶ LUT4 83(8) – 83 represents the total LUT4 count utilization throughout the design and 8 represents the LUT4 utilized only in the design module topcount.
- ▶ PFU Registers 85(1) – 85 represents the total PFU register utilization throughout the design and 1 represents the PFU registers utilized only in the design module topcount. Similar utilization is shown for the I/O registers, carry cells and SLICES.
- ▶ my_pll_uniq_1, LEDtest_uniq_1, count8_uniq_0, count4_uniq_2, count4_uniq_1 and clockDivider_uniq_1 are the sub-modules (instances) of the design. For example, the sub-module count4_uniq_1, LUT4 4(4) represents the total LUT4 count in the sub-module count4_uniq_1.
- ▶ PFU Registers 8(8) represents the total PFU Registers in the sub-module count4_uniq_1.
- ▶ SLICE represents the total number of SLICES in the sub-module.

Hence, the total number of logic resources (adding the resources from the individual module) is reflected in the top level module topcount.

Task 7: Set Timing and Location Assignments

Timing and location assignments constrain logic synthesis, as well as back-end map, place, and route programs to help meet your design requirements. A well-constrained design helps optimization algorithms work as efficiently as possible. In this section you'll set default timing constraints for the operating frequency and I/O timing then assign package pins to specific I/O signals.

To set timing and location assignments:

1. From the Process view, double-click **Translate Design** and then **Map Design**.

The batch interface to logic synthesis, EDIF translation, and the design mapper run. Report files appear in the Reports view. To view each process report, select the process in the **Design Summary** pane.

Each major stage of an FPGA implementation is illustrated as a milestone in the Process view: Synthesize Design, Translate Design, Map Design, Place & Route Design, and Export Files. The status of any stage is represented by the following color-coded icons:

- ▶ Completed (Green check mark) - The stage finished successfully and produced output.
- ▶ Warning (Yellow Exclamation mark) - The stage finished with warning messages generated. You can go to the Warning panel to view the warning messages.
- ▶ Error (Red cross mark) - The stage failed. You can go to the Error panel to view the error messages.

2. From the **Design Summary** pane of the Reports view, select **Process Reports > Map**.

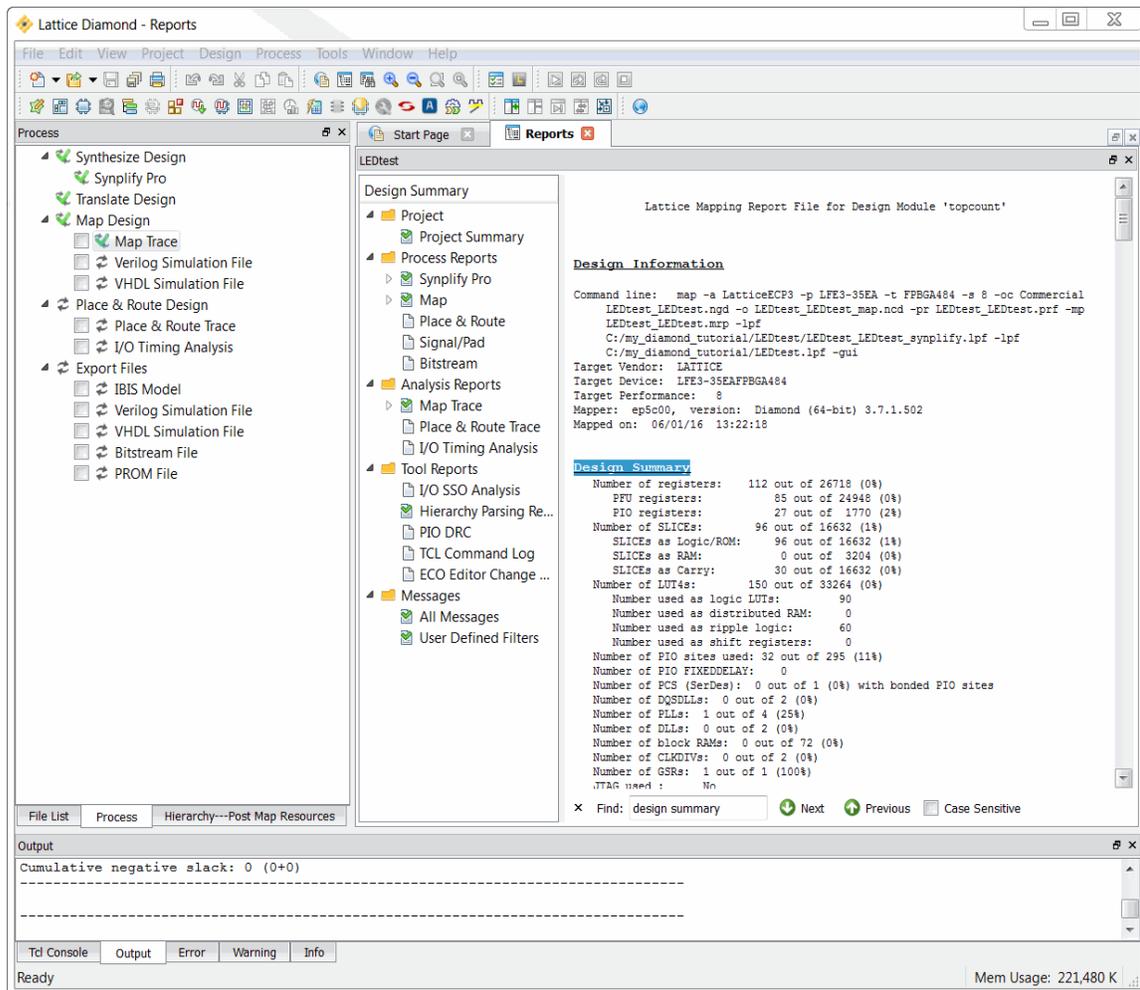
The Map Report, as shown in Figure 17, appears in the right panel.

3. Right-click in the right pane of the Reports view.
4. Choose **Find in Text**.
5. Type in **Design Summary**.

The report highlights the Design Summary section of the report.

In the Design Summary pane, there is the report icon . If a report has been generated, the icon appears as . If the report is not the most recent version, the icon appears as . To view the contents of the entire report, click on the report to be viewed. The entire report is then displayed in the right pane of the Reports view. Use the scroll bar to navigate through the report. Some of the reports are divided into sections (for example, Map, Place & Route, and Signal/Pad). Click the plus  sign before the report to display the sections in a list. Choose the desired section. The whole report will be displayed with the selected section displayed at the top of the right pane of the Reports view.

Figure 17: MAP Report



- Choose **Tools > Spreadsheet View**, or click .

The Spreadsheet View appears. The Spreadsheet View is one of several preference editors available to you to define timing, I/O and floorplan constraints for the place and route tools. Preferences are organized by type into separate tabs of the Spreadsheet View.

- Click the **Detach Tool** icon  at the upper right corner of the Spreadsheet View.

The Spreadsheet View is detached from the Diamond main window.

- Click the **PERIOD/FREQUENCY Preference** icon  on the Spreadsheet View tool bar.

The PERIOD/FREQUENCY Preference dialog box appears.

- Enter the following preference settings:

Type: **FREQUENCY**
 Second Type: **Net**
 Available Clock Nets: **clk_c**

Frequency: **100MHz**

10. Click **OK**.

The Timing Preferences tab of the Spreadsheet View appears with the new FREQUENCY preference defined.

11. Click the **INPUT_SETUP/CLOCK_TO_OUT Preference** button  on the Spreadsheet View toolbar.

The INPUT_SETUP/CLOCK_TO_OUT Preference dialog box appears.

12. Enter the following preference settings:

Type: **INPUT_SETUP**

Second Type: **All Ports**

Clock Ports/Nets: **clk**

Time: **10ns**

Hold Time: **1ns**

13. Click **OK**.

The Timing Preferences tab of the Spreadsheet View appears with the new INPUT_SETUP preference defined. You can define preferences in the relevant preference dialog box.

14. From the Timing Preferences tab, right-click the INPUT_SETUP entry, and select **New INPUT_SETUP**.

The INPUT_SETUP/CLOCK_TO_OUTPUT Preference dialog box appears.

15. Enter the following settings:

Type: **INPUT_SETUP**

Second Type: **Individual Ports**

Available Input Ports: **reset**

Clock Ports/Nets; **clk**

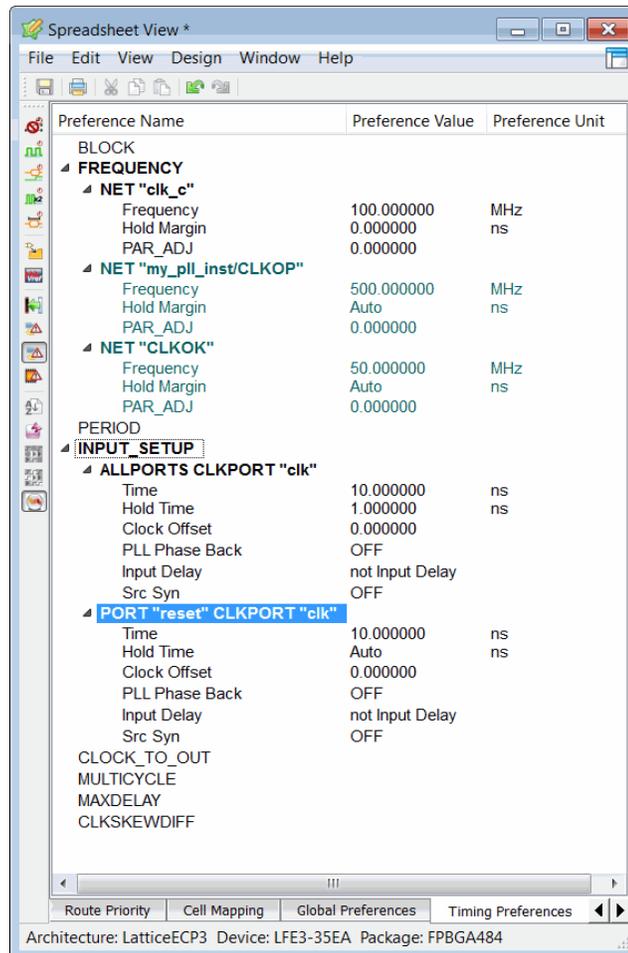
Time: **10ns**

16. Click **OK**.

The Timing Preferences tab of the Spreadsheet View appears, as shown in Figure 18, with the new INPUT_SETUP preference defined.

The preference dialog box can be invoked from the toolbar icon, the menu item (Edit > Preferences from the Spreadsheet View), or from the right-click menu of the Spreadsheet View. You can also double click on a value in Timing Preferences tab and edit the value directly.

Figure 18: Spreadsheet View



17. Select the Port Assignments tab from the Spreadsheet View.

18. Right-click the cell in Row 1, Column IO_Type. The Name of this row is All Ports.

A pull-down menu of signal standards appears. Select **LVC MOS33**, if it is not already selected. The port attributes display is updated with the new IO_TYPE. Cell entries in the Spreadsheet View are color-coded to indicate the source of a preference setting:

- ▶ Black - User-defined setting.
- ▶ Blue - Default.
- ▶ Orange - Implied by another user-defined setting.

19. Choose **File > Save pin_assignments.lpf** from the detached Spreadsheet View.

The project Logical Preference File (.lpf) is updated. Close the Spreadsheet View.

