Lattice Diamond Tutorial



April 2015

Copyright

Copyright © 2015 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.

Meaning or Use
Items in the user interface that you select or click. Text that you type into the user interface.
Variables in commands, code syntax, and path names.
Press the two keys at the same time.
Code examples. Messages, reports, and prompts from the software.
Omitted material in a line of 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.

Type Conventions Used in This Document



Contents

Lattice Diamond Tutorial 1 Learning Objectives 1 Time to Complete This Tutorial 2 System Requirements 2 Accessing Online Help and Diamond User Guide 2 About the Tutorial Design 3 About the Tutorial Data Flow 3 Task 1: Create a New Lattice Diamond Project 5 Task 2: Create an IPexpress Module 10 Task 3: Verify Functionality with Simulation 13 Task 4: Inspect Strategy Settings 15 Task 5: Examine Resources 17 Task 6: Run Synthesis Process 20 Task 7: Set Timing and Location Assignments 22 Task 8: Running Place and Route 26 Task 9: Examine Post Place and Route Results 29 Task 10: Adjust Static Timing Constraints and Review Results 33 Task 11: Comparing Multiple Place and Route Runs 36 Task 12: Analyze Power Consumption 37 Task 13: Run Export Utility Programs 40 Task 14: Download a Bitstream to an FPGA 40 Task 15: Convert a File Using Deployment Tool 42 Task 16: Use Reveal Inserter to Add On-chip Debug Logic 46 Setting Up the Trigger Units 48 Setting Up the Trigger Expressions 49 Inserting the Debug Logic 50 Generating a Bitstream and Programming the FPGA 52

Task 17: Use Reveal Logic Analyzer to Perform Logic Analysis 53
Creating a New Reveal Logic Analyzer Project 53
Running Logic Analyzer 56
Summary of Accomplishments 57

Recommended References 58



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/latticediamond/. 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/latticediamond#Tab6. 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**.

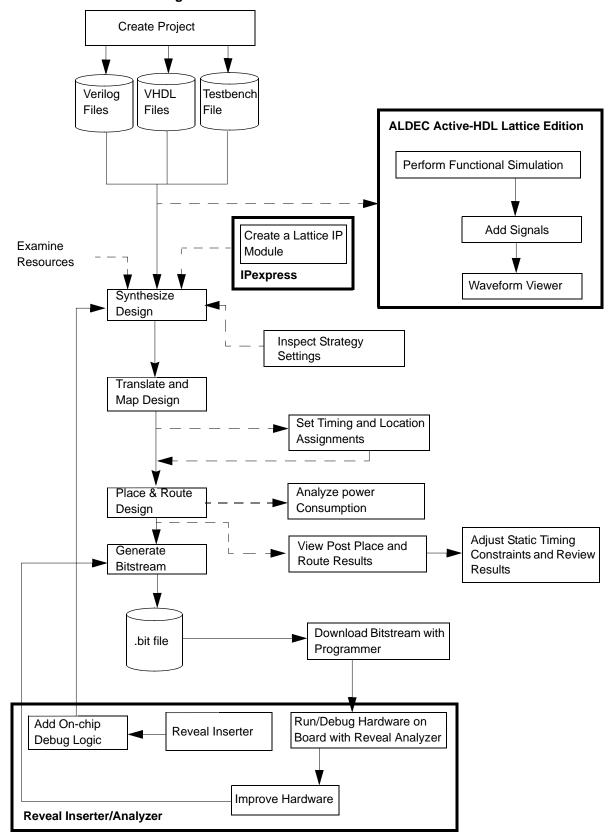
Another excellent resource is the *Diamond User Guide*, available from the start page of Lattice Diamond online help.

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.





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.

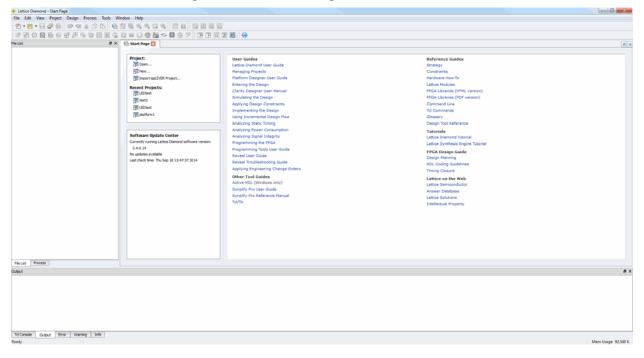
To create a new project:

- 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/lin/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 > Froject.
 - Click the down arrow in the root 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 (_).

