

# Lattice Diamond 3.13 Tutorial



August 17, 2023

---

## Copyright

Copyright © 2023 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](http://www.latticesemi.com/legal). Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. Modelsim and Questa are trademarks or registered trademarks of Siemens Industry Software Inc. or its subsidiaries in the United States or other countries. 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
<b>Bold</b>	Items in the user interface that you select or click. Text that you type into the user interface.
<i>&lt;Italic&gt;</i>	Variables in commands, code syntax, and path names.
<b>Ctrl+L</b>	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

<b>Lattice Diamond 3.13 Tutorial</b>	<b>6</b>
About the Tutorial	6
Task 1: Create a New Lattice Diamond Project	9
Task 2: Create an IPexpress Module	14
Task 3: Verify Functionality with Simulation	16
Starting a Simulation Run	17
Checking the Simulation Results	18
Rerunning the Simulation	20
Task 4: Inspect Strategy Settings	21
Task 5: Examine Resources	22
Task 6: Run Synthesis Process	24
Task 7: Set Timing and Location Assignments	25
Task 8: Running Place and Route	28
Task 9: Examine Static Timing Analysis Results	30
Task 10: Adjust Static Timing Constraints and Review Results	32
Task 11: Analyze Power Consumption	33
Task 12: Run Export Utility Programs	35
Task 13: Download a Bitstream to an FPGA	35
Task 14: Add On-Chip Debug Logic	37
Setting Up the Trigger Units	38
Setting Up the Trigger Expressions	39
Inserting the Debug Logic	40
Generating a Bitstream and Programming the FPGA	40
Task 15: Perform Logic Analysis	41
Creating a New Reveal Logic Analyzer Project	41
Running Logic Analyzer	43
Summary of Accomplishments	44
Recommended References	45

**Revision History 46**

# Lattice Diamond 3.13 Tutorial

This tutorial leads you through all the basic steps of designing and implementing a mixed VHDL and Verilog design targeted to the MachXO3L 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 examining the device resources, setting timing and location assignments, programming the device, and adding a logic analyzer to the design.

## About the Tutorial

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.
- ▶ Set timing and location assignments.
- ▶ Process the design.
- ▶ Examine static timing analysis results.
- ▶ Analyze power consumption.
- ▶ Run export utility programs.
- ▶ Download a bitstream to an FPGA.
- ▶ Use Reveal Inserter to add on-chip debug logic.
- ▶ Use Reveal Logic Analyzer to perform logic analysis.

## Time to Complete This Tutorial

About 90 minutes.

You can stop at the end of any task and restart at the beginning of the next task. When you restart the Diamond software, it shows a Recent Projects list. Just click the name of your project.

## System Requirements

You need:

- ▶ Lattice Diamond software, version 3.13, free or subscription version

To download Diamond software, go to:

<http://www.latticesemi.com/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond.aspx>

- ▶ MachXO3L Starter Kit (Optional)

Many of the tasks in this tutorial can be performed without an actual MachXO3L Starter Kit, but the low-cost MachXO3L Starter Kit is recommended to perform all of the tasks in this tutorial.

To purchase a MachXO3L Starter Kit, go to:

<http://http://www.latticesemi.com/en/Products/DevelopmentBoardsAndKits/MachXO3LStarterKit.aspx>

## Online Help

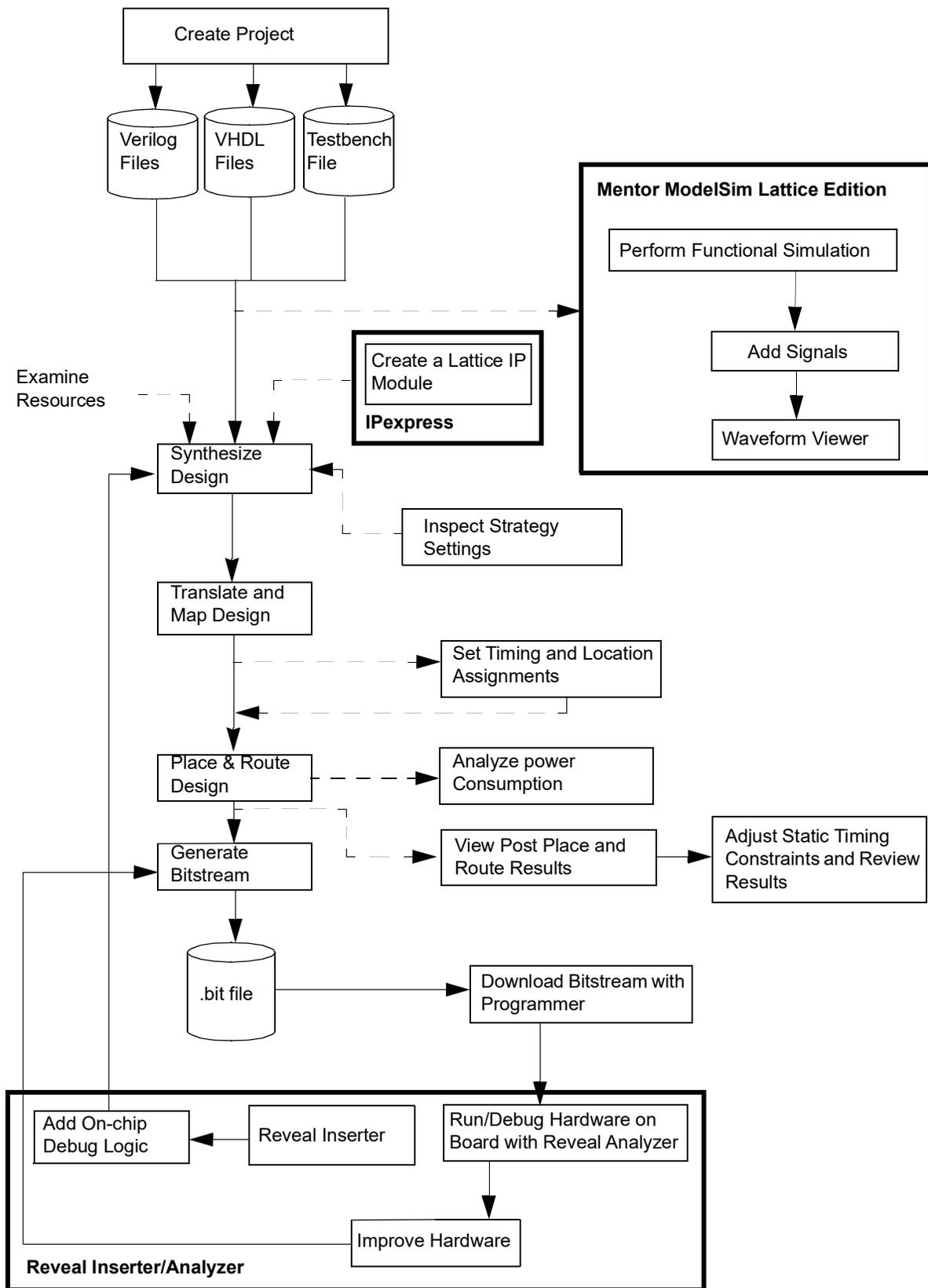
You can find additional information on any tool included in the tutorial at any time by choosing **Help > Lattice Diamond Help** or **Help > <tool name>**. The Help also provides easy access to many other information sources.

## About the Tutorial Design

The design in this tutorial consists of four Verilog HDL modules and two VHDL modules. The design that you create is targeted to the MachXO3L device family.

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.

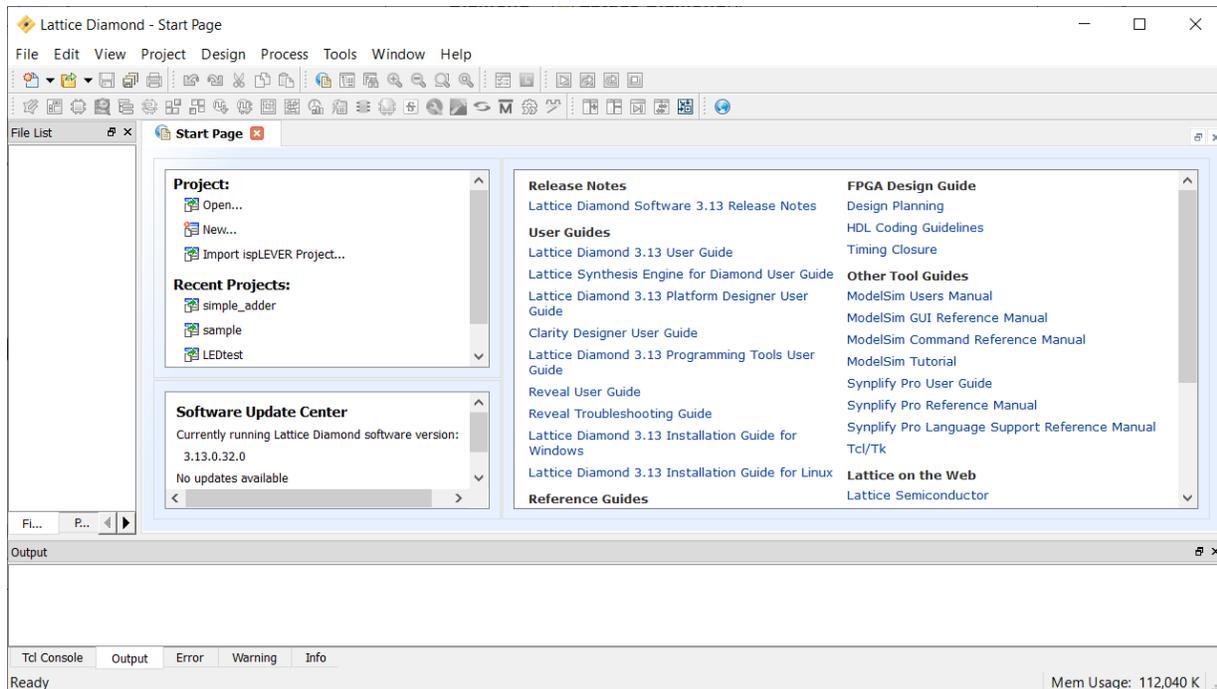
### To create a new project:

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

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

The main window of Lattice Diamond appears, as shown in Figure 2. This will take a few moments.

Figure 2: Diamond Main Window



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. Links in the right pane of the Start Page provide access to user guides, reference material, and online resources available from [www.latticesemi.com](http://www.latticesemi.com).

For almost all questions, the place to start is the Lattice Diamond 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**.
  - ▶ Choose **File > New >  Project**.
  - ▶ In the toolbar, click the down arrow in the New  button and then choose  **Project**.

The New Project wizard opens.