20. From the File List view of the Diamond main window, LPF Constraint Files folder, double-click the pin_assignments.lpf file.

The Source Editor appears with the ASCII LPF file. Note the timing and location preferences defined so far. Close the Source Editor.

Task 8: Running Place and Route

Use the Process view to run the Translate Design, Map Design, and Place & Route Design process stages.

To run place and route:

1. From the Process List double-click **Place & Route Design**.

The place and route tools run. Intermediate results appear in the Output frame of the Diamond main window.

2. From the Design Summary pane of the Reports view, find the Process Reports section. You will find a green check mark appears before the reports generated successfully. Expand the **Process Reports** section. Select **Place & Route**.

Details about Place & Route appear in the right pane of the Reports view.

3. From the Process List double-click **Place & Route Trace**.

The TRACE timing analyzer runs.

4. From the Design Summary pane of the Reports view, expand **Analysis Reports**, and then select **Place & Route Trace**.

The Place & Route Trace Report appears in the right pane of the Reports view.

5. From the Process List double-click **I/O Timing Analysis**.

The timing analysis runs.

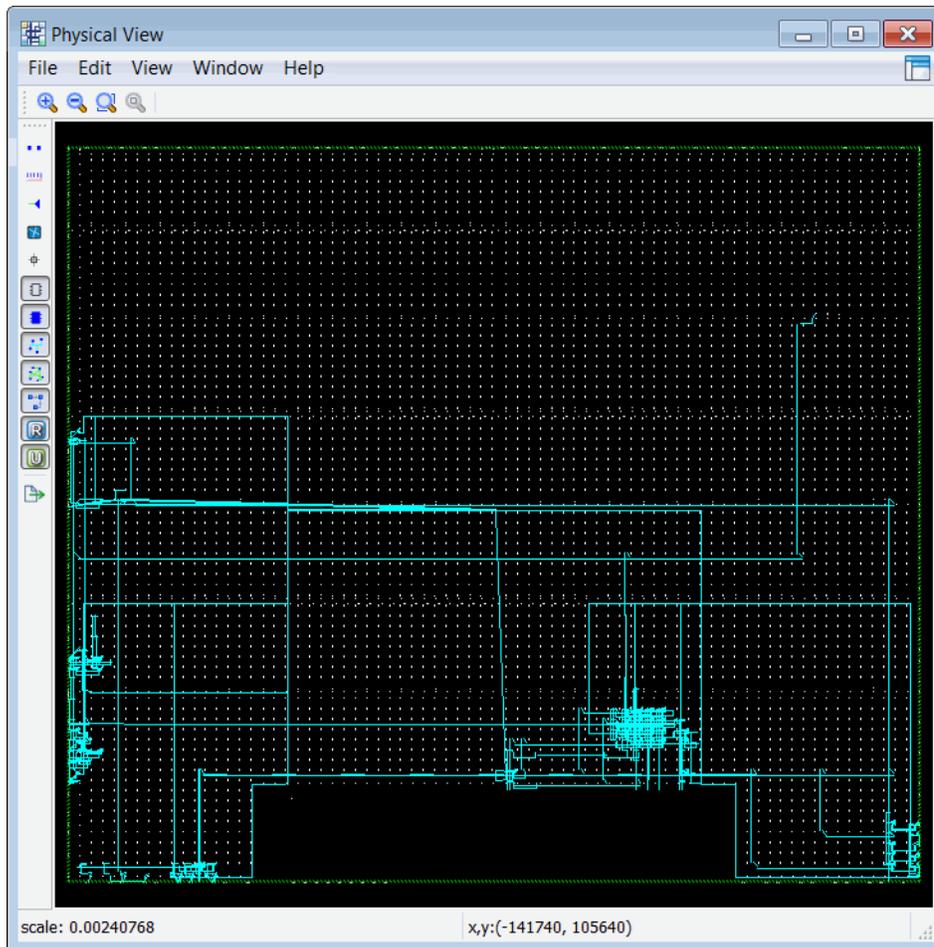
6. From the Design Summary pane of the Reports view, select the **I/O Timing Analysis** section of Analysis Reports.

The I/O Timing Report appears in the right pane of the Reports view.

7. Choose **Tools > Physical View**.

The Physical View appears, shown in Figure 19. Physical View provides a read-only detailed layout of your design that includes switch boxes and physical wire connections.

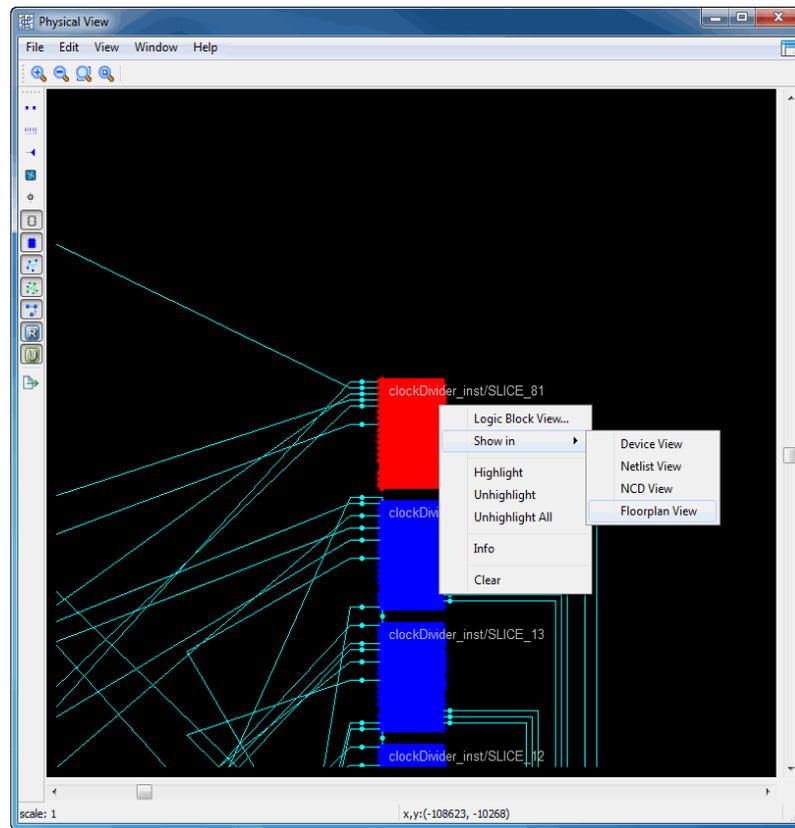
Figure 19: Physical View



8. To zoom into a component:
 - a. Magnify the surrounding area by clicking and dragging a box around it from left to right.
 - b. Click the component.
 - c. Click the  button.

Right click on the component and choose **Show in > Floorplan View**, as shown in Figure 20, to display the Floorplan View.

Figure 20: Physical View

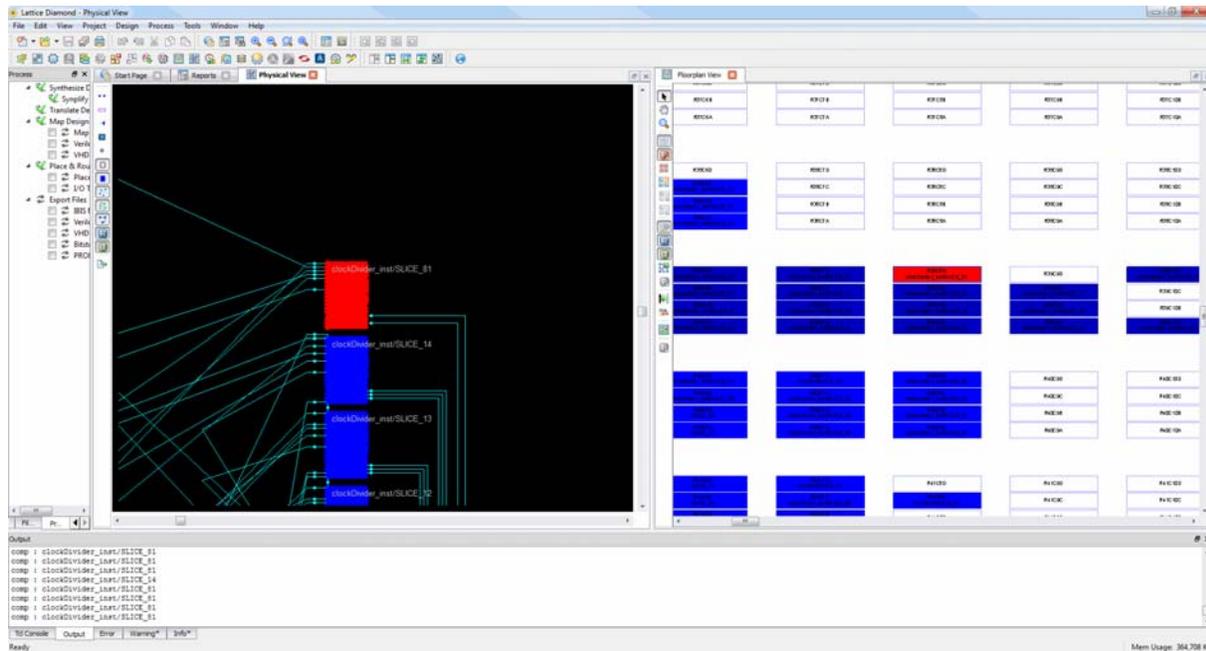


9. To auto cross-probe between Floorplan and Physical Views, ensure both views are attached to the Diamond main window and then right-click on the Floorplan View tab and select **Split Tab Group**.

The two views display in parallel, as shown in Figure 21.

When both Floorplan View and Physical View are open, an item that you select in one of these views is automatically selected in the other. Auto cross-probing is especially useful for immediately examining connections in both views.

Figure 21: Cross Probing



10. Right click on the Floorplan View tab and choose **Move to Another Tab Group**.

Now both tabs are merged into a single group as before.

11. Close Floorplan View and Physical View tabs.

Task 9: Examine Post Place and Route Results

Static Timing Analysis (STA) can determine if your circuit design meets timing constraints. Rather than simulation, it employs conservative modeling of gate and interconnect delays that reflect different ranges of operating conditions on various dies, providing complete verification coverage.

In this task, you will view the results of the Static Timing Analysis and then use the Timing Analysis view to enter clock jitter values.

To examine timing analysis results:

1. Choose **Tools > Timing Analysis View**, or click  in the Diamond toolbar.

The Timing Analysis view appears.

2. Click the **Detach Tool** icon in the upper right corner of the Timing Analysis view.

The Timing Analysis view is detached from the Diamond main window.

A summary of the post-route Static Timing Analysis settings such as target device information, preference file, performance grade, and environment conditions appear in the upper left pane. The lower left pane provides an

index of the available analysis results. Related timing preferences appear in each analysis section.

- From the Preference Reports tab (on the lower left of the Timing Analysis view), select **INPUT_SETUP ALLPORTS 10ns HOLD 1ns CLKPORT "clk" setup**.

The Path Table in the upper right of the Timing Analysis view is populated with Source, Destination, Weighted Slack, Arrival, Required, Data Delay, Route%, Levels, and other details.

- Select Row 1 of the Path Table.

The Detailed Path Tables in the lower pane are populated with details.