- 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

🚸 New Proje	ct 🤋 🗙
	me name for your project and specify a directory where the project es will be stored.
Project:	
Name:	LEDtest
Location:	C:/my_diamond_tutorial
Implementa	tion:
Name:	LEDtest
Location:	C:/my_diamond_tutorial/LEDtest
	< Back Next > Cancel

7. Click Next.

The Add Source dialog box appears.

8. Click Add Source.

The Import File dialog box appears.

- 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

Select Device: Family: LatticeECP		Device:	Device Info Voltage: LUT:	rmation: 1.2V 33264
LatticeECP2 LatticeECP2 LatticeECP3 LatticeSC LatticeSCM LatticeSCM LatticeXP2 MachXO2 MachXO2 Platform Manager Terformance grade:()	E	LFE3-35EA LFE3-70E LFE3-70EA LFE3-95EA LFE3-95EA LFE3-95EA LFE3-150EA	Registers: EBR Bits: EBR Blocks: Dist RAM: DSP: PLL: DLL: PCS: PIO Cells: APIO:	66.75K 16 4 2 1 332 18
8	•	FPBGA484	▼ PIO Pins:	295
Operating conditions: Commercial	•			
Part Names:			•	

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.

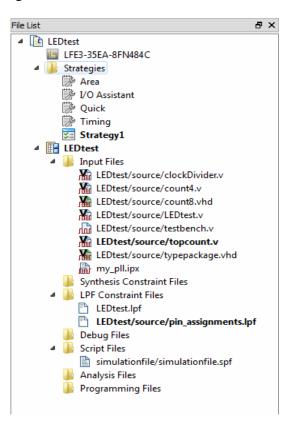
16. Click Finish.

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

The File List view, shown in Figure 5, 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.

Figure 5: File List View



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.

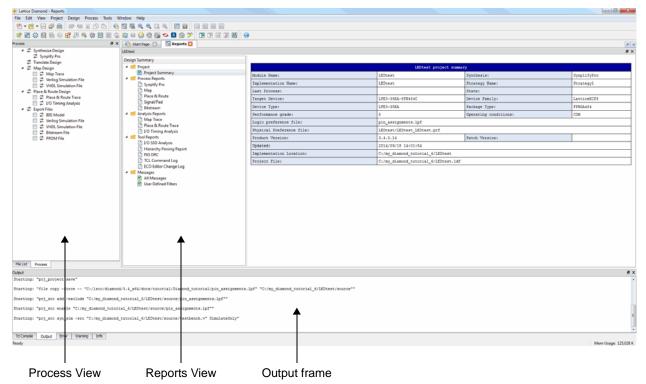
- Navigate to <diamond_install_directory>/docs/tutorial/Diamond_tutorial and select the file pin_assignments.lpf. Choose Copy file to Implementation's Source directory, and click Add.
- 19. In the File List, right-click pin_assignments.lpf and choose **Set as Active Preference File**.
- 20. In the File List, right-click testbench.v and choose Include for > Simulation.

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

H IPexpress	
File Help	
i i 🖳 🍠 🛸 🗇 😫 🛍	
Name	PLL
▲ Module ▲ Module ▲ Architecture_Modules ▲ 한 DDR 월 DDR 월 DDR,GENERIC 월 DDR_MEM 월 DDR_5 월 GDDR_7:1 월 SDR 월 DLL 월 DLL 월 Digital CDR 월 Digital CDR	Macro Type: Module Version: Module Name: PLL Project Path: C:/my_diamond_tutorial Browse File Name: my_pll Module Output: Verilog Device Family: LatticeECP3 Part Name: LFE3-3SEA-SFN484C Synthesis: SynplifyPro
좋 FFB 좋 ORCAstra 좋 PCS 좋 PLL 좋 Power Controller 좋 Power Guard * System_Bus * Tag Memory * Arithmetic_Modules * Adder Subtractor Adder Subtractor Complex_Multiplier Convert Counter FFT_Butterfly	Customize
	Configuration

- 3. In the module tree, under Module > Architechture_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 13.