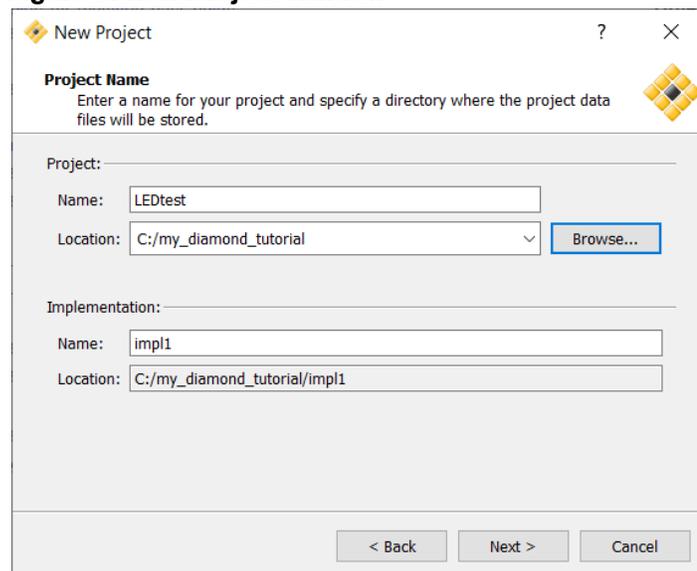
3. Click **Next**.  
The Project Name page opens.
4. Specify the project name: **LEDtest**.

#### 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 Window**



7. Click **Next**.

The Add Source dialog box appears.

8. Click **Add Source**.

The Import File dialog box appears.

9. Browse to: `<diamond_install_directory>/docs/tutorial/Diamond_tutorial`.

10. Select the following files (you can use **Control+A**):

- ▶ count8.vhd
- ▶ count32.v
- ▶ testbench.v
- ▶ topcount.v
- ▶ typepackage.vhd

Click **Open**.

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

11. Select **Copy source to implementation directory**.

12. Click **Next**.

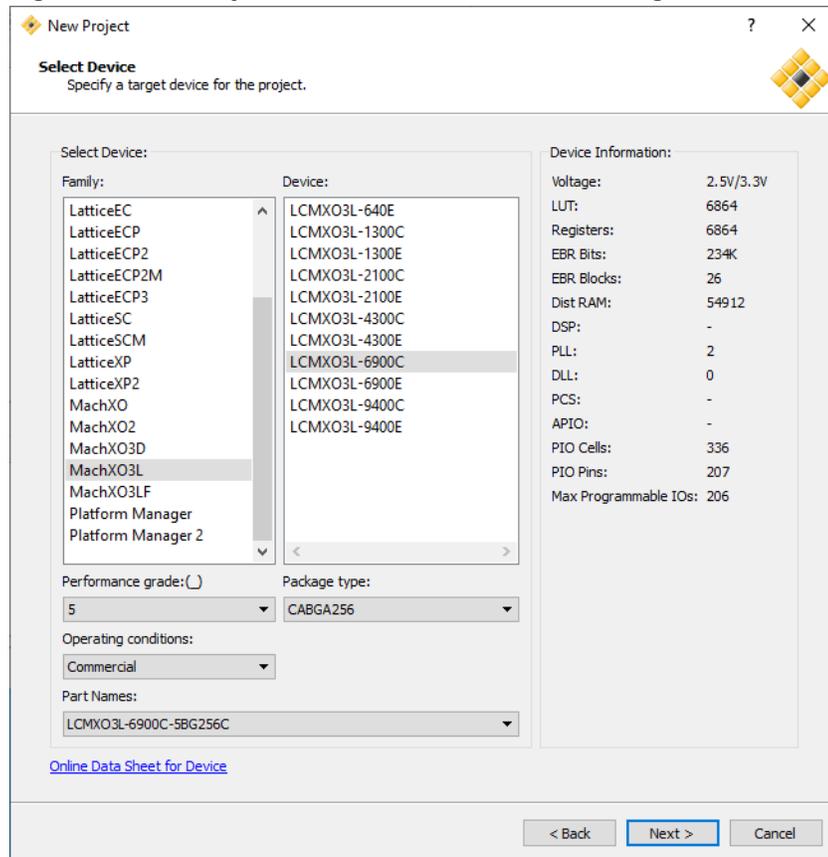
The Device Selector dialog box appears.

13. Select the following device options:

- ▶ Family: **MachXO3L**
- ▶ Device: **LCMXO3L-6900C**
- ▶ Performance grade: **5**
- ▶ Package type: **CABGA256**
- ▶ Operating conditions: **Commercial**

Part Names, at the bottom, changes as you make selections.

The dialog box should resemble Figure 4.

**Figure 4: New Project Wizard Device Selector Dialog Box**

14. Click **Next**.

The Select Synthesis Tool dialog box opens.

15. Choose **Synplify Pro**.

16. Click **Next**.

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

17. 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 and Verilog 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 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 MachXO3L 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.

18. In the File List, right-click **LPF Constraint Files** and choose **Add > Existing File**.

The Add Existing File dialog box appears.

19. Navigate to `<diamond_install_directory>/docs/tutorial/Diamond_tutorial` and select **pin\_assignments.lpf**.

Select **Copy file to Implementation's Source directory** and click **Add**.

20. In the File List view, right-click **pin\_assignments.lpf** and choose **Set as Active Preference File**.

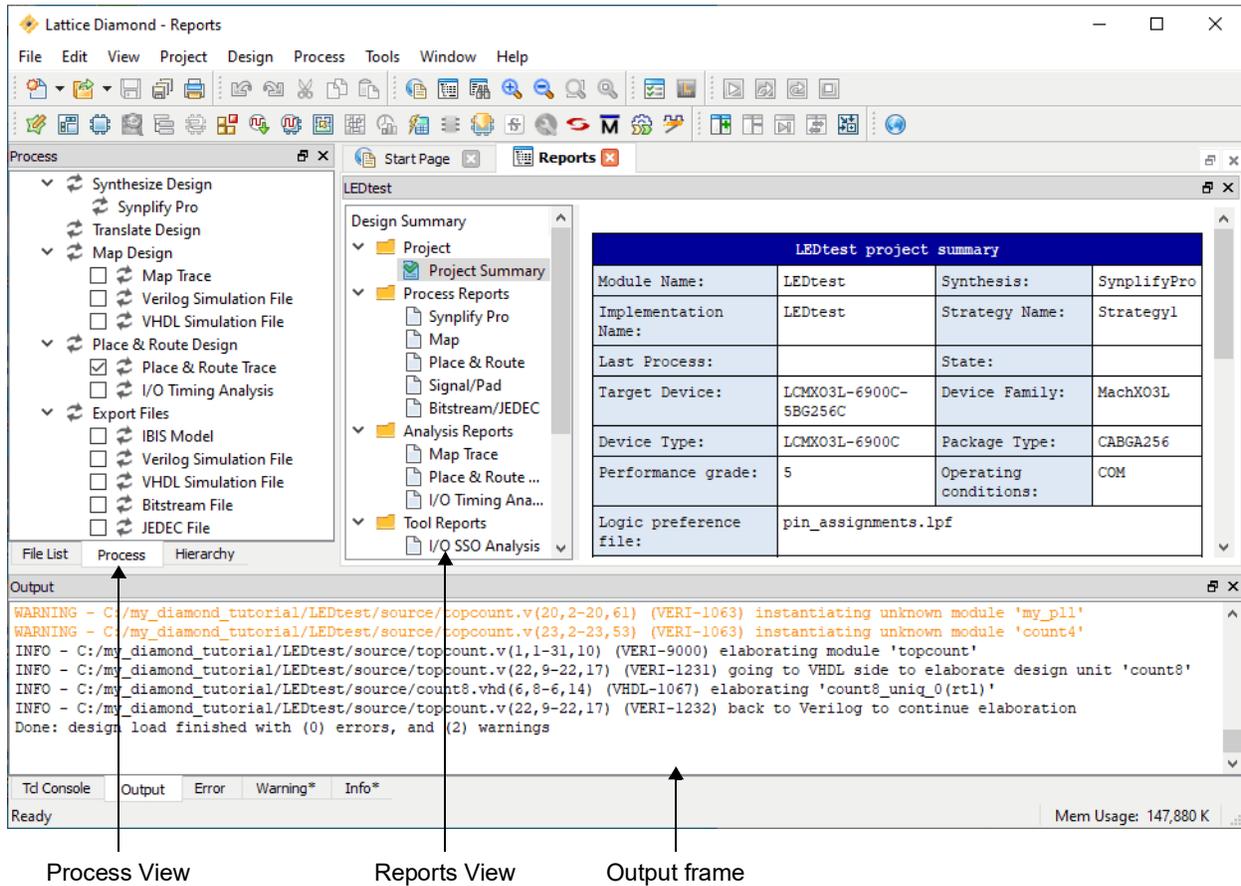
21. Under Input Files, right-click **testbench.v** and choose **Include for > Simulation**.

The Process view, shown in Figure 5, 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 5: 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.

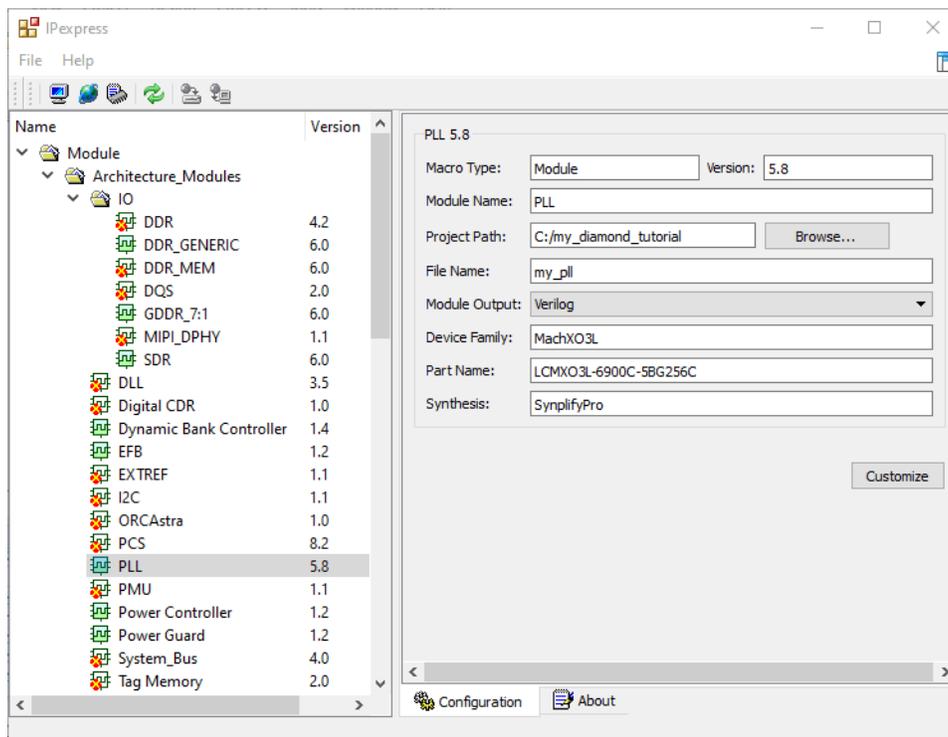
In this task, you will generate a phase lock loop (PLL) module to import into your design.