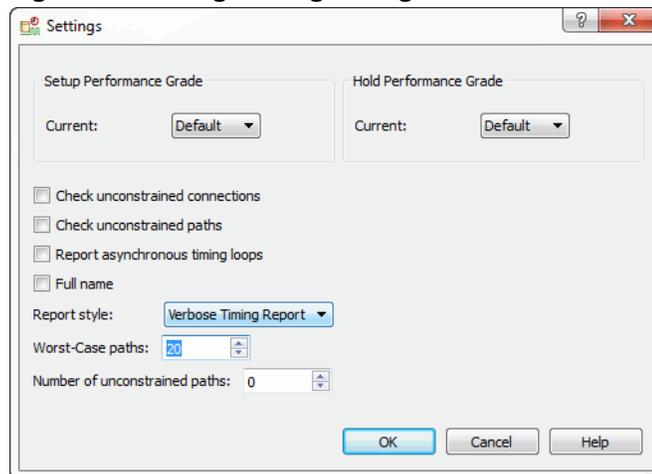
- Choose **Edit > Settings**, or click  from the toolbar in the Timing Analysis view.

The Settings dialog box appears, as shown in Figure 22.

- Enter **20** into the Worst-Case Paths field and click **OK**.

The Timing Analysis view is refreshed with the additional path data.

Figure 22: Timing Setting Dialog Box



- After selecting a Preference Name from the Preference Reports tab, you can use the Source Filter field of the Path Table to filter out all the wanted paths. Delete the text from the Source Filter field to display the full Source list again.

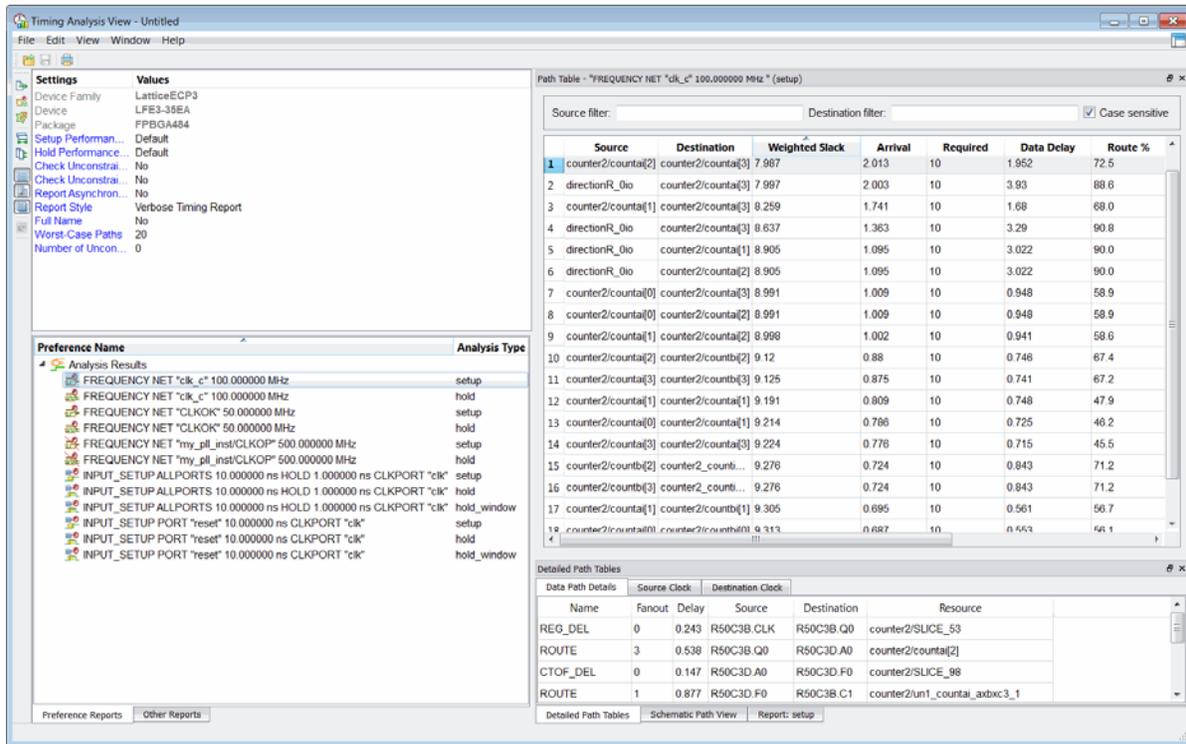
- Select the first row in the **Path Table**.

The Detailed Path Tables are updated.

- Select the **Detailed Path Tables** tab, then select the **Data Path Details** tab.

Each component of the data path delay is identified, alternating between route delays and combinatorial or clock-to-output type delays, as shown in Figure 23.

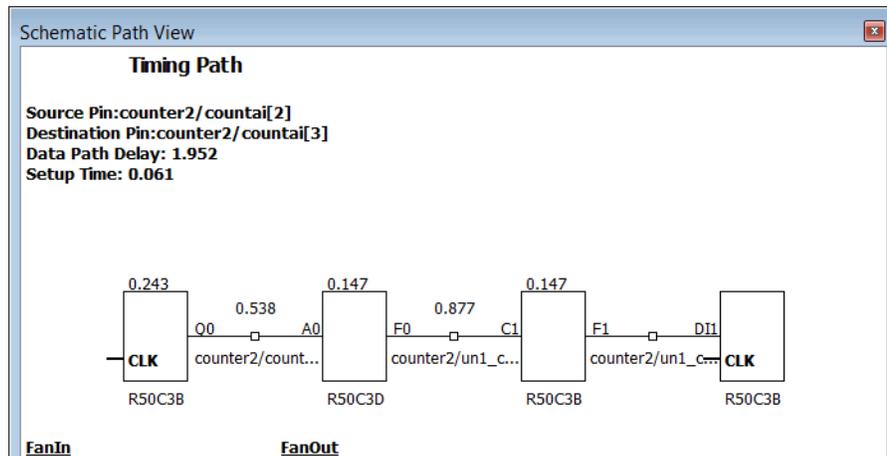
Figure 23: Timing Analysis



10. Select the **Schematic Path View** tab.

A schematic graphic of the data path timing path appears, as shown in Figure 24.

Figure 24: Schematic Path View



Diamond allows users to enter peak-to-peak clock jitter on an input CLOCK PORT. The jitter is propagated through to various design modules that use this clock. The trace will use half of p-p jitter in the direction that will cause the total timing slack to reduce.

11. To enter clock jitter values, click the **Change Timing Preferences** button in the Timing Analysis view.

The TPF Spreadsheet View appears.

12. Click the **PERIOD/FREQUENCY Preference**  button in the TPF Spreadsheet View.

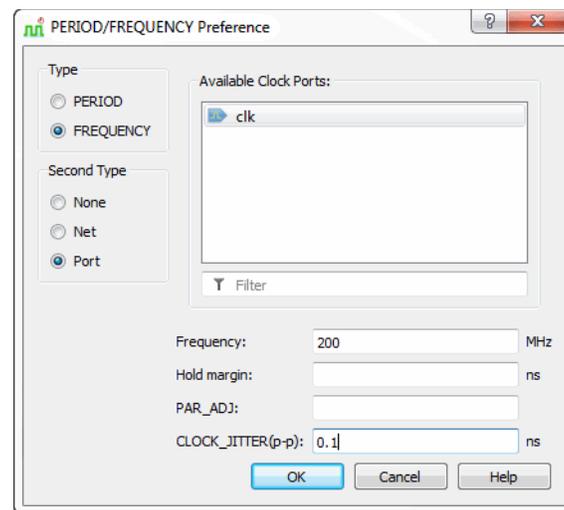
The Period/Frequency Preference dialog box appears, shown in Figure 25.

13. Enter the following preference settings:

Type:	Frequency
Second Type:	Port
Available Clock Ports:	clk
Frequency:	200MHz
Hold margin:	0.0ns
CLOCK_JITTER(P-P):	0.1ns

14. Click **OK**.

Figure 25: Period/Frequency Preference



15. In Timing Analysis view, click on the **Update**  button.

After selecting a Preference Name from the Preference Reports tab, the resulting clock jitter values can be viewed in the **Data Path Details** tab, within the **Detailed Path Tables** tab on the right.

16. Close the Timing Analysis view and click **No** to discard the changes.

Diamond allows the user to specify the peak-to-peak system jitter for timing analysis. When not used, a default value of 0 is used for all analysis. System jitter affects all clocks in the design.

17. To enter system jitter values, choose **Tools > Spreadsheet View**. Select the **Global Preferences** tab.

18. In the Global Preferences tab, double-click the **Preference Value** for SYSTEM_JITTER(ns), shown in Figure 26. Enter **0.1** into the dialog box.

By default, the timing analysis engine, TRACE, uses those timing constraints applied by timing-driven map, place, and route. However, you can modify timing preferences to manage the timing objectives of the implementation tools independent of Static Timing Analysis. To accommodate an experimental Static Timing Analysis loop, the TPF Spreadsheet View allows you to edit the timing preferences for use with the Timing Analysis view. This allows you to establish modified or additional timing preferences independent of the constraint set used for MPAR.

To tighten the timing objective of a preference and examine the results:

1. Choose **Tools > Timing Analysis View**, or click  in the Diamond toolbar.

The Timing Analysis view appears.

2. From the Preference Name list on the lower left of Timing Analysis View, select **INPUT_SETUP ALLPORTS 10ns HOLD 1ns CLKPORT "clk" Setup**, right-click and choose **TPF Preferences**.

Spreadsheet View – TPF appears, as shown in Figure 27.

3. In the Timing Preferences tab of Spreadsheet View – TPF, right-click 10.00000ns in the Preference Value column for the ALLPORTS CLKPORT "clk" and choose **Edit Value**.
4. Enter **15ns** into the Preference Value field.
5. Press **Enter**.

Figure 27: Spreadsheet View

Preference Name	Preference Value	Preference Unit
BLOCK		
▲ FREQUENCY		
▲ NET "CLKOK"		
Frequency	50.000000	MHz
Hold Margin	0.000000	ns
PAR_ADJ	0.000000	
▲ NET "my_pll_inst/CLKOP"		
Frequency	500.000000	MHz
Hold Margin	0.000000	ns
PAR_ADJ	0.000000	
▲ NET "clk_c"		
Frequency	100.000000	MHz
Hold Margin	0.000000	ns
PAR_ADJ	0.000000	
PERIOD		
▲ INPUT_SETUP		
▲ ALLPORTS CLKPORT "clk"		
Time	15.000000	ns
Hold Time	1.000000	ns
Clock Offset	0.000000	
PLL Phase Back	OFF	
Input Delay	not Input Delay	
Src Syn	OFF	
▲ PORT "reset" CLKPORT "clk"		
Time	10.000000	ns
Hold Time	Auto	ns
Clock Offset	0.000000	
PLL Phase Back	OFF	
Input Delay	not Input Delay	
Src Syn	OFF	
CLOCK_TO_OUT		
MULTIPLY		

- After a few moments, return to Timing Analysis View.

The Update button  on the toolbar is now rotating.

- Click the **Update** button.

After a short while, the indicator stops rotating and the new analysis results become available in Timing Analysis View. In the title bar of Timing Analysis View, "Untitled" appears with an asterisk, which indicates an in-memory change to the timing preferences. You can save the change to a Timing Preference File (.tpf) by choosing File > Save Untitled As, giving it a name and location, and clicking Save. The .tpf file will then appear in the Analysis Files folder of the File List pane. These .tpf files enable you to experiment with different timing settings without affecting the .lpf source file. For more information, see Analyzing Static Timing > Using Timing Analysis View in the Diamond online Help.