11. Choose File > Close Window to close IPexpress.

Figure 8: IPexpress Generated Log Tab

Log Errors/Warnings Msg occess: Module occess: ersion Diamond (c) 1991-1994 by NeoCAD Inc. All right (c) 1995 AT&T Corp. All rights resen (c) 1995-2001 Lucent Technologies II (c) 2001-Agere Systems All rights re (c) 2002-2012 Lattice Semiconductor UBA Module Synthesis command : C:\lscc\diamond\ispfpg	ved. nc. All rights reserve served.			
ocess: ersion Diamond (c) 1991-1994 by NeoCAD Inc. All right (c) 1995 AT&T Corp. All rights resen (c) 1995-2001 Lucent Technologies II (c) 2001 Agere Systems All rights re (c) 2002-2012 Lattice Semiconductor :UBA Module Synthesis command : C:\lscc\diamond\jspfpq	ved. nc. All rights reserve served.			
ersion Diamond (c) 1991-1994 by NeoCAD Inc. All right (c) 1995 AT&T Corp. All rights resen (c) 1995-2001 Lucent Technologies in (c) 2001 Agere Systems. All rights re (c) 2002-2012 Lattice Semiconduct (c) 2002-2012 Lattice Semiconduct UBA Module Synthesis command. : C.\lscc\diamond\ispfpq	ved. nc. All rights reserve served.			
(c) 1991-1994 by NeoCAD Inc. All rigit (c) 1995 AT&T Corp. All rights resen- (c) 1995-2001 Lucent Technologies in (c) 2001 Agere Systems. All rights re (c) 2002-2012 Lattice Semiconductor UBA Module Synthesis command. : C.\lscc\diamond\jspfpq	ved. nc. All rights reserve served.			
(c) 1995 AT&T Corp All rights resen (c) 1995-2001 Lucent Technologies In (c) 2007 Agere Systems - All rights re (c) 2002-2012 Lattice Semiconductor :UBA Module Synthesis command :: C:\scc\diamond\;sptpg	ved. nc. All rights reserve served.			
command : C:\lscc\diamond\ispfpg				
command : C:\lscc\diamond\ispfpg				
outs - CLKOP, CLKOK, LOCK r : not inserted tput : suppressed output : my_pllx emplate: my_pllxn ourpose : for synthesis and simulatio tion : not used output : my_pll.srp ad Resource Usage: JBA Module Synthesis		-w -n my_pil Hang verilog	-synth synplity -arch ep5	c00 -type pil -fin 100 -
ss: completed successfully.				
nings: 0				
rs: O				
			01	Help
iutieuiii lata de lui p ∈ r	: uts : CLK tputs : CLKOP, CLKOK, LOCK = : :not inserted utput : :suppressed output : :my_pll.v template :mv, pll tmpl.v	: :CLK tputs :CLKOP, CLKOK, LOCK er :notinsented output :suppressed output :rmy_plix template:my_pli_mplx purpose :for synthesis and simulation ation :notused output :my_plismp led Resource Usage: UBA Module Synthesis oll lpc created. ess: completed successfully. mings: 0	: :CLK tputs :CLKOP, CLKOK, LOCK er :notinserted output :suppressed output :rmy_plix template:my_pli_mplx purpose :for synthesis and simulation ation :not used output :my_pli.srp led Resource Usage: UBA Module Synthesis ell lpc created. ess: completed successfully.	: :CLK tputs :CLKOP, CLKOK, LOCK er :notinserted output :suppressed output :suppressed output :suppressed output :suppressed output :my_plit_mplv purpose :for synthesis and simulation ation :not used output :my_plit_spp led Resource Usage: UBA Module Synthesis plitpic created. ess: completed successfully. mings: 0

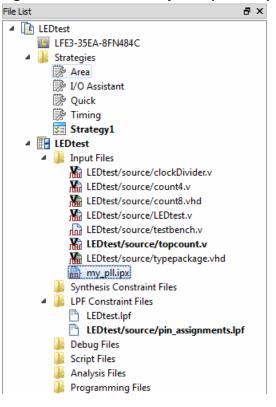


Figure 9: PLL Generated by IP Express Imported into Diamond Project

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 local 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.

- 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 10 is not present in the Source files list, click the Add File button M, 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.