**To generate and import a module with IPexpress:**

1. Choose **Tools > IPexpress**.  
The IPexpress tool appears.
2. In the left pane, under **Module > Architecture\_Modules**, select **PLL**.

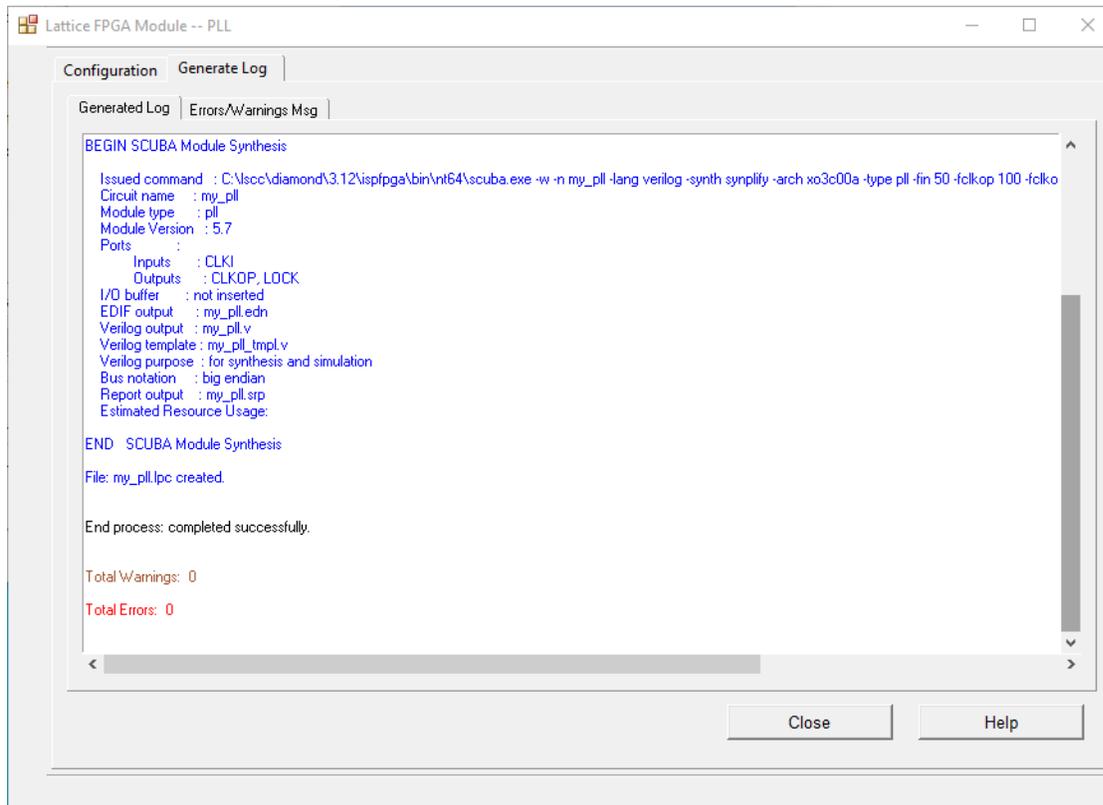
3. In the Configuration tab, all information is filled in from the design project except for File Name and Module Output. For this tutorial:
  - ▶ Enter **my\_pll** as the File Name.
  - ▶ Choose **Verilog** as the Module Output.
 IPexpress should appear similar to Figure 6.

**Figure 6: New Project Wizard Device Selector Dialog Box**



4. Click **Customize**.  
The Lattice FPGA Module — PLL dialog box appears.
5. In the Configuration tab:
  - ▶ Select **Frequency Mode**.
  - ▶ For CLKI Frequency, enter **20**.
  - ▶ For CLKOP Desired Frequency, enter **80**.
  - ▶ For CLKOP Tolerance, choose **1.0**.
  - ▶ Select **Provide PLL Lock signal**.
6. Click **Calculate**.
7. Select **Import IPX to Diamond project**.
8. Click **Generate**.  
The IPexpress .ipx file is generated.
9. The **Generated Log** tab appears, as shown in Figure 7.

Figure 7: IPexpress Generated Log Tab



10. Click **Close**.

The my\_pll.ipx file appears in the File List view under Input Files.

11. Close IPexpress by clicking the red X in the tab.

## Task 3: Verify Functionality with Simulation

Now that the design is finished, you can simulate it to test the logic. With Diamond, you can run a simulation at different stages of the development process:

- ▶ Before synthesis (RTL)
- ▶ Post-map, gate-level
- ▶ Post-route, gate-level and timing

In this tutorial, we will just do the RTL simulation. For the other stages, the process is similar.

For a simulator, this tutorial uses the Mentor® ModelSim® Lattice FPGA Edition simulator that comes with Diamond on Windows.

If you are not using an HDL simulator that is integrated with Diamond, you can skip this task. “Integrated” means that you can run the simulator from Diamond. What is available depends on your operating system. You can use other simulators outside of Diamond.

If you are not using the ModelSim that comes with Diamond, you need to compile the primitive library. For instructions, open the Diamond Help and see User Guides > Simulating the Design > Third-Party Simulators.

This tutorial comes with a simple testbench. You will probably create your own testbenches using your simulator. Simulators usually include tools for creating testbenches.

## Starting a Simulation Run

While you can start your simulator directly, it is good to create a simulator project that allows you to run the simulator from Diamond.

### To start simulating the design:

1. Choose **Tools** >  **Simulation Wizard**.  
The Simulation Wizard dialog box appears.
2. Click **Next**.  
The Simulator Project Name page appears.
3. Enter the Project name: **sim\_test**.  
Make sure ModelSim is selected for Simulator.
4. Click **Next**.
5. If you left the default for the project location, a dialog box opens saying, “sim\_test does not exist. Do you want to create it?” Click **Yes**. This creates a sim\_test folder.  
The Process Stage page appears with RTL selected.
6. Click **Next**.  
The Add and Reorder Source page appears.
7. In the Source Files list, select **typepackage.vhd**. Then click the up  arrow until typepackage.vhd is at the top of the list.  
ModelSim does not automatically reorder VHDL files. Similarly, SystemVerilog packages need to be placed in the proper order for the compiler.
8. Click **Next**.  
The Parse HDL files for simulation page appears.
9. Verify that the simulation top module is “testbench.” This is shown at the bottom of the dialog box. Click **Next**.  
The Summary dialog box appears.

10. Make sure that the **Run simulator**, **Add top-level signals to waveform display**, and **Run simulation** options are all selected.
11. Click **Finish**.  
The selected simulator launches and the simulation starts automatically. After completing the simulation, the waveform appears. This takes several moments. Wait for the waveform to appear.  
If you see the Welcome to ModelSim dialog box, select **Don't show this again**, at the bottom of the dialog box, and click **Close**. Do not click Jumpstart.
12. Look at the File List view in the Diamond window. Under Script Files, you see sim\_test/sim\_test.spf.

You can rerun the simulation by double-clicking the .spf file. The Simulation Wizard will open with a Skip to End button. Click it to jump to the last page of the wizard. Then click **Finish** to start the simulation running.

## Checking the Simulation Results

### Note

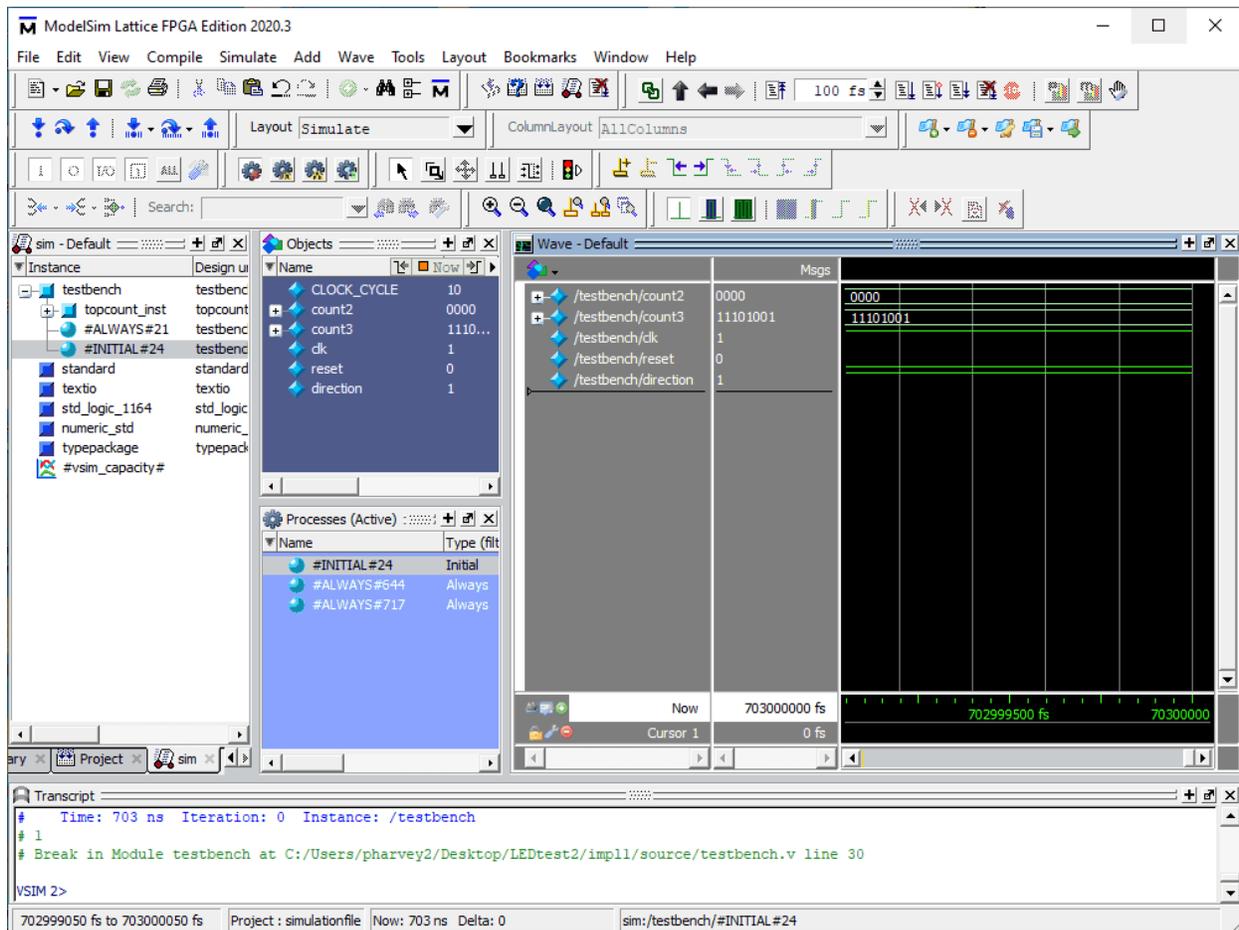
If you are not using ModelSim Lattice FPGA Edition, you can skip the rest of the simulator task.