- Close the Timing Analysis View. In the Save dialog box, click **No** to discard the change. Spreadsheet View – TPF will close automatically.

Task 11: Comparing Multiple Place and Route Runs

Use Run Manager to run multiple synthesis and place and route passes, compare the timing score results, and load the native circuit description (NCD) database of the best run into the workspace for further analysis.

You can create multiple strategies or implementations for the design, then compare the runs with different implementation and strategy combination. One implementation can only be bound with one active strategy.

To create a new implementation:

1. Choose **File > New > Implementation** from the Diamond main window, or right-click on the project name icon  from the File List view and choose **Add > New Implementation**.
2. In the New Implementation dialog box, type **verilog_vhdl** in the Name text box.

By default, the directory and location will be the same name as the implementation name. You can change the directory or location to a desired one.
3. Choose **Strategy1** from the Default Strategy drop-down menu.
4. Click **Add Source** and choose **From Existing Implementation > LEDtest**.
5. Select testbench.v and click the **Remove Source** button.
6. Select the **"Copy source to implementation directory"** option and click **OK**.

The new implementation **verilog_vhdl** is now displayed in the File List pane.

Note

If you want to make this new implementation active, right-click verilog_vhdl and choose Set as Active Implementation. You can have multiple implementations in your project, but you can make only one implementation active in your project at one time.

Now you will compare the run results of the LEDtest and verilog_vhdl implementations.

1. From the Diamond main window, choose **Tools > Run Manager**.

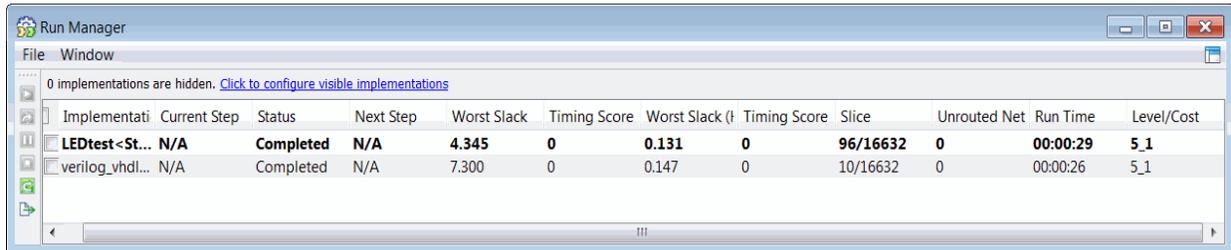
Run Manager displays a table of implementation<strategy>:
LEDtest<Strategy1> and verilog_vhdl<Strategy1>.
2. Enable **LEDtest<Strategy1>** and **verilog_vhdl<Strategy1>** by setting the check boxes for each implementation, as shown in Figure 28.
3. Click the **Rerun**  button on the Run Manager toolbar.

If Run Manager is running for the first time, it informs you that some Diamond functionality will be disabled during the run. If you don't want to see this warning again, check **Do not prompt this message in the future**. Click **OK** to proceed with the run.

If there are any active instances of certain Tool Views (such as the Source Editor), Run Manager needs to close them before it can run. Click **OK** to close those Tool Views and proceed with the run.

The two implementations start to run simultaneously.

Figure 28: Run Manager



0 implementations are hidden. [Click to configure visible implementations](#)

Implementati	Current Step	Status	Next Step	Worst Slack	Timing Score	Worst Slack (t	Timing Score	Slice	Unrouted Net	Run Time	Level/Cost
LEDtest<St...	N/A	Completed	N/A	4.345	0	0.131	0	96/16632	0	00:00:29	5_1
verilog_vhdl...	N/A	Completed	N/A	7.300	0	0.147	0	10/16632	0	00:00:26	5_1

In a few minutes, the results of the run appear in the table. Statistics such as Start time, Run time, Score, Unrouted, Level/Cost, and Description appear. The row in bold font indicates the active implementation that is loaded. The table provides a quick review of the quality of results produced by a particular strategy. To closely examine a particular run with analysis tools, such Timing Analysis View or Power Calculator, you can set the active strategy to be loaded.

If your system provides a multiple-core processor, you can set more implementations to be run concurrently. Choose Tools > Options in the Diamond main window. Select Environment > General tab, enter a number in the box in front of the Maximum number of implementation processes in run manager option and the same number in the box in front of the Maximum number of implementation processes in run manager option. The default value is 2. The maximum allowed value is 32. For best performance, set these values to match the number of cores in your system's CPU, then click OK.

4. Choose **View > Reports**.

In the Reports view, you can view results related to the run of the current active implementation. The report for LEDtest appears in the Reports view.

5. Close Run Manager.

Task 12: Analyze Power Consumption

Included with the Diamond software is Power Calculator, which estimates the power dissipation for a given design. Power Calculator uses parameters such as voltage, temperature, process variations, air flow, heat sink, resource utilization, activity, and frequency to calculate the device's static and dynamic power consumption.

To analyze power consumption:

1. Choose **Tools > Power Calculator** or click the  button on the toolbar.

Power Calculator opens in Calculation mode.

Power Calculator provides two modes for reporting power consumption:

▶ Estimation Mode:

In estimation mode, Power Calculator provides estimates of power consumption based on the device resources or template that you provide. This mode enables you to estimate the power consumption for your design before the design is complete or even started.

▶ Calculation Mode:

In calculation mode, Power Calculator calculates power consumption on the basis of device resources taken from a design's .ncd file, or from an external file such as a value change dump (.vcd) file, after placement and routing. This mode is intended for accurate calculation of power consumption, because it is based on the actual device utilization.

Editing data in white cells, such as voltage, frequency, activity factor, and thermal data, does not change mode. Editing data in blue cells, such as design data, will change calculation mode to estimation mode.

2. Click the  icon in the upper right corner to detach Power Calculator from the Diamond main window, as shown in Figure 29.

Figure 29: Power Calculator

The screenshot shows the Lattice Power Calculator software interface. The main window is titled "Power Calculator - Untitled" and has a menu bar with "File", "Edit", "Window", and "Help". The software mode is set to "Calculation".

The interface is divided into several sections:

- Power Summary:** Includes tabs for "Power Matrix", "Logic Block", "Clocks", "I/O", "I/O Term", "Block RAM", "DSP", "PLL", "DLL", "DQSDLL", "SERDES", "Graph", and "Report".
- Device:** Family: LatticeECP3, Performance grade: 8, Device: LFE3-35EA, Operating conditions: Commercial, Package type: FPBGA484, Part Names: LFE3-35EA-8FN484C.
- Environment:** Thermal Profile... button, Ambient Temperature(°C): 25, Effective Theta-JA: 8.79, Junction Temperature(°C): 26.37, Maximum Safe Ambient(°C): 82.58.
- Device Power Parameters:** Process Type: Typical, Power File Revision: Final.
- Voltage/Dynamic Power Multiplier:**

	Voltage	DPM
Vcc	1.200	1.00
Vccaux	3.300	1.00
Vccj	1.200	1.00
Vccpll	3.300	1.00
Vccio 3.3	3.300	1.00
Vccio 2.5	2.500	1.00
Vccio 1.8	1.800	1.00
Vccio 1.5	1.500	1.00
Vccio 1.2	1.200	1.00
Vccib	1.200	1.00
Vccob	1.200	1.00
Vcca	1.200	1.00
- Current by Power Supply:**

	Static (A)	Dynamic (A)	Total (A)
	0.0219	0.0000	0.0219
	0.0175	0.0000	0.0175
	0.0025	0.0000	0.0025
	0.0046	0.0000	0.0046
	0.0113	0.0000	0.0113
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0079	0.0000	0.0079
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0060	0.0000	0.0060
	0.0716	0.0000	0.0716
- Power by Power Supply:**

	Static (W)	Dynamic (W)	Total (W)
	0.0262	0.0000	0.0262
	0.0577	0.0000	0.0577
	0.0030	0.0000	0.0030
	0.0150	0.0000	0.0150
	0.0372	0.0000	0.0372
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0095	0.0000	0.0095
	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000
	0.0072	0.0000	0.0072
	0.1559	0.0000	0.1559
- Power by Block (W):**

Block	Power (W)
Logic Block	0.0141
Clocks	0.0014
I/O	0.1005
Block RAM	0.0030
DSP	0.0011
PLL	0.0150
DLL	0.0000
DQSDLL	0.0000
SERDES	0.0084
Other	0.0122
Total	0.1559

- In the Device Power Parameters section select the following parameter:
Process Type: **Worst**.
- Click the **Thermal Profile** button in the Environment section.
The Power Calculator – Thermal Profile dialog box appears.
- In the Board Selection section, select the following parameter:
Small board.
- Click **OK**.

After a short while the new power analysis results become available in the Power summary tab.

In the title bar of Power Calculator, "Untitled" appears with an asterisk, which indicates an in-memory change to the timing preferences. You can save the change to a Power Calculator File (.pcf) by choosing File > Save File As and giving it a name and location. The .pcf file will then appear in the Analysis Files folder of the File List Pane. These .pcf files enable you to experiment with Power Analysis settings without affecting the .lpf source file.

7. Close Power Calculator. If you chose not to save in the previous step, a Save dialog box will now appear. Click **No** to discard the change.

Task 13: Run Export Utility Programs

Use the Process view to generate files for exporting. One of the files exported will be a bitstream file (.bit) which will be used to program a LatticeECP3 device in the next task.

1. From the Process view, choose **Export Files**.

A set of export files appear under the Export Files process.

2. Select the following Export Files:

IBIS Model

VHDL Simulation File

Bitstream File

3. Click the **Run** button  on the Diamond toolbar.

Diamond generates the selected files and saves them in your project directory.

Task 14: Download a Bitstream to an FPGA

This task requires that you have a LatticeECP3 Versa Development Kit.

In the previous section, you generated export files including a bitstream file (.bit). In this section, you will use Diamond Programmer to download a bitstream to a LatticeECP3 FPGA mounted on a LatticeECP3 Versa Development Kit board.

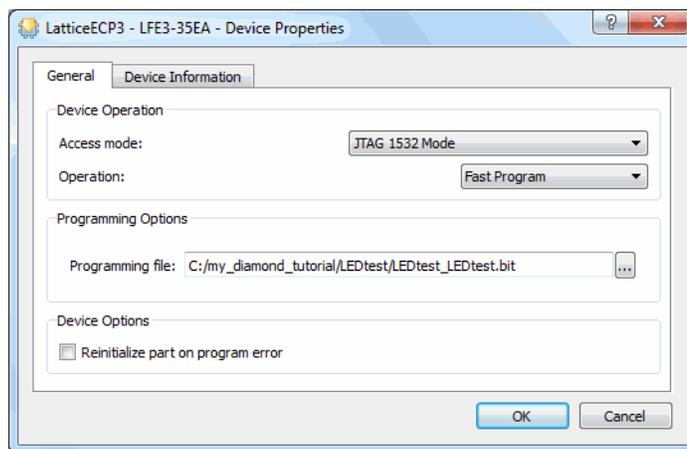
To download the bitstream to the FPGA on the board:

1. Remove any Lattice USB Programming cables from your system.
2. Connect the power supply to the development board.
3. Connect a USB cable from your computer to the LatticeECP3 Versa Development Kit board. Give the computer a few seconds to detect the USB device before moving to step 4.
4. Choose **Tools > Programmer**, or click the  icon on the toolbar.
5. In the Getting Started dialog box, select **Create a new Project from a Scan**.
 - a. In the Cable box, select **HW-USBN-2B (FTDI)**.
 - b. In the Port box, choose **FTUSB-0** from the drop-down menu.
 - c. Click **OK**.