Active-HDL 9.3 (simulationfile ,si	imulationfile) - untitled.awc									- O X
<u>File Edit Search View Works</u>	pace <u>D</u> esign <u>S</u> imulation	Waveform Tool	s <u>W</u> indow <u>H</u> elp							↔ ×
🗗 🕶 🖬 💥 📖 🚳	💱 📃 🔎 🔛 🌫 🔍 📶	I 🐨 🖓 🛅 🙀 🕴	🕲 🔓 🗢 🎒 😓	► ► ► 100	ns 🗄 📢 🔳 🔌	⊳ ९३ ६३ द≋		703ns +	0	
Design Browser ×	📕 🕐 🐁 🗟 🗸	• 🕂 🕅 🏭 🖌		» 🛗 🖓 🕫 🦽	% % % 🖪 🔂	😘 🖿 🕺 🕂 🏒) 🔲 🖬 🏘 💪	a 🗖 🗸	Bold	
testbench 💌	Signal name	Value	80	· · · 160		320		400	480	560 · · · 640 · · · no
Hierarchy	# clk	1	mmm	mmm	mm	mm	mm	www	mmm	11111111111111111111111111111111111111
Theranchy	🖽 🖛 count2	0	×	X	• XX	00000000	OS O CONSCIENCE		DISCO BOARD	
- std.standard	⊞ # count3	08		83						
-P vl.verilog logic	# direction	1								
-P std.TEXTIO	# reset	0								
Piece.std logic 1164										
-P ieee.NUMERIC_STD										
P work.typepackage										E
, <u> </u>										
Name Value										
. E = count3 08										
R= clk 1										
R= reset 0										
R ⁼ direction 1										
	Cursor 1									
	Cursor 1		4							703.1 ▼ ► 4 @ ₩
· · · · · ·										wave.asdb SIM
Files / Stru/@Res/	🙀 untitled.awc									Wave.asub Silvi
# 5 signal(s) trace	d.									*
<pre>* • run 1000ns • # Waveform file 'un</pre>										
• # RUNTIME: Info: RU				_6/Simulation	ifile/src/wave	.asdp'.				
<pre># KERNEL: Time: 703</pre>	ns, Iteration: 0,	Instance: /	testbench, Prod	cess: @INITIAI	1#24 20.					
# KERNEL: stopped a	t time: 703 ns				-					-
>										
J Console										
1000000 A										NUM INS //

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.

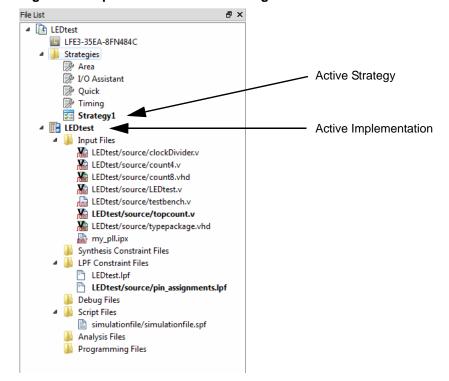


Figure 11: Implementations and Strategies

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

Proce	is	Synp	lify P	ro	
4 🧧	Synthesize Design		catalog: A		fault
_	🔄 Precision	Name	Туре	Value	
*		Allow Duplicate Modules	T/F	False	
⊿ 🌾	Map Design	Area	T/F	False	
	Map Trace	Arrange VHDL Files	T/F	True	
4 3	Place & Route Design	Clock Conversion	T/F	True	
	IO Timing Analysis	Command Line Options	Text		
* -	San IO Timing Analysis San Timing Simulation San Bitstream	Default Enum Encoding	List	Default	
~		Disable IO Insertion	T/F	False	
V		Export Diamond Settings to Synplify Pro GUI	List	No	
		FSM Encoding	T/F	True	
		Fanout Guide	Num	1000	
		Force GSR	List	False	
		Frequency (MHz)	Num		l
		Number of Critical Paths	Num	10	
		Number of Start/End Points	Num		
		Output Netlist Format	List	None	
		Pipelining and Retiming	List	Pipelining Only	
		Push Tristates	T/F	True	
		n i w in:	T/F		
		Specifies the number of critical timing paths to be	reported in	the timing report.	

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.

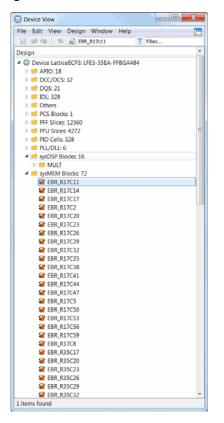
Several folders organized by feature type appear.

- 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.