Now that you have run the simulation, you can see what happened on the top-level signals of the testbench as shown in Figure 8. ModelSim stopped automatically after the first microsecond of simulation time. The testbench is set to run longer, over 5  $\mu$ s, but this is enough to see the startup.

### To check the simulation results:

1. If you want to expand the Wave view, do one of the following:
  - ▶ Expand the ModelSim window.
  - ▶ Undock the Wave view. Click the Dock/Undock  button that is in the upper-right corner of the Wave view. Then expand the Wave window.
2. To make other adjustments to the Wave view, choose **Simulate > Runtime Options** in the main ModelSim window.  
The Runtime Options dialog box opens showing a variety of options that you can set.
3. Make the following changes in the Defaults tab:
  - ▶ For Default Radix, select **Hexadecimal**. This is how the values of signals are normally displayed.
  - ▶ For Default Run, enter **100ns**. This is the amount of time that the Run command simulates.
4. Click **OK**.

Figure 8: Simulated Waveform



The values shown in the Objects and Wave views change to hexadecimal.

5. Choose **Wave > Zoom > Zoom Full** or click the Zoom Full  button in the toolbar to see the whole waveform. The Zoom toolbar looks like this:



In the Wave view, you see the reset signal activated by the testbench. After reset is released, count2 and count3 start counting up.

6. The values of count2 may not be visible. Click the Zoom In  button in the toolbar until you can see the values.
7. Choose **Simulate > Run > Run 100** or click the Run  button to see more of the simulation. The Run toolbar looks like this:



Another 100 ns is added to the waveforms. This is the time you set in the Runtime Options dialog box. You can change this amount in the box next to the Run button.

8. Click anywhere to see what the values are at that moment.

The nearest cursor (a vertical yellow line) jumps to where you clicked. The value column shows all the values at that moment.

You can click on the cursor and drag it to other positions on the timeline. You can also return to the cursor after scrolling away by clicking the Zoom In on Active Cursor  button.

## Rerunning the Simulation

In ModelSim, you can make changes in the simulation and rerun it. For example, you can add more signals.

### To add a signal and rerun the simulation:

1. In the List View, click the sim tab (also known as the Structure View), and expand: **testbench > topcount\_inst**

2. Click on **counter1**.

The Objects view changes.

3. Drag countbi from the Objects view to the Wave view.

4. Rerun the simulation to see what is happening with the countbi register. Choose **Simulate > Restart** or click the Restart  button.

The Restart dialog box opens with a variety of features that you might have changed. You can leave them all selected.

5. Click **OK**.

The waveforms in the Wave view disappear.

6. Then choose **Simulate > Run > Run -All** or click the Run -All  button.

The Finish Vsim dialog box opens. It asks if you want to finish.

7. Click **No**.

### Warning

Do *not* click Yes. If you do, the \$finish statement in the testbench causes ModelSim to exit.

If this happens, go to the File List view in the Radiant window and look under the Script Files folder. Double-click **sim\_test/sim\_test.spf** to restart ModelSim.

ModelSim's source editor opens with the testbench.v file.

8. Close testbench.v to see the Wave view. You can now see the full 5  $\mu$ s.
9. You can take this opportunity to explore ModelSim more.

There is a lot more that you can do with ModelSim. For more information, see the **Help** menu in the ModelSim window.

10. When you are done exploring ModelSim, choose **File > Quit** to close ModelSim.

The Quit Vsim dialog box opens.

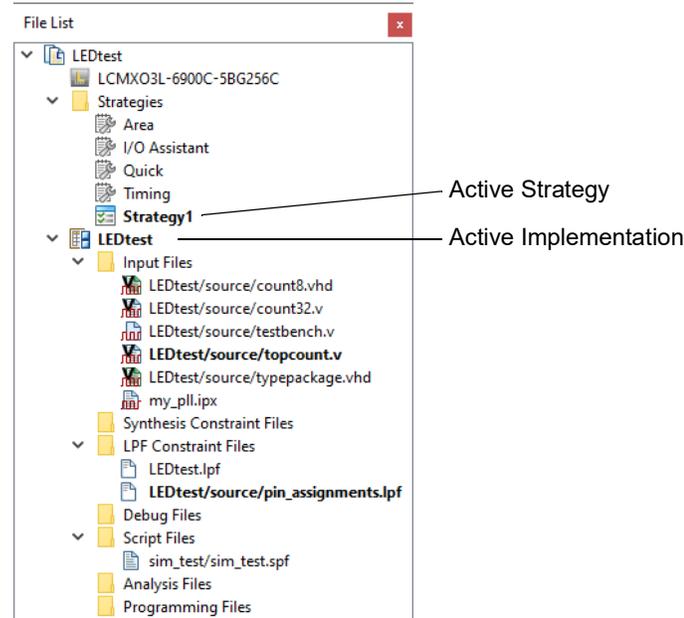
11. Click **Yes**.

## Task 4: Inspect Strategy Settings

Implementations define the design structural elements for a project, including source code, constraint files, and debug insertion. Implementations contain all source files, constraint files, debug files, scripts, and analysis files. Source can mix VHDL, Verilog, and 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 and 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, shown in Figure 9, and create other strategies to experiment with or to use in different circumstances. Predefined strategies can also be cloned and then modified.

**Figure 9: Implementations and Strategies**



### To adjust synthesis settings:

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

The Strategies — Strategy1 dialog box, shown in Figure 10, appears.

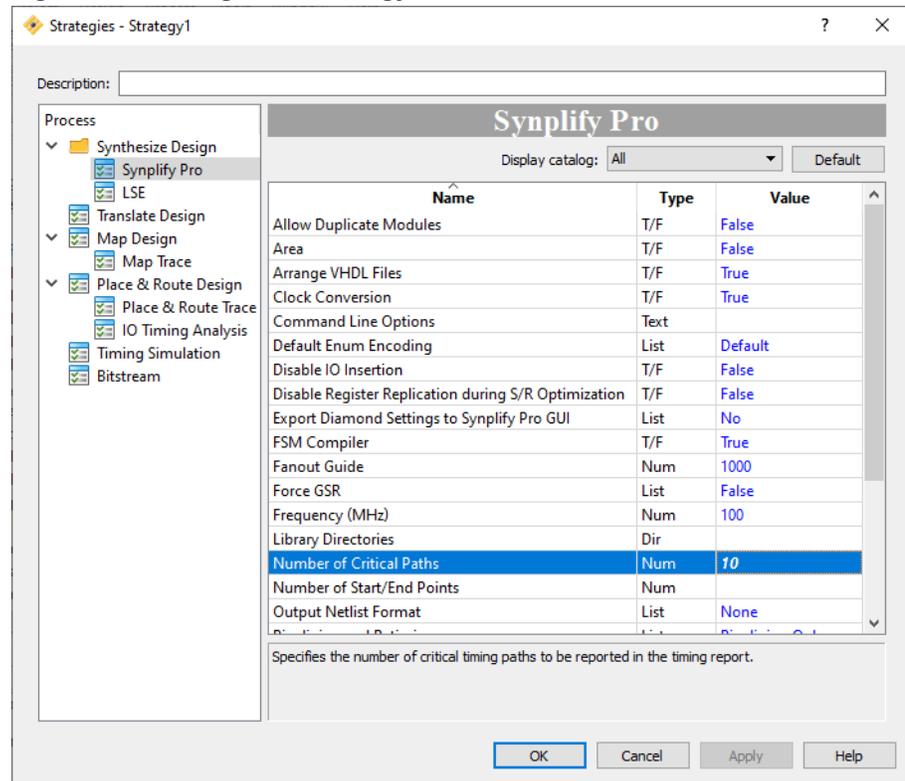
2. Click **Synthesize Design > Synplify Pro**.

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

For information on FPGA Design Constraints File (.fdc) use in Synplify Pro, see the *Synplify Pro for Lattice Reference Manual* in the Synplify Pro Help menu.

3. Click **Number of Critical Paths** and enter **10** in the Value column.

Figure 10: Strategies -- Strategy 1



When each strategy is selected, descriptive text appears in the lower panel of the dialog box. Changed values are shown in italics.

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.

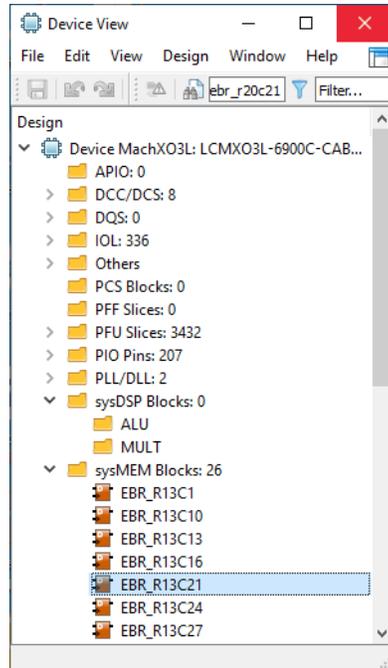
### To browse device resources:

1. Choose **Tools** >  **Device View**.  
The Device View appears.
2. Click the Detach Tool  button 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 MachXO3L** folder.  
Several folders organized by feature type appear.
4. Expand the **sysDSP Blocks** and **sysMEM Blocks** folders.

5. In the Find box at the top of the Device View, enter **EBR\_R13C21** (Embedded Block RAM Row 13, Column 21).

The EBR design symbol is highlighted, as shown in Figure 11.