Programmer scans the device database, and then the Programmer view displays in Diamond.

6. Ensure that the proper device is selected by clicking the cell in the Device column and selecting **LFE3-35EA** from the drop-down menu.
7. Double-click the cell labeled **Fast Program** in the Operation column to display the Device Properties dialog box, and choose the following settings:
 - ▶ For Access Mode, choose **JTAG 1532 Mode** from the pull-down menu.
 - ▶ For Operation, choose **Fast Program** from the pull-down menu.
 - ▶ Ensure that the bitstream file named LEDTest_LEDTest.bit is selected as the programming file, as shown in Figure 30.

Figure 30: Device Properties Dialog Box



8. Click **OK**.
9. On the LatticeECP3 Versa board, ensure that user switches (DIP switches) J6 and J7 are in the **OFF** position, and all of the rest of the user switches are in the **ON** position.

Note

The LatticeECP3 Versa Development Kit is set up so that, when the digital display is right-side up, each switch's upward position (indicated by the "o" label) corresponds to a setting of "1." The downward position (indicated by the "ON" label) corresponds to a setting of "0." When switch J6 is up, the counter counts up. When J6 is down, the counter counts down.

Refer to the [LatticeECP3 Versa Evaluation Board User's Guide](#) for more information about the LatticeECP3 Versa evaluation board.

10. Click the **Program** button  on the Programmer toolbar to initiate the download.
11. If the programming process succeeded, you will see a green-shaded **PASS** in the Programmer Status column. Check the Programmer output console to see if the download passed.
12. At the end of this process, the FPGA is loaded with the sample test bitstream. This bitstream allows you to test the functionality of the

LatticeECP3 Versa Development Kit board. If the design is successfully downloaded onto the LatticeECP3 device, the multi-segment LED display will display 0.

To further test the design:

- a. Push the user switch (DIP switch) J7 to the **ON** position to activate the forward counter on the LED display.
- b. Push the user switch (DIP switch) J6 to the **ON** position while also keeping switch J7 in the **ON** position to activate the reverse counter on the LED display. The general purpose LEDs will now start flashing.

13. In Diamond, choose **File > Save LEDtest.xcf**.

Task 15: Convert a File Using Deployment Tool

In Task 14, you used Diamond Programmer to download a bitstream (.bit) to a LatticeECP3 FPGA.

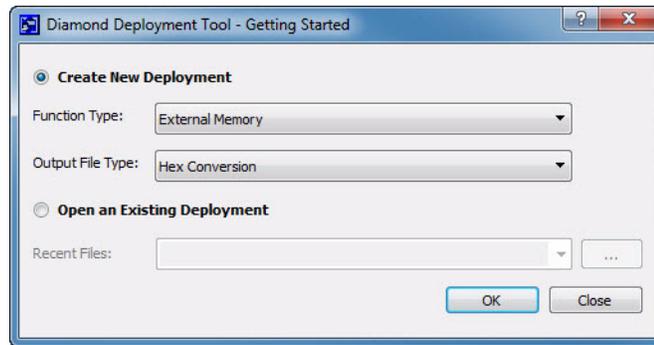
You will now use Deployment Tool to convert the .bit to an industry-standard Hex file.

Deployment Tool is a stand-alone tool found in Diamond Accessories. The Deployment Tool graphical user interface is separate from the Diamond design environment. Deployment Tool allows you to generate files for deployment for single devices and for a chain of devices. Deployment Tool can also convert data files to other formats and use the data files it produces to generate other data file formats.

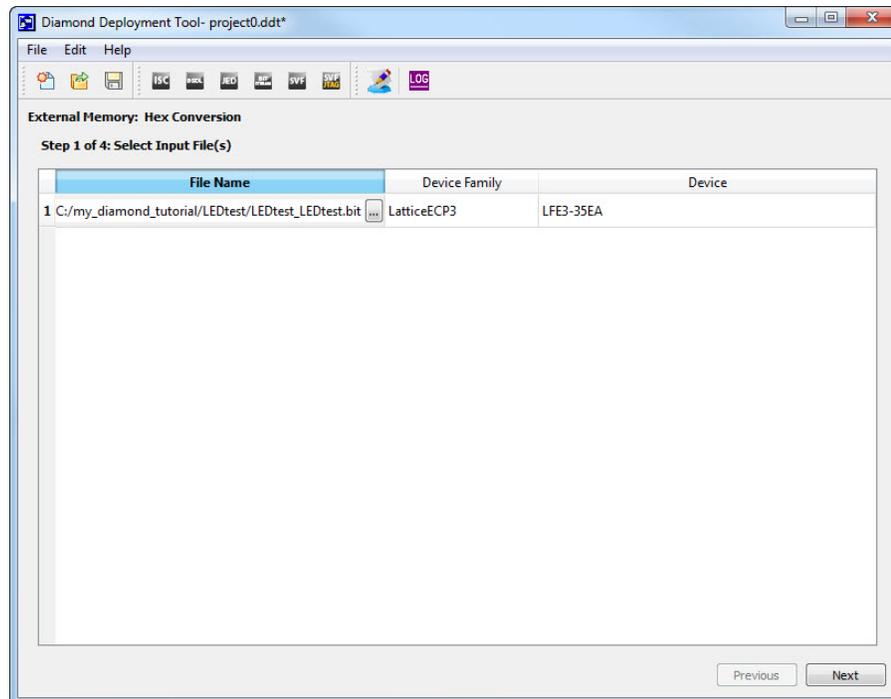
For the purpose of this tutorial, you will convert the same .bit file from Task 14 into an Intel Hex file.

To convert the .bit to an Intel Hex file using Deployment Tool:

1. Choose **Programs > Lattice Diamond <version_number> > Accessories > Deployment Tool** from the Windows Start menu.
2. In the Diamond Deployment Tool - Getting Started dialog box, choose **Create New Deployment**.
 - a. In the Function Type dropdown, choose **External Memory**.
 - b. In the Output File Type dropdown, choose **Hex Conversion**, as shown in Figure 31.

Figure 31: Deployment Tool Getting Started Dialog Box

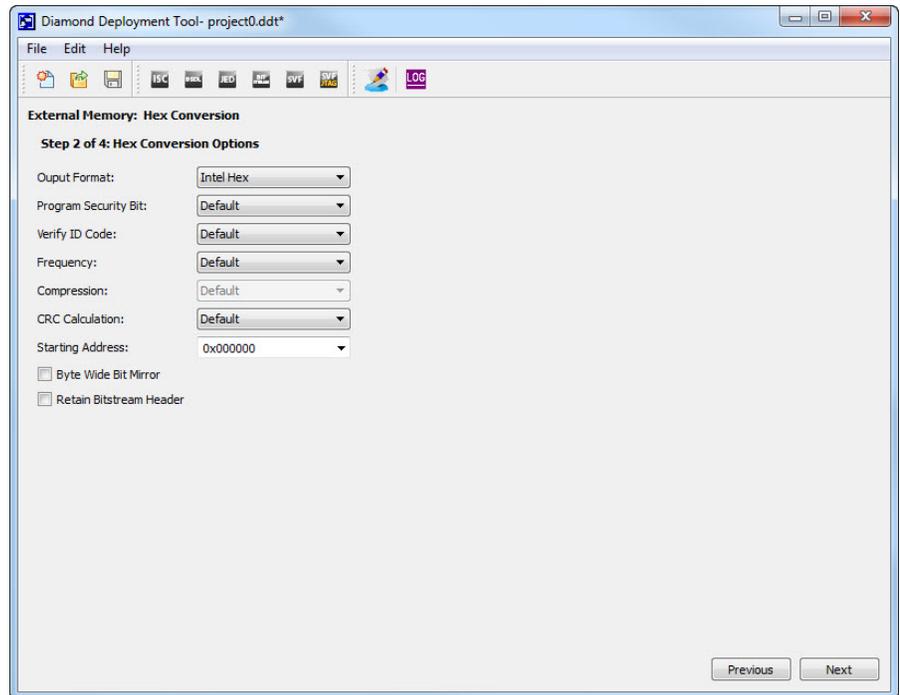
- c. Click **OK**.
3. In the Step 1 of 4: Select Input File(s) dialog box, as shown in Figure 32:
 - a. Click in the File Name box.
 - b. Click  to display the Open File dialog box.
 - c. Browse to the LEDtest_LEDtest.bit file located in the tutorial directory.
 - d. Click **Open**.
 - e. Click **Next**.

Figure 32: Deployment Tool Step 1 of 4 Dialog Box

4. In the Step 2 of 4: Hex Conversion Options dialog box, as shown in Figure 33:
 - a. Choose Output Format as **Intel Hex**.

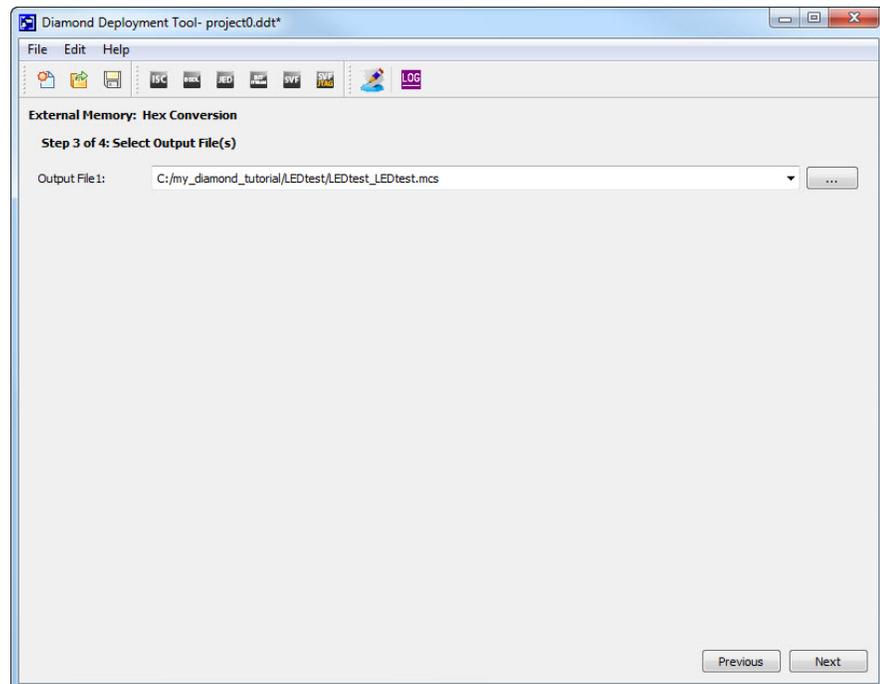
- b. Leave all other options (Program Security Bit, Verify ID Code, Frequency, Compression, and ORC Calculation) as **Default**.
- c. Leave Starting Address as **0x000000**.
- d. Leave Byte Wide Bit Mirror and Retain Bitstream Header unchecked.
- e. Click **Next**.

Figure 33: Deployment Tool Step 2 of 4 Dialog Box



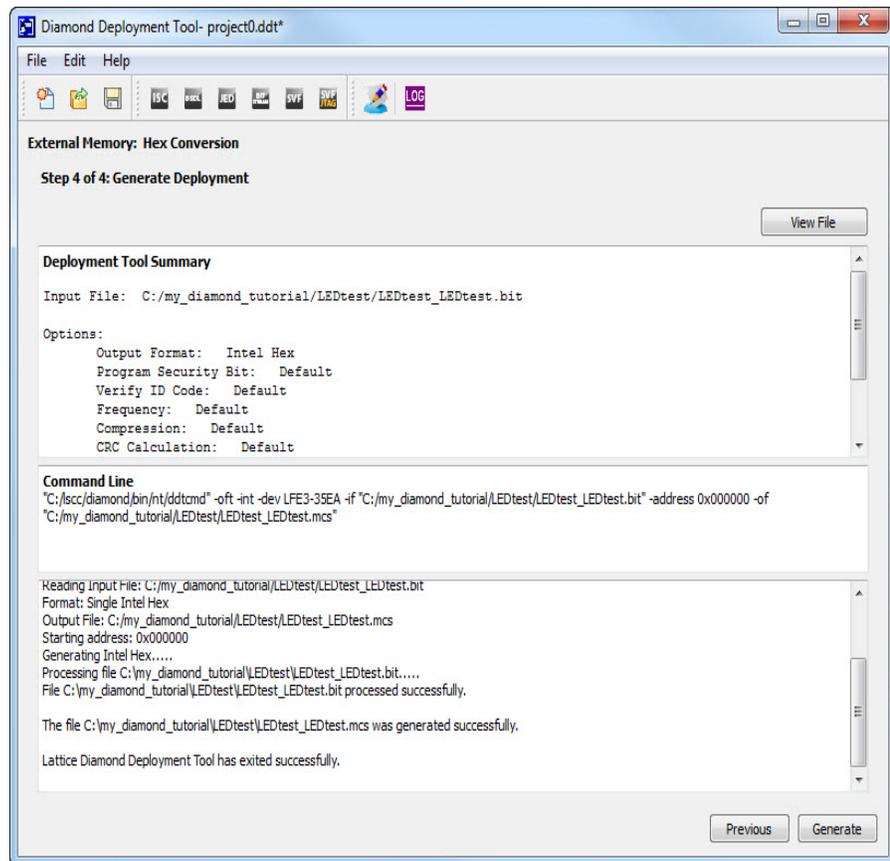
5. In the Step 3 of 4: Select Output File(s) dialog box, as shown in Figure 34:
 - a. Ensure that the Output File1 is **LEDtest_LEDtest.mcs**.
 - b. Click **Next**.

Figure 34: Deployment Tool Step 3 of 4 Dialog Box



6. In the Step 4 of 4: Generate Deployment dialog box, as shown in Figure 35:
 - a. Review the Deployment Tool Summary.
 - b. Click **Generate**.

The Hex file (LEDtest_LEDtest.mcs) is created in the tutorial directory.

Figure 35: Deployment Tool Step 4 of 4 Dialog Box

The .mcs file can be used to program the SPI Flash on the LatticeECP3 Versa Development Kit board using Programmer.

To save the Deployment Tool project:

1. Choose **File > Save**.
2. In the Save As dialog box, browse to the tutorial directory.
3. Choose a file name for the Deployment Tool (.ddt) file or use the default file name.
4. click **Save**.
5. Choose **File > Exit** to close the Deployment Tool.

Task 16: Use Reveal Inserter to Add On-chip Debug Logic

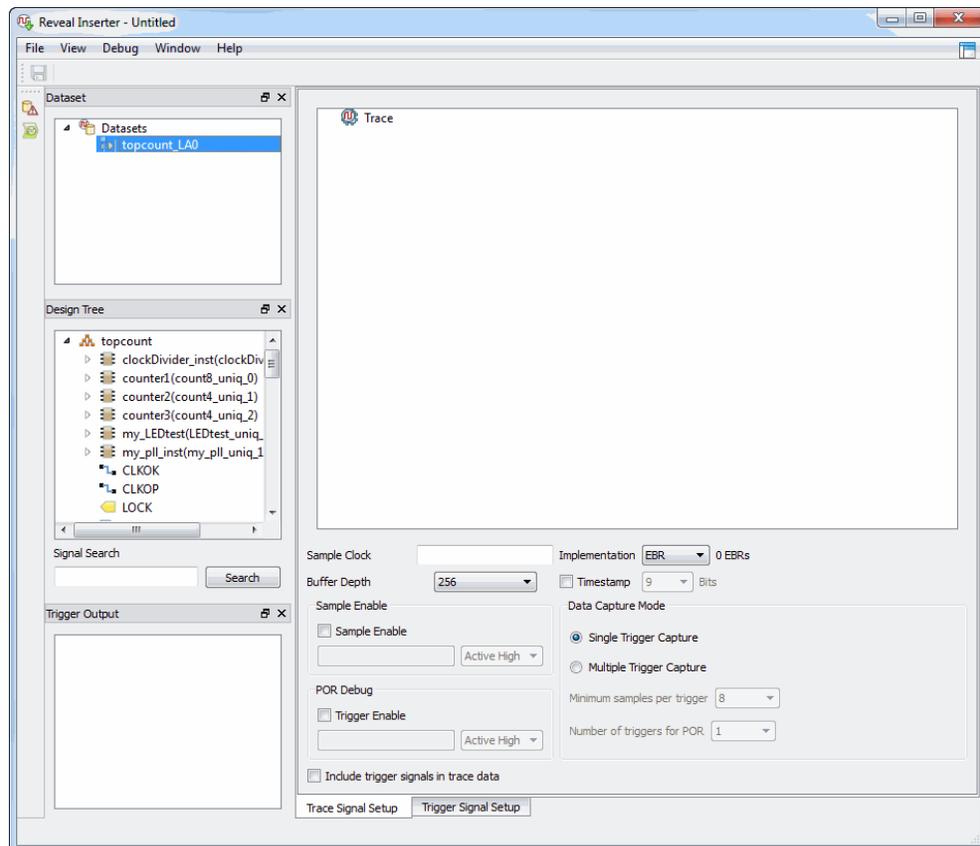
In this task, you will use Reveal Inserter to configure a Reveal core based on triggering conditions and the desired trace buffer. The primary output of Reveal Inserter is a modified version of your design with one or more cores instantiated and the core logic ready for mapping, placement, and routing.

To generate and add a Reveal core:

1. Choose **Tools > Reveal Inserter** or click the  button on the toolbar.
2. Click the  icon in the upper right corner to detach Reveal Inserter.

Reveal Inserter is detached from the Diamond main window, as shown in Figure 36.

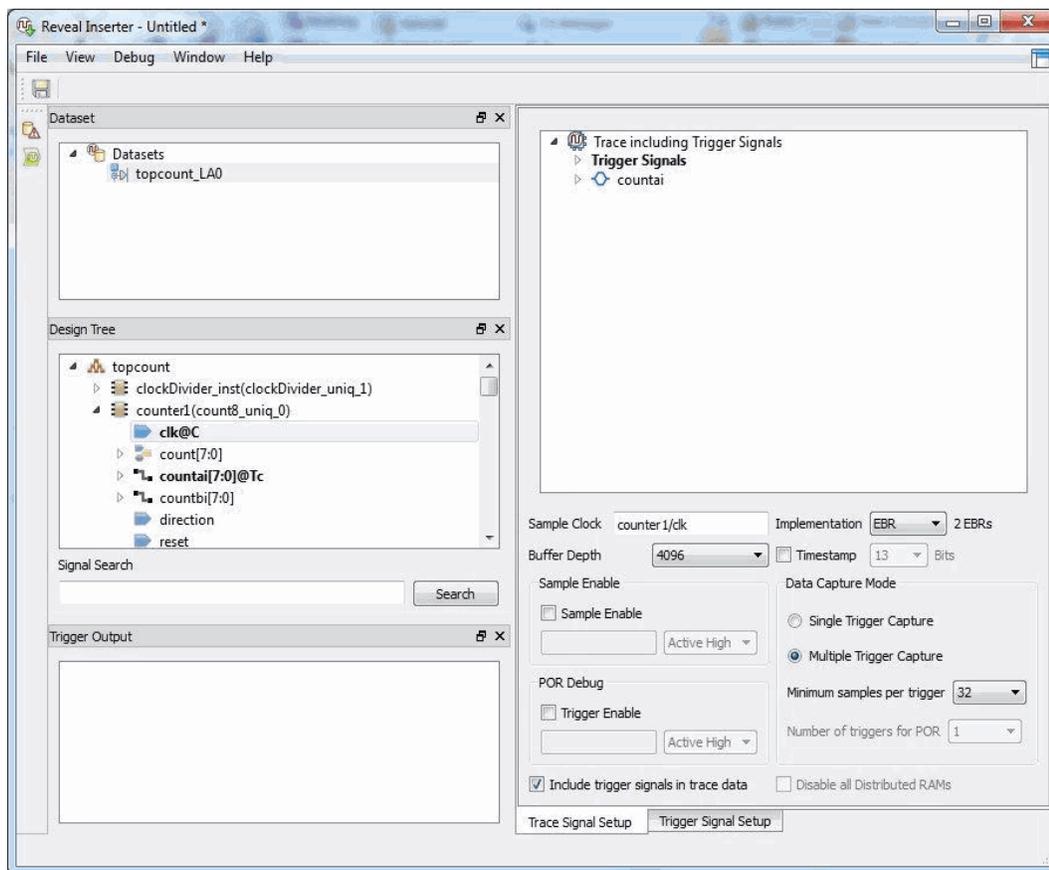
Figure 36: Reveal Inserter Main Window



3. Click on the **Trace Signal Setup** tab, if it is not already selected.
4. From the Design Tree pane, expand the **counter1(count8_uniq_0)** category and drag the **countai[7:0]** bus to the Trace Data pane on the right. Right-click the created trace bus, choose **Rename Trace Bus**, and name the bus **countai**.
5. Select the **Include Trigger Signals in Trace Data** option.
The name of the bus now appears in bold font in the Design Tree pane.
6. Drag the **clk** signal from the Design Tree pane to the Sample Clock box, or type **counter1/clk** in the Sample Clock box.
7. From the pulldown menu in the Buffer Depth box, select **4096**.
8. Set Data Capture Mode to **Multiple Trigger Capture** and Minimum Samples Per Trigger to **32**.

The Trace Signal Setup tab should now resemble Figure 37.

Figure 37: Trace Signal Setup Tap



Setting Up the Trigger Units

You will set up the trigger units in the Trigger Unit section of the Trigger Signal Setup tab.

To set up the trigger units:

1. Click on the **Trigger Signal Setup** tab.

One line appears in the Trigger Unit section of the tab with a default name of TU1.
2. Double-click the TU1 name in the Name box, backspace over "TU1," and type **countbi**.
3. Drag the **countbi[7:0]** signals from the **counter1(count8_uniq_0)** category in the Design Tree pane to the Signals (MSB:LSB) box in the Trigger Unit pane.
4. In the Operator box of the trigger unit, select **<=** from the drop-down menu.
5. In the Radix box, select **Hex** from the drop-down menu.
6. In the Value box, double-click, backspace, and type **88**.