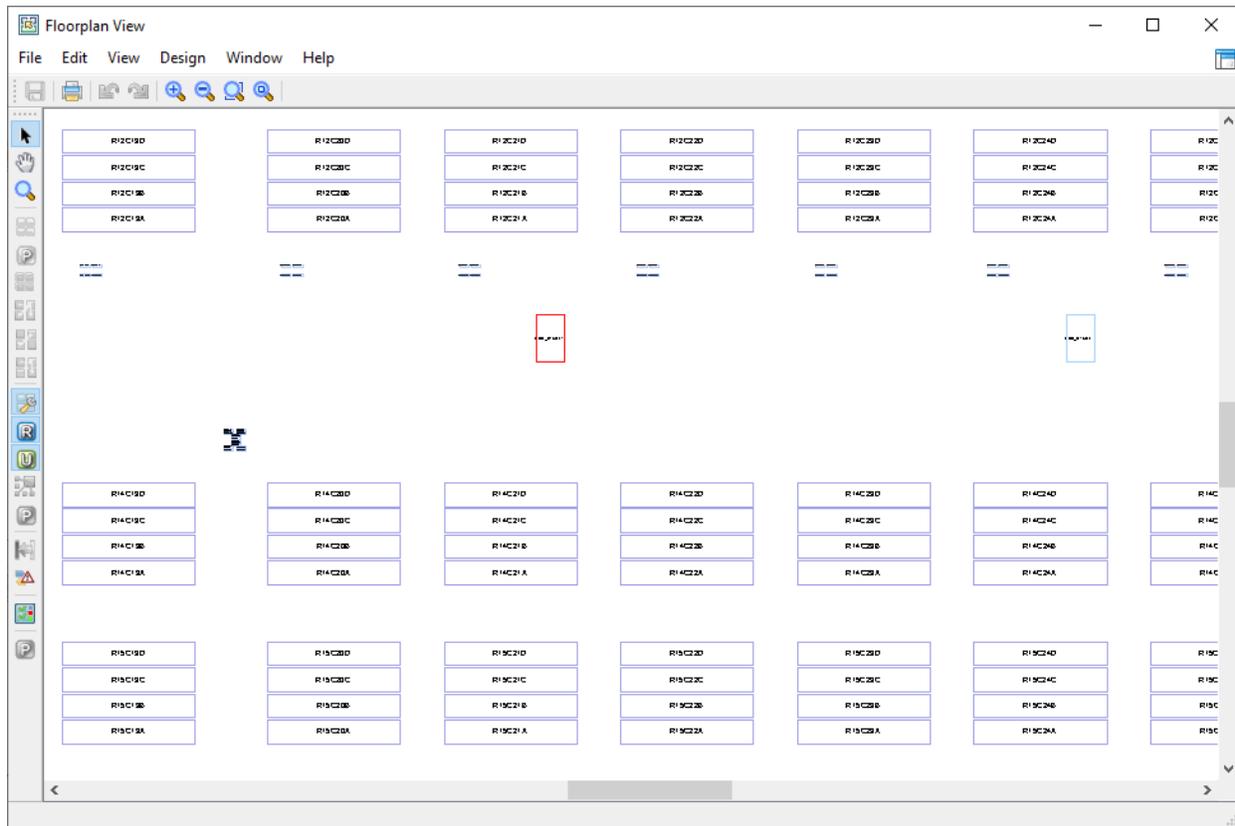
**Figure 11: Device View**



6. Right-click **EBR\_R13C21** and choose **Show in > Floorplan View**.

Floorplan View, shown in Figure 12, provides a large-component layout of your design. It displays user constraints from the logical preference file (.lpf), and placement and routing information.

Figure 12: Floorplan View



7. Close Floorplan View and 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.

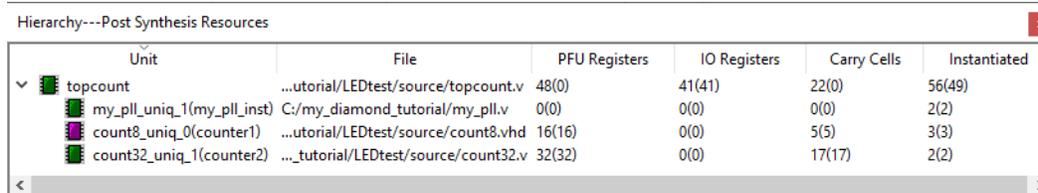
You will be using Synopsys Synplify Pro for Lattice to synthesize your design for the MachXO3L FPGA. If you are designing for most devices, you can use Lattice Synthesis Engine or another third-party synthesis tool instead of Synplify Pro for Lattice. LSE is not available with LatticeEC, LatticeECP, LatticeSC/M, or LatticeXP. 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. In 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.

2. Click the Hierarchy---Post Synthesis Resources tab, as shown in Figure 13.

**Figure 13: Post Synthesis Resources**


Unit	File	PFU Registers	IO Registers	Carry Cells	Instantiated
topcount	...utorial/LEDtest/source/topcount.v	48(0)	41(41)	22(0)	56(49)
my_pll_uniq_1(my_pll_inst)	C:/my_diamond_tutorial/my_pll.v	0(0)	0(0)	0(0)	2(2)
count8_uniq_0(counter1)	...utorial/LEDtest/source/count8.vhd	16(16)	0(0)	5(5)	3(3)
count32_uniq_1(counter2)	..._tutorial/LEDtest/source/count32.v	32(32)	0(0)	17(17)	2(2)

The post-synthesis Hierarchy View displays the number of logical resources within each level of the design.

In the Hierarchy table shown in Figure 13, topcount is the top module displaying the resource utilization.

- ▶ PFU Registers – 48 represents the total PFU register utilization throughout the design and 0 represents the PFU registers used only in the design module topcount. Similar utilization is shown for the I/O registers and carry cells.
- ▶ my\_pll\_uniq\_1, count8\_uniq\_0, and count32\_uniq\_1 are the sub-modules (instances) of the design. For example, for the sub-module count32\_uniq\_1, 32(32) under PFU Registers represents the total PFU registers used 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 and 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 will set default timing constraints for the operating frequency and I/O timing.

**To set timing and location assignments:**

1. In the Process view, double-click **Translate Design**.  
Many of the functions available to create preferences require the translated design.
2. Choose **Tools >  Spreadsheet View**.  
Spreadsheet View appears. 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 Spreadsheet View.
3. Click the Detach Tool  button at the upper-right corner of Spreadsheet View.  
Spreadsheet View is detached from the Diamond main window.
4. Click the PERIOD/FREQUENCY Preference  button on the Spreadsheet View tool bar.  
The PERIOD/FREQUENCY Preference dialog box appears.
5. Enter the following preference settings:  
Type: **FREQUENCY**  
Second Type: **Net**  
Available Clock Nets: **CLKOP**  
Frequency: **80**
6. Click **OK**.  
The Timing Preferences tab of Spreadsheet View appears with the new FREQUENCY preference defined.
7. Click the **INPUT\_SETUP/CLOCK\_TO\_OUT Preference**  button on the Spreadsheet View toolbar.  
The INPUT\_SETUP/CLOCK\_TO\_OUT Preference dialog box appears.
8. Enter the following preference settings:  
Type: **INPUT\_SETUP**  
Second Type: **All ports**  
Clock Ports/Nets: **clk**  
Time: **50**  
Hold time: **12**
9. 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.
10. In the Timing Preferences tab, right-click **INPUT\_SETUP** and choose **New INPUT\_SETUP**.  
The INPUT\_SETUP/CLOCK\_TO\_OUTPUT Preference dialog box appears.

11. Enter the following settings:

Type: **INPUT\_SETUP**

Second Type: **Individual Ports**

Available Input Ports: **reset**

Clock Ports/Nets; **clk**

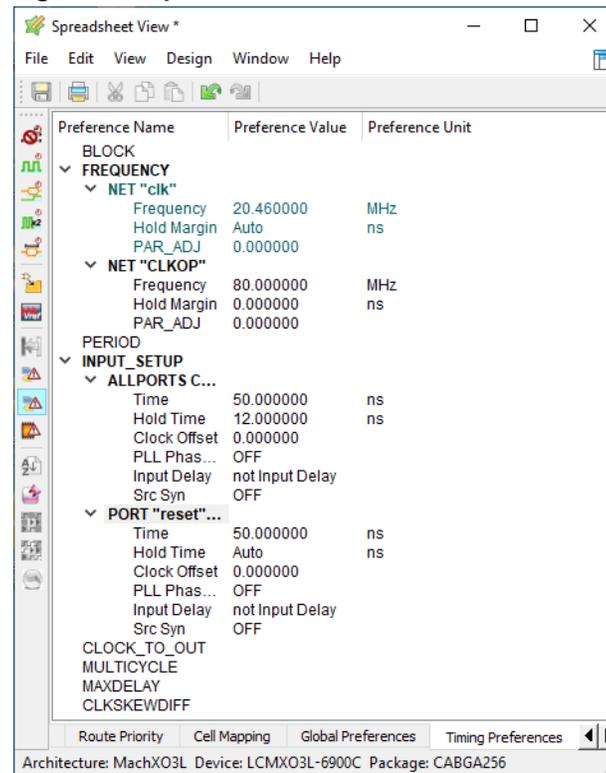
Time: **50**

12. Click **OK**.

The Timing Preferences tab of the Spreadsheet View appears, as shown in Figure 14, with the new INPUT\_SETUP preference defined.

The preference dialog box can be invoked from the toolbar icon, the menu item (Edit > Preference 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 14: Spreadsheet View**



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

This tab shows the port assignments that came with the pin\_assignments.lpf file and specifications for each port. You could make changes here if you needed to.

14. Choose **File > Save pin\_assignments.lpf** from the Spreadsheet View.

15. Close Spreadsheet View.

16. In the File List view, under the LPF Constraint Files folder, double-click **pin\_assignments.lpf**.  
Source Editor appears. Note the timing and location preferences defined.
17. Close Source Editor.
18. In the Process view, double-click **Map Design**.  
The batch interface to logic synthesis, EDIF translation, and the design mapper run. Report files appears in the Reports view. To view each process report, select the process in the Design Summary pane.
19. From the Design Summary pane of the Reports view, select **Process Reports > Map**.  
The Map Report appears in the right pane.
20. Right-click in the right pane of the Reports view.
21. Choose **Find in Text**.  
A box labeled "Find" appears at the bottom of the Reports view.
22. In the Find box, enter **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 arrow before the report name 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.

## 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. In the Process view, double-click **Place & Route Design**.  
The place and route tools run. Intermediate results appear in the Output pane 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. In the Process view, double-click **Place & Route Trace**.  
The TRACE timing analyzer runs.

4. In 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. In the Process view, double-click **I/O Timing Analysis**.

The timing analysis runs.

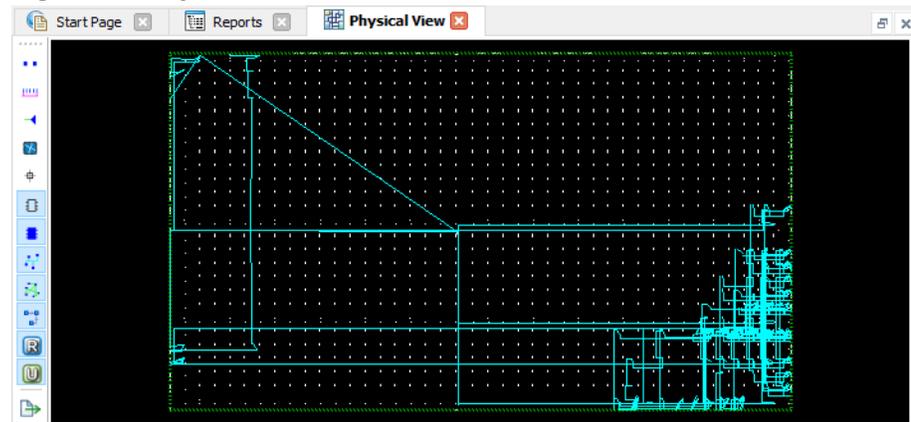
6. In the Design Summary pane of the Reports view, select the **I/O Timing Analysis** report.