7. Click **Add** to add a second trigger unit.
8. In the Name box, double-click TU2, backspace over "TU2," and type **dir**.
9. Drag the **directionR** signal from the Design Tree pane to the Signals (MSB:LSB) box in row 2 of the Trigger Unit pane.
10. In the Operator box, select **<=** from the drop-down menu.
11. In the Radix box, select the default of **Bin**.
12. In the Value box, double-click, backspace, and type **1**.

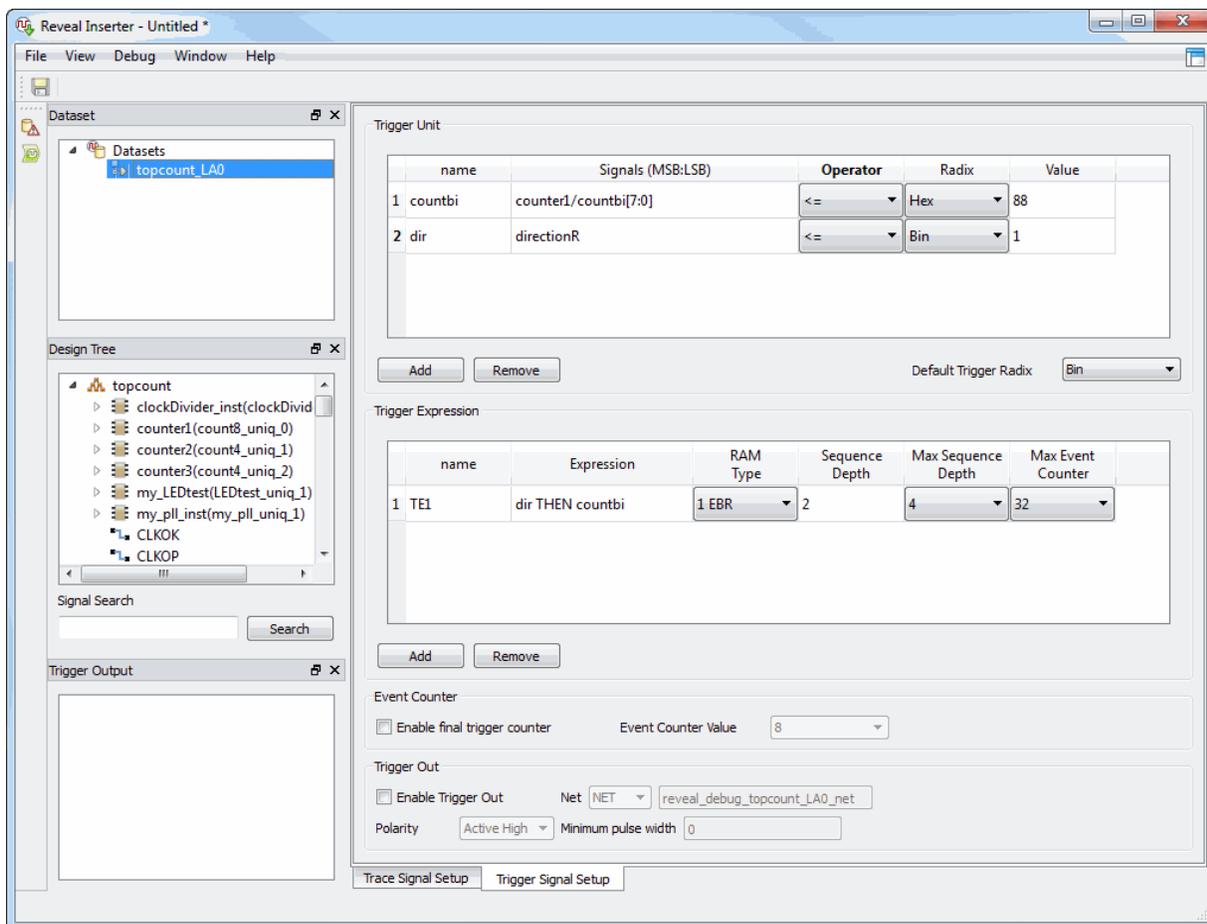
Setting Up the Trigger Expressions

Now you will set up the trigger expressions in the Trigger Expression section of the tab.

To set up the trigger expressions:

1. In the Name box in the Trigger Expressions section, use the default name of TE1.
2. In the Expression box, select the countbi and dir trigger units by typing **dir THEN countbi**.
3. In the RAM Type box, select **1 EBR** from the drop-down menu.
4. In the Sequence Depth box, make sure a value of **2** appears.
5. In the Max Sequence Depth box, select **4** from the drop-down menu.
6. In the Max Event Counter box, select **32** from the drop-down menu.
7. The Trigger Signal Setup tab should now resemble Figure 38.

Figure 38: Trigger Signal Setup Tap

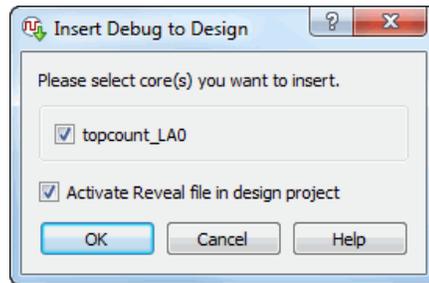


Inserting the Debug Logic

Now you will insert the debug logic into the design project.

To insert the debug logic:

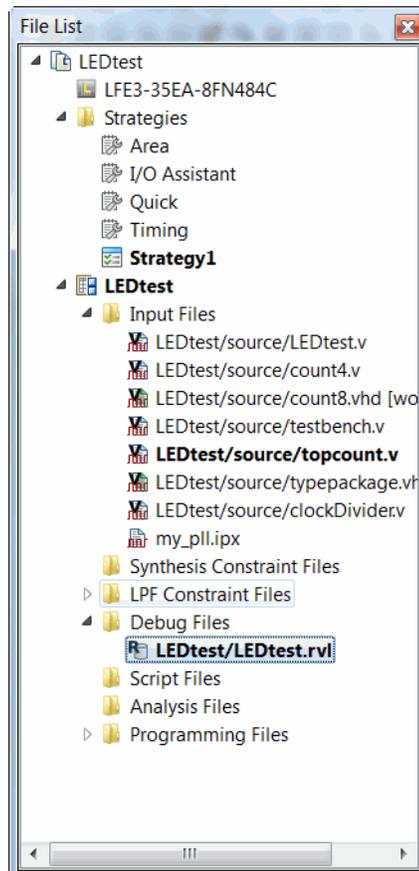
1. Choose **Debug > Insert Debug** or click .
2. In the Insert Debug to Design dialog box, shown in Figure 39, be sure that the **Activate Reveal File in Design Project** option is selected.

Figure 39: Insert Debug to Design Dialog Box

3. Click **OK**. Name the file `<project_directory>/LEDtest/LEDtest.rvl`.
4. Click **Save**.

Reveal Inserter imports the Reveal project (.rvl) file into Diamond.

The .rvl file is now added to the Debug Files in the File List, as shown in Figure 40.

Figure 40: File List with Debug File

5. In the Reveal Inserter window, choose **File > Close Window**.

Generating a Bitstream and Programming the FPGA

Use the Process view to generate files for exporting, and use Programmer to download the bitstream to the FPGA. This task assumes that the LatticeECP3 Versa board is connected to your computer with a download cable, as described in Task 14.

To generate a bitstream:

1. From the Process view, choose **Export Files**.

A set of export files appears under the Export Files process.

2. Select the following Export Files:

Bitstream File

3. Click the **Run** button  on the Diamond toolbar.

Diamond reruns all processes, generating the selected files and saving them in your project directory.

4. Choose **Tools > Programmer**, or click the  button on the toolbar.
5. In the Programmer: Getting Started dialog box, choose **Create a new Project from a Scan**. If the Getting Started dialog box does not appear, right-click the Programming File **LEDtest/LEDtest.xcf** in the File List and choose **Set as Inactive** before starting Programmer.
 - a. In the Cable box, select **HW-USBN-2B (FTDI)**.
 - b. In the Port box, choose the only setting available in the drop-down menu, **FTUSB-0**.
 - c. Click **OK**.

▶ If a New File dialog box appears, click **Yes**.

Programmer scans the device database, and then the Programmer view displays in Diamond.

6. Ensure that the device **LFE3-35EA** is selected in the Device column.
7. Double-click the cell labeled **Fast Program** in the Operation column to display the LatticeECP3 - LFE3-35EA - Device Properties dialog box and choose the following settings:
 - ▶ For Access Mode, choose **JTAG 1532 Mode** from the pull-down menu.
 - ▶ For Operation, choose **Fast Program** from the pull-down menu.
8. Ensure that the bitstream file named **LEDTest_LEDTest.bit** is selected as the programming file.
9. Click **OK**.
10. Click the Program button  on the Programmer toolbar to initiate the download.

11. If the programming process succeeded, you will see a green-shaded PASS in the Programmer Status column. Check the Programmer output console to see if the download passed.

Task 17: Use Reveal Logic Analyzer to Perform Logic Analysis

In this task, you will use Reveal Logic Analyzer to set up trigger conditions and view trace buffer data from the on-chip Reveal core operating within the device on the LatticeECP3 standard evaluation board. The trigger setup influences under what specific conditions and how the Reveal core trace signal states are displayed in Reveal Logic Analyzer's graphical user interface. You will explore just a few of the many ways to trigger and trace the system.

This task assumes that the LatticeECP3 Versa board is connected to your computer with a download cable, as described in Task 14.

Creating a New Reveal Logic Analyzer Project

You must first create a Reveal Analyzer project.

To create a new Reveal Logic Analyzer project:

1. In the Diamond main window, choose **Tools > Reveal Analyzer** or click the  button on the toolbar.

The Reveal Analyzer Startup Wizard dialog box appears, as shown in Figure 41.

2. In the upper left of the Reveal Analyzer Startup Wizard dialog box, select **Create a new File**.

3. Type **Test** in the box to name the file.

The .rva extension is added automatically.

4. In the drop-down menu on the top row, choose **HW-USBN-2B (FTDI)**, if it is not already selected.

5. Click **Detect**.

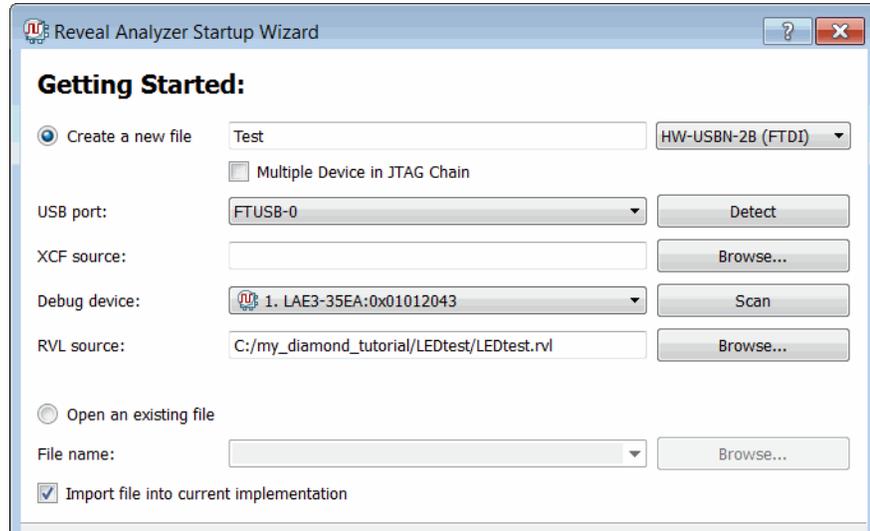
The cable connected to the PC is detected, and is listed in the USB Port box.

6. Click **Scan** to find the FPGA.

The LatticeECP3 device on the Versa board is displayed in the Debug device box.

- In the RVL Source box, browse to **<project_directory>/LEDtest/LEDtest.rvl**.

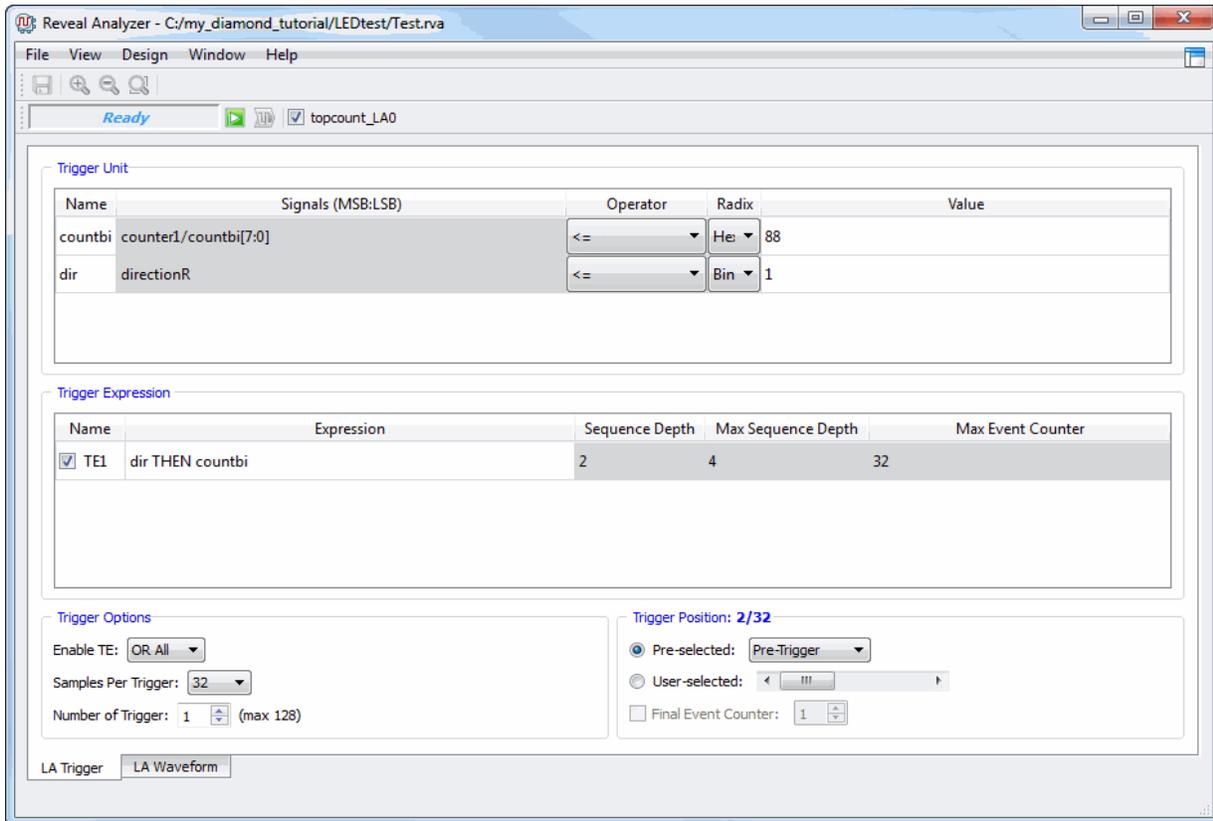
Figure 41: Reveal Analyzer Startup Wizard



- Click **OK**.

The Reveal Logic Analyzer main window now appears with the LA Trigger tab selected, as shown in Figure 42. It contains the same trigger units and trigger expressions that you set up in Reveal Inserter.

Figure 42: Reveal Analyzer



9. Choose Trigger Options:

Samples Per Trigger: **512**.

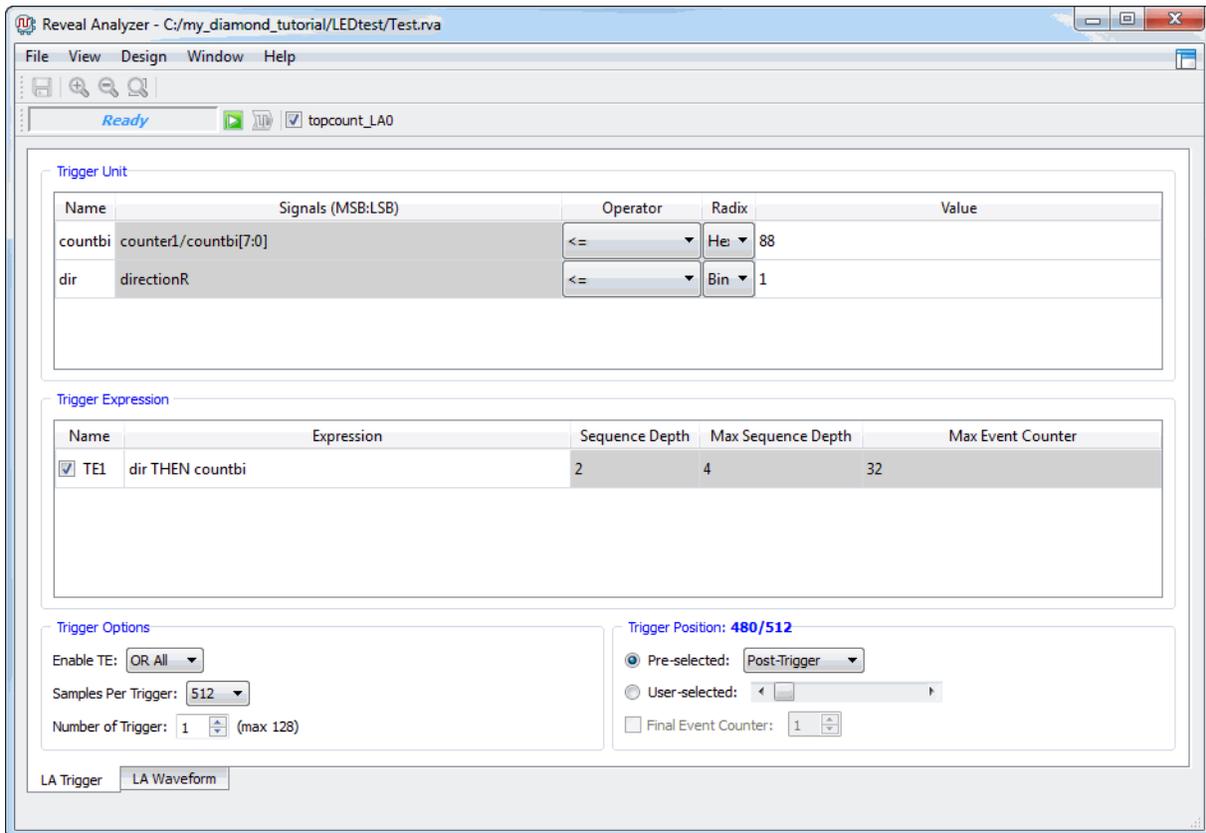
10. Choose Trigger Position: **Pre-selected: Post-Trigger**.

In the Trigger position section, you can specify the trigger position relative to the trace data. The numbers in the section title show the current position. The two options to choose from include:

- ▶ **Pre-selected** allows you to choose one of the standard positions.
 - ▶ Pre-Trigger: 32/512 of the way from the beginning of the samples.
 - ▶ Center-Trigger: 256/512 of the way from the beginning of the samples.
 - ▶ Post-Trigger: 480/512 of the way from the beginning of the samples.
- ▶ **User-selected** allows you to choose a position with the slider.

The Reveal Analyzer LA Trigger tab should now appear as shown in Figure 43.

Figure 43: Reveal Analyzer LA Trigger Tab



Running Logic Analyzer

Now that Reveal Logic Analyzer is set up, you can run Logic Analyzer.

To capture data:

1. Click the Run  button in the Reveal Analyzer toolbar.

The Run button changes into the Stop  button and the status bar next to the button shows the progress.

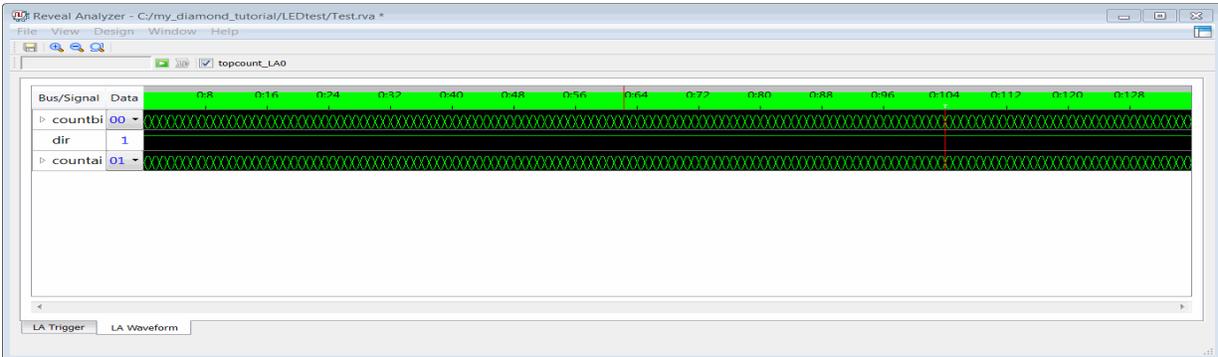
Reveal Analyzer first configures the modules selected for the correct trigger condition, then waits for the trigger conditions to occur. When a trigger occurs, the data is uploaded to your computer. The resulting waveforms appear in the LA Waveform tab.

Since the “countbi” trigger was set to ≤ 88 and the “dir” trigger was set to ≤ 1 , any DIP switch setting will immediately set off a trigger. The trigger expression can now evaluate the next trigger unit and generate a trigger for data to be captured.

If no trigger occurs click the **Manual Trigger**  button.

You now see the waveforms displayed, but the data may be varied as shown in Figure 44.

Figure 44: Reveal Analyzer Waveform



Summary of Accomplishments

You have completed the *Lattice Diamond Tutorial*. In this tutorial, you have learned how to:

- ▶ Create a new Lattice Diamond project
- ▶ Create an IPexpress module
- ▶ Check Hardware Description Language (HDL)
- ▶ Inspect strategy settings
- ▶ Examine resources
- ▶ Run synthesis processes
- ▶ Set timing and location assignments
- ▶ Run place and route
- ▶ Examine post place and route results
- ▶ Adjust static timing constraints and review results
- ▶ Compare multiple place and route runs
- ▶ Verify functionality with simulation
- ▶ Analyze power consumption
- ▶ Run export utility programs
- ▶ Download a bitstream to an FPGA
- ▶ Convert a file using Deployment Tool
- ▶ Use Reveal Inserter to add on-chip debug logic
- ▶ Use Reveal Logic Analyzer to perform logic analysis

Recommended References

You can find additional information on the subjects covered by this tutorial in the Diamond software online Help, and in the *Diamond User Guide*.