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 15. Physical View provides a read-only detailed layout of your design that includes switch boxes and physical wire connections.

**Figure 15: 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 Zoom To  button.
9. Right-click on the component and choose **Show in > Floorplan View**.

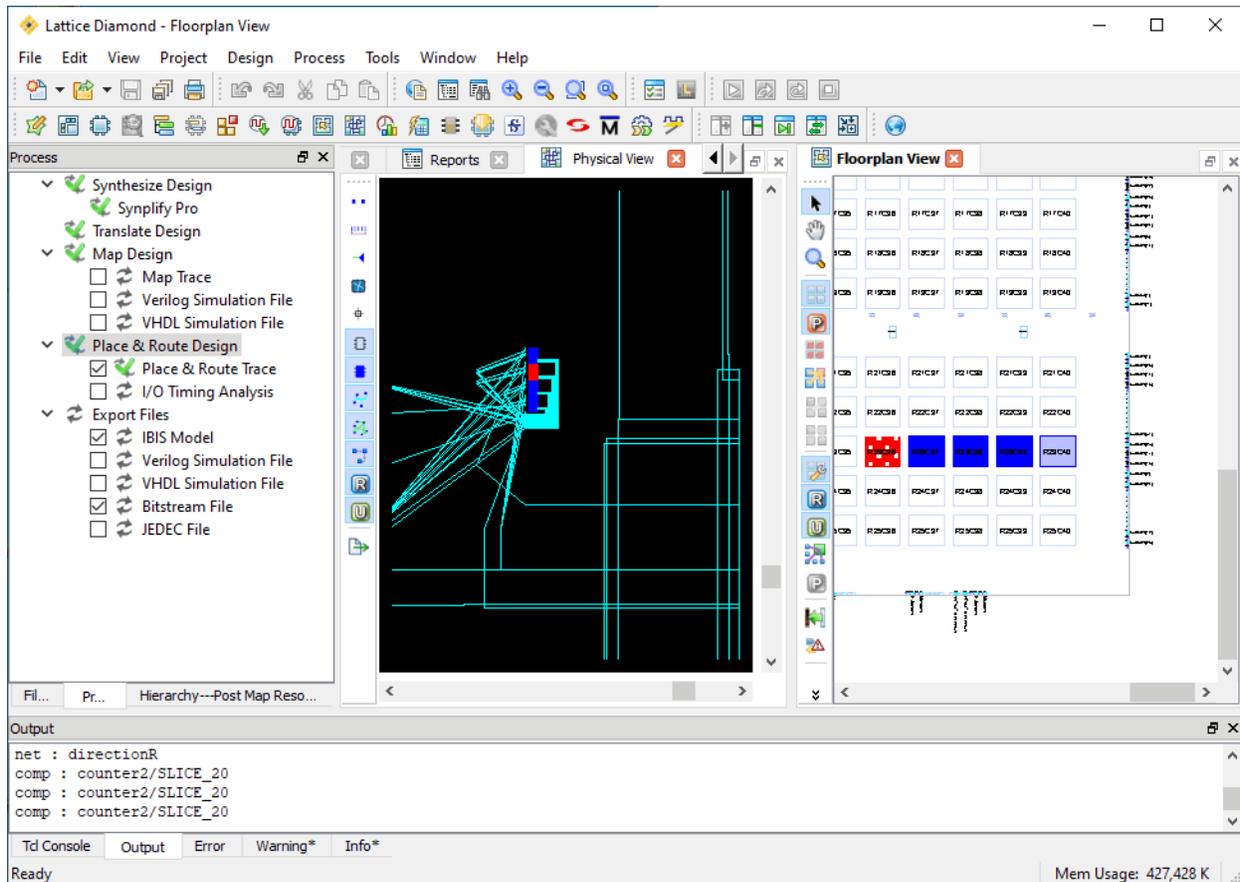
Floorplan View appears.

10. 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 choose **Split Tab Group**.

The two views display in parallel, as shown in Figure 16.

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 16: Cross Probing



11. 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.

12. Close Floorplan View and Physical View.

## Task 9: Examine Static Timing Analysis Results

Static Timing Analysis (STA) can determine if your circuit design meets timing constraints. Rather than simulation, STA 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**.

Timing Analysis View appears. 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.

2. Click the Detach Tool  button in the upper-right corner.

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

3. From the Preference Reports tab (on the lower-left of Timing Analysis View), click **FREQUENCY NET "CLKOP"** (either one).

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

4. Select row 1 of the Path Table.

The three tabs in the lower-right pane are populated with details.

5. Click the Detailed Path Tables tab, then click 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.

6. Click the Schematic Path View tab.

A schematic of the data path timing path appears.

7. Close Timing Analysis View.

Diamond allows you 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.

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

9. In the Global Preferences tab, double-click the Preference Value for SYSTEM\_JITTER(ns). Enter **0.1** in the box.

10. Close Spreadsheet View.

The Confirm dialog box appears asking if you want save the change.

11. Click **Discard**.

## Task 10: Adjust Static Timing Constraints and Review Results

In this task, you will edit timing constraints for STA (Static Timing Analysis) using the Timing Preference File (TPF) version of Spreadsheet View, and then you will use Timing Analysis View to review the results.

Timing analysis within Lattice Diamond can be performed at four points in a typical design flow: post-synthesis, post-map when the post-synthesis netlist of the design has been translated to the target device, post-placement, and post-route. Each stage provides a progressively more accurate report of delay characteristics. Timing analysis at the synthesis stage is performed by the respective synthesis tool: Synplify Pro or Precision. Diamond provides additional features for post-map stages of STA.

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 constraints used for MPAR.

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

1. Choose **Tools** >  **Timing Analysis View**.

The Timing Analysis View appears.

2. Click the Change timing preferences  button.

Spreadsheet View – TPF appears.

3. Click the Timing Preferences tab.
4. Scroll down to INPUT SETUP > ALLPORTS CLKPORT "clk" > Time.
5. Double-click the Preference Value for the Time row and enter **15**, as shown in Figure 17.

Figure 17: Spreadsheet View

Preference Name	Preference Value	Preference Unit
BLOCK		
FREQUENCY		
NET "clk"		
Frequency	20.460000	MHz
Hold Margin	0.000000	ns
PAR_ADJ	0.000000	
NET "CLKOP"		
Frequency	80.000000	MHz
Hold Margin	0.000000	ns
PAR_ADJ	0.000000	
PERIOD		
INPUT_SETUP		
ALLPORTS CLKNET "clk"		
Time	15.000000	ns
Hold Time	12.000000	ns
Clock Offset	0.000000	
PLL Phase Back	OFF	
Input Delay	not Input Delay	
Src.Svn	OFF	

Global Preferences    Timing Preferences    Group

Architecture: MachXO3L Device: LCMXO3L-6900C Package: CABGA256

The Update button on the toolbar of Timing Analysis View is rotating.

- Click the Update button.

After a short while, the indicator stops rotating and the new analysis results become available. 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. 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 Help.

- Close Timing Analysis View. Spreadsheet View – TPF closes automatically.

## Task 11: 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:

- Choose **Tools** > **Power Calculator**.

Power Calculator opens in Calculation mode.

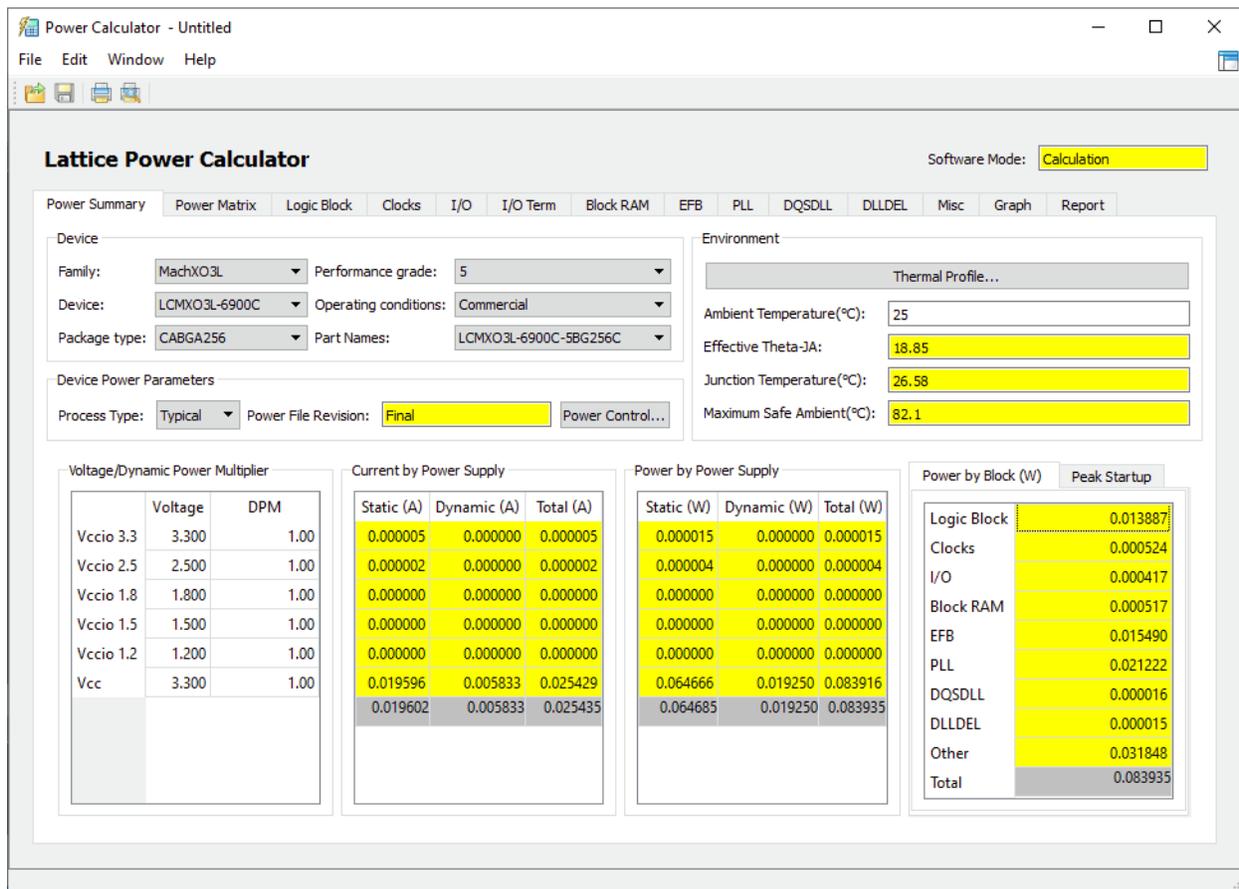
Power Calculator provides two modes for reporting power consumption:

- ▶ 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: 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 Detach Tool  button in the upper-right corner to detach Power Calculator from the Diamond main window, as shown in Figure 18.

**Figure 18: Power Calculator**



**Lattice Power Calculator** Software Mode: **Calculation**

Power Summary | Power Matrix | Logic Block | Clocks | I/O | I/O Term | Block RAM | EFB | PLL | DQSDLL | DLLDEL | Misc | Graph | Report

**Device**

Family: MachXO3L Performance grade: 5  
 Device: LCMXO3L-6900C Operating conditions: Commercial  
 Package type: CABGA256 Part Names: LCMXO3L-6900C-5BG256C

**Environment**

Thermal Profile...

Ambient Temperature(°C): 25  
 Effective Theta-JA: 18.85  
 Junction Temperature(°C): 26.58  
 Maximum Safe Ambient(°C): 82.1

**Device Power Parameters**

Process Type: Typical Power File Revision: Final Power Control...

**Voltage/Dynamic Power Multiplier**

	Voltage	DPM
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
Vcc	3.300	1.00

**Current by Power Supply**

	Static (A)	Dynamic (A)	Total (A)
	0.000005	0.000000	0.000005
	0.000002	0.000000	0.000002
	0.000000	0.000000	0.000000
	0.000000	0.000000	0.000000
	0.000000	0.000000	0.000000
	0.019596	0.005833	0.025429
	0.019602	0.005833	0.025435

**Power by Power Supply**

	Static (W)	Dynamic (W)	Total (W)
	0.000015	0.000000	0.000015
	0.000004	0.000000	0.000004
	0.000000	0.000000	0.000000
	0.000000	0.000000	0.000000
	0.000000	0.000000	0.000000
	0.064666	0.019250	0.083916
	0.064685	0.019250	0.083935

**Power by Block (W) Peak Startup**

Logic Block	0.013887
Clocks	0.000524
I/O	0.000417
Block RAM	0.000517
EFB	0.015490
PLL	0.021222
DQSDLL	0.000016
DLLDEL	0.000015
Other	0.031848
<b>Total</b>	<b>0.083935</b>

3. In the Device Power Parameters section, for Process Type, choose **Worst**.
4. Click the **Thermal Profile** button in the Environment section.  
The Power Calculator – Thermal Profile dialog box appears.

5. In the Board Selection section, choose **Small board**.
6. 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.

The Confirm dialog box appears asking if you want to save changes.

8. Click **No**.

## Task 12: 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 MachXO3L device in the next task.

### To run Export Files:

1. In the Process view, under Export Files, select:

- ▶ IBIS Model
- ▶ VHDL Simulation File
- ▶ Bitstream File

2. Click Export Files.

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

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

## Task 13: Download a Bitstream to an FPGA

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 MachXO3L FPGA mounted on a MachXO3L Starter Kit board.

### Note

If you do not have a MachXO3L Starter Kit, skip this task.

**To download the bitstream to the FPGA:**

1. Connect the USB cable (included with the Starter Kit) from your computer to the board.

On the board, the blue power light should turn on and the red LEDs should be blinking a pattern. The pattern depends on how the board's DIP switch is set.

2. Choose **Tools > Programmer**.

The Programmer: Getting Started dialog box opens.

3. In the dialog box, select **Create a new project from a JTAG scan**.

4. Click **Detect Cable**.

The Cable and Port boxes should automatically change to show the connection to your board. If you see the Programmer: Multiple Cables Detected dialog box, select the correct port and click **OK**.

5. Click **OK**.

Programmer scans the device database, and then the Programmer view displays in Diamond. This will take a few moments.

6. In Programmer, click anywhere in the first row.

7. Choose **Edit > Device Properties**.

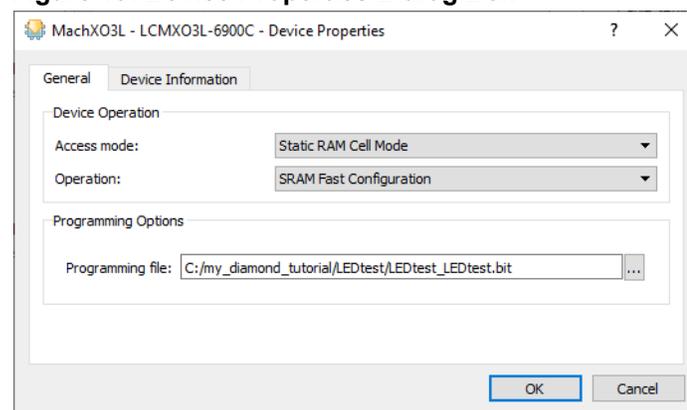
The Device Properties dialog box opens.

8. Choose the following settings:

- ▶ Access Mode: **Static RAM Cell Mode**
- ▶ Operation: **SRAM Fast Configuration**
- ▶ Programming File: *<project\_folder>/LEDtest/LEDTest\_LEDTest.bit*.

The dialog box should resemble Figure 19.

**Figure 19: Device Properties Dialog Box**



9. Click **OK**.

10. Click the Program button on the Programmer toolbar to initiate the download.

If the programming process succeeds, you will see a green-shaded PASS in the Programmer Status column. On the board, the red LEDs switch to a counting pattern.

11. Toggle DIP switch 1 to reverse the counting pattern. DIP switch 1 is the switch furthest from the LEDs.
12. In Diamond, choose **File > Save LEDtest.xcf**.
13. Close Programmer and unplug the board's USB cable.

## Task 14: Add On-Chip Debug Logic

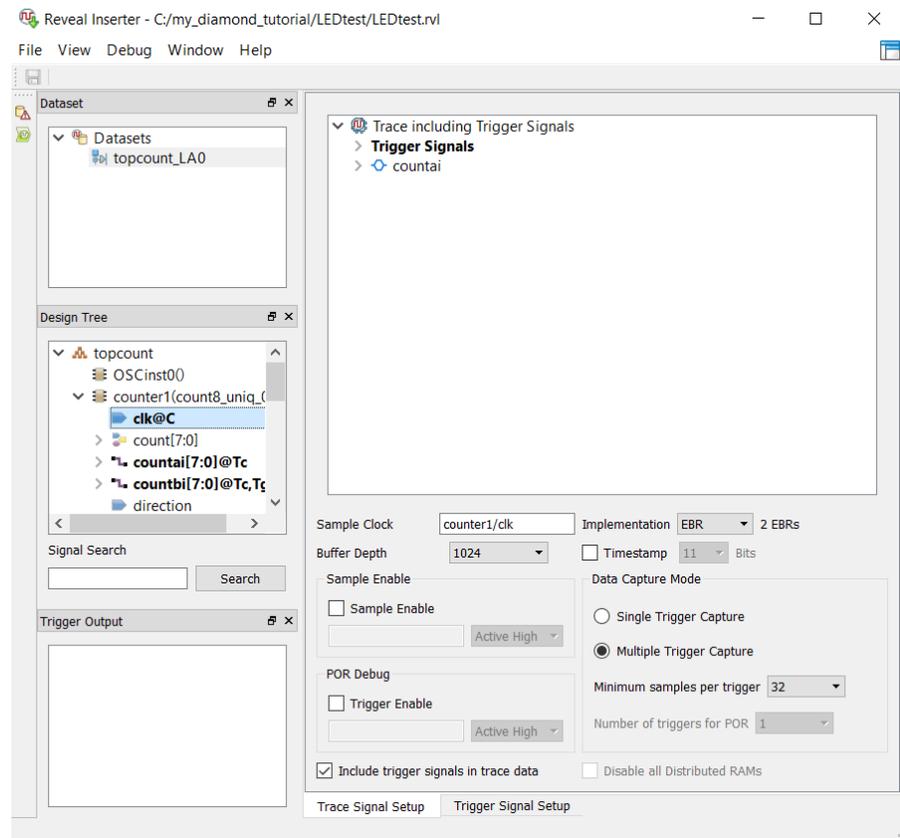
In this task, you will use Reveal Inserter to configure a Reveal core based on triggering conditions and the desired trace buffer. The primary output of Reveal Inserter is a modified version of your design with one or more cores instantiated and the core logic ready for mapping, placement, and routing.

### To generate and add a Reveal core:

1. Choose **Tools >  Reveal Inserter**.
2. Click the Detach Tool  button in the upper-right corner to detach Reveal Inserter.  
Reveal Inserter is detached from the Diamond main window.
3. Click on the **Trace Signal Setup** tab, if it is not already selected.
4. In the Design Tree pane, expand **counter1(count8\_uniq\_0)** and drag the **countai[7:0]** bus to the Trace Data pane on the right.  
The name of the bus now appears in bold font in the Design Tree pane.
5. Right-click the created trace bus and choose **Rename Trace Bus**. Name the bus **countai**.
6. Select the **Include Trigger Signals in Trace Data** option.
7. Drag the **clk** signal from the Design Tree pane to the Sample Clock box, or type **counter1/clk** in the Sample Clock box.
8. For Buffer Depth box, choose **1024**.
9. Set Data Capture Mode to **Multiple Trigger Capture** and Minimum Samples Per Trigger to **32**.

The Trace Signal Setup tab should now resemble Figure 20.

Figure 20: Trace Signal Setup Tab



## 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. Click the TU1 name in the Name box, and enter **countbi**.
3. Drag the **countbi[7:0]** bus from under **counter1(count8\_uniq\_0)** in the Design Tree pane to the Signals (MSB:LSB) box in the Trigger Unit section.
4. In the Operator box of the trigger unit, choose **<=**.
5. In the Radix box, choose **Hex**.
6. In the Value box, enter **88**.
7. Click **Add** to add a second trigger unit.
8. In the Name box, enter **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, choose **<=**.
11. In the Radix box, choose the default of **Bin**.
12. In the Value box, enter **1**.

## Setting Up the Trigger Expressions

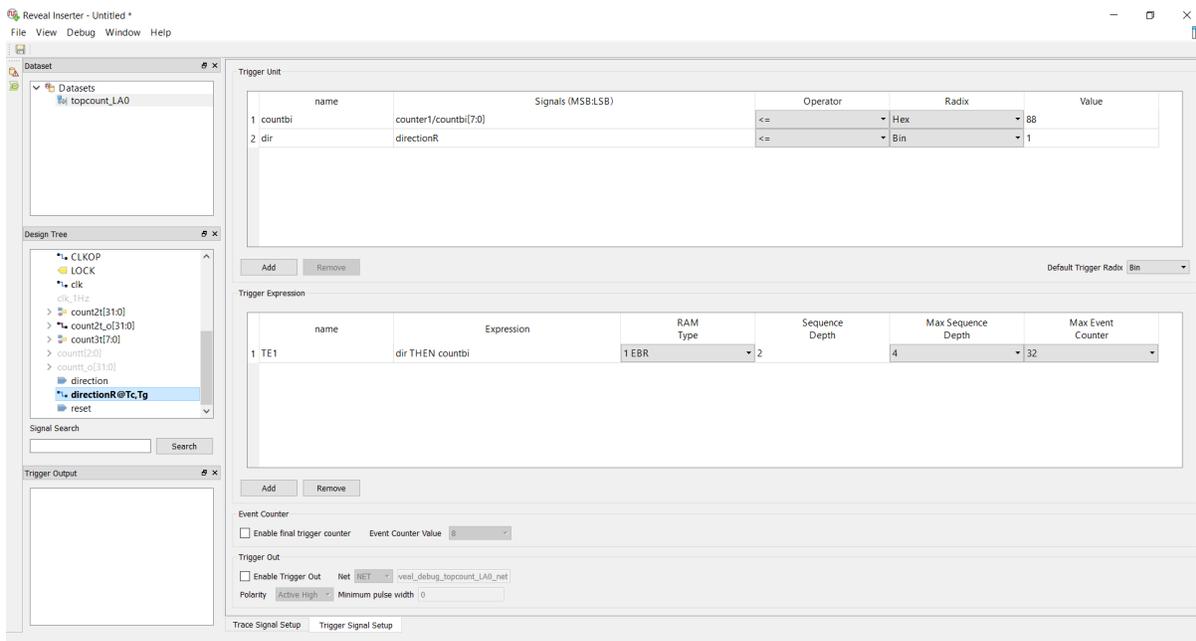
Now you will set up the trigger expressions in the Trigger Expression section of the tab.

### 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, choose **1 EBR**.
4. In the Sequence Depth box, make sure a value of **2** appears.
5. In the Max Sequence Depth box, choose **4**.
6. In the Max Event Counter box, choose **32**.

The Trigger Signal Setup tab should now resemble Figure 21.

**Figure 21: 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**.
2. In the Insert Debug to Design dialog box, make sure that the **Activate Reveal File in Design Project** option is selected.
3. Click **OK**.
4. Name the file `<project_directory>/LEDtest/LEDtest.rvl`.
5. Click **Save**.  
Reveal Inserter imports the Reveal project (.rvl) file into Diamond. The .rvl file is added to the Debug Files in the File List view.
6. Close Reveal Inserter.

## 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 MachXO3L Starter Kit board is connected to your computer with a USB cable.

### Note

If you do not have a MachXO3L Starter Kit, skip the rest of the tutorial. Go to [“Summary of Accomplishments” on page 44](#).

### To generate a bitstream:

1. In the Process view, click **Bitstream File** under Export Files.
2. Click the **Run**  button on the Diamond toolbar.  
Diamond reruns all processes, generating the selected files and saving them in your project directory.
3. Choose **Tools** >  **Programmer**.  
Programmer opens using the LEDtest.xcf file.
4. Click the Program button  on the Programmer toolbar to initiate the download.  
If the programming process succeeds, you will see a green-shaded PASS in the Programmer Status column.
5. Close Programmer.

## Task 15: 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 FPGA. The trigger setup influences under what specific conditions and how the Reveal core trace signal states are displayed in Reveal Logic Analyzer's waveform view. You will explore just a few of the many ways to trigger and trace the system.

This task assumes that the MachXO3L Starter Kit board is connected to your computer with a USB cable.

### 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**.

The Reveal Analyzer Startup Wizard dialog box appears, as shown in Figure 22.

2. In the upper-left of the dialog box, select **Create a new file**.
3. Type **Test** in the box to name the file.
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 computer is detected, and is listed in the USB port box.
6. Click **Scan** to find the FPGA.  
The MachXO3L device on the board is displayed in the Debug device box.
7. In the RVL source box, browse to `<project_directory>/LEDtest/LEDtest.rvl`.

Figure 22: Reveal Analyzer Startup Wizard

**Getting Started:**

Create a new file    Test    HW-USBN-2B (FTDI)     Multiple Device in JTAG Chain

USB port: FTUSB-0    Detect

XCF source:    Browse...

Debug device: 1. LCMXO3L-6900C:0x412BD043    Scan

RVL source: C:/my\_diamond\_tutorial/LEDtest/LEDtest.rvl    Browse...

Open an existing file

File name:    Browse...

Import file into current implementation

OK    Cancel

8. Click **OK**.

The Reveal Logic Analyzer main window now appears with the LA Trigger tab selected, as shown in Figure 23. It contains the same trigger units and trigger expressions that you set up in Reveal Inserter.

Figure 23: Reveal Analyzer

**Trigger Unit**

Name	Signals (MSB:LSB)	Operator	Radix	Value
countbi	counter1/counter1[7:0]	<=	Hex	88
dir	directionR	<=	Bin	1

**Trigger Expression**

Name	Expression	Sequence Depth	Max Sequence Depth	Max Event Counter
<input checked="" type="checkbox"/> TE1	dir THEN countbi	2	4	32

**Trigger Options**

Enable TE: OR All

Samples Per Trigger: 32

Number of Trigger: 1 (max 32)

**Trigger Position: 2/32**

Pre-selected: Pre-Trigger

User-selected: < >

Final Event Counter: 1

9. In the Trigger Options section, for Samples Per Trigger, choose **64**.

10. In the Trigger Position section, choose **Post-Trigger**.

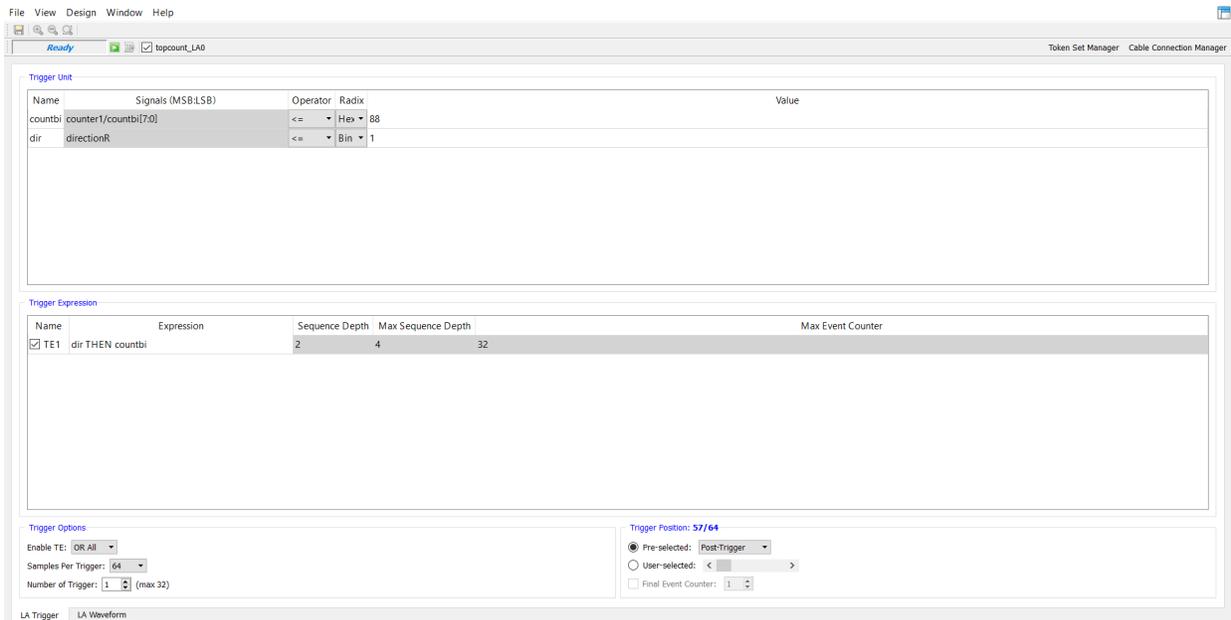
In the Trigger Position section, you can specify the trigger position relative to the trace data. The numbers in the section title show the current position. The two options to choose from include:

- ▶ Pre-selected allows you to choose one of the standard positions.

- ▶ Pre-Trigger: 4/64 of the way from the beginning of the samples.
- ▶ Center-Trigger: 32/64 of the way from the beginning of the samples.
- ▶ Post-Trigger: 57/64 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 24.

**Figure 24: 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.

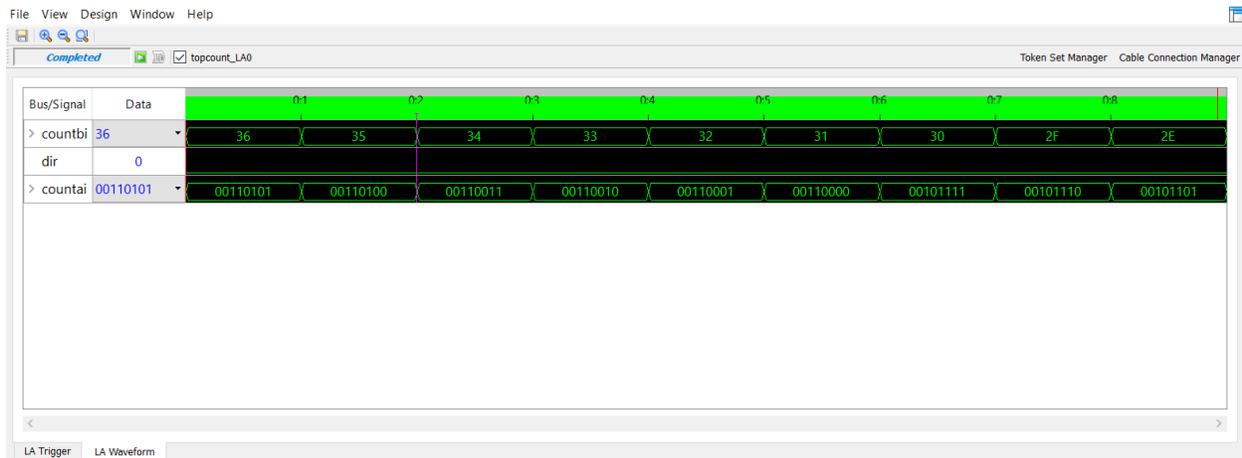
Reveal Analyzer first configures the modules selected for the correct trigger condition, then waits for the trigger conditions to occur. When a trigger occurs, the data is uploaded to your computer. The resulting waveforms appear in the LA Waveform tab.

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 25.

**Figure 25: Reveal Analyzer Waveform**



## Summary of Accomplishments

You have completed the *Lattice Diamond 3.13 Tutorial*. In this tutorial, you have learned how to:

- ▶ Create a new Lattice Diamond project.
- ▶ Create an IPexpress module.
- ▶ Check Hardware Description Language (HDL).
- ▶ Verify functionality with simulation.
- ▶ Inspect strategy settings.
- ▶ Examine resources.
- ▶ Set timing and location assignments.
- ▶ Process the design.
- ▶ Examine static timing analysis results.
- ▶ Analyze power consumption.
- ▶ Run export utility programs.
- ▶ Download a bitstream to an FPGA.
- ▶ 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*.

# Revision History

The following table gives the revision history for this document.

<b>Date</b>	<b>Version</b>	<b>Description</b>
8/17/2023	3.13	Updated to reflect changes in Diamond 3.13.
12/9/2021	3.12	Fixed the screen capture in Figure 20.