) De	evice V	iew 👄	Contrast of		-		×
				Window	Help		
8	10 0	1 35	BR EBR	_R17c11		Tilter	_
Desig	gn 🛛						^
•	Devid	e Lattic	eECP3: LI	E3-35EA-F	PBGA	484	
		PIO: 18					
		CC/DCS	: 32				
	= IO						
		ners S Block	w- 1				
			: 12360				
		U Slice					Ξ
		O Cells:					
		L/DLL:					
4	🛋 sy	SDSP BI	ocks: 16				
	Þ 📕	MULT					
4			Blocks: 72				
		EBR_R		Prohibit			
		EBR_R		Release			
		EBR_R		Show in	•	Floorplan View	
		EBR_R EBR_R			-	Physical View	
		EBR				Thysical field	
		EBRR					
		EBR_R					
		EBRR	17C32				
	- 22	EBR_R	17C35				
	- 22	EBR_R	17C38				
		EBR_R					
		EBR_R					
		EBR_R					
		EBR_R					
		EBR_R EBR_R					
		EBR_R					
		EBRR					
		EBR_R					
		EBR_R					
		EBR_R	35C20				
	- 22	EBR_R	35C23				
		EBR_R					
		EBR_R					
	- 22	EBR_R	35C32				Ŧ

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.

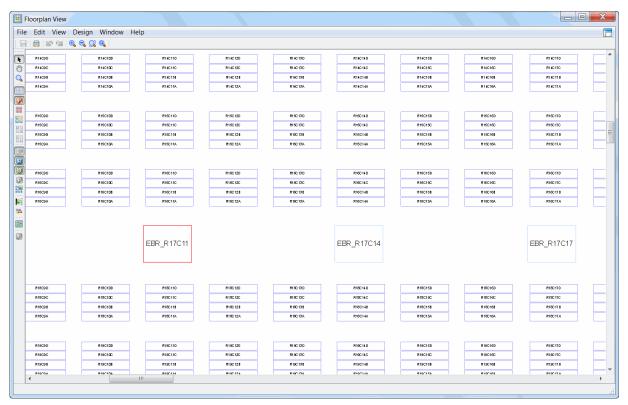


Figure 15: Floorplan View

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 Synopsis 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 thirdparty 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 \swarrow indicates success; a yellow triangle \gtrsim indicates success with warnings; a red X \approx 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

erarchyPost Map Resources				đΧ
Unit	File	LUT4	PFU Registers	
🛛 🌆 topcount	C:/lscc/diamond/docs/tutorial/Diamond_tutorial/LED_test/source/topcount.v	73(8)	102(1)	27(
my_pll_uniq_2(my_pll_inst)	C:/Iscc/diamond,/docs/tutorial/Diamond_tutorial/my_pll.v	1(1)	0(0)	0(0
LEDtest_uniq_2(my_LEDtest)	C:/Iscc/diamond/docs/tutorial/Diamond_tutorial/LED_test/source/LEDtest.v	15(15)	0(0)	0(0
count8_uniq_1(counter1)	C:/Iscc/diamond/docs/tutorial/Diamond_tutorial/LED_test/source/count8.vhd	1(1)	16(16)	0(0
count4_uniq_4(counter3)	C:/lscc/diamond,/docs/tutorial/Diamond_tutorial/LED_test/source/count4.v	6(6)	12(12)	0(0
count4_uniq_3(counter2)	C:/Iscc/diamond/docs/tutorial/Diamond_tutorial/LED_test/source/count4.v	5(5)	8(8)	0(0
clockDivider_uniq_2(clockDivider_inst)	$\label{eq:c:lscc/diamond/docs/tutorial/Diamond_tutorial/LED_test/source/clockDivider.v} C:/lscc/diamond/docs/tutorial/Diamond_tutorial/LED_test/source/clockDivider.v$	37(37)	65(65)	0(0
1	III			Þ
File List Process HierarchyPost Map Resources				

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 73(8) 73 represents the total LUT4 count utilization throughout the design and 8 represents the LUT4 utilized only in the design module topcount.
- PFU Registers 102(1) 102 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_2, LEDtest_uniq_2, count8_uniq_1, count4_uniq_4, count4_uniq_3 and clockDivider_uniq_2 are the sub-modules (instances) of the design. For example, the sub-module count4_uniq_3, LUT4 5(5) represents the total LUT4 count in the sub-module count4_uniq_3.
- PFU Registers 8(8) represents the total PFU Registers in the sub-module count4_uniq_3.
- SLICE 5(5) represents the total number of SLICEs in the sub-module count4_uniq_3.

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 backend 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 appears 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 <a>I. If a report has been generated, the icon appears as <a>I. If the report is not the most recent version, the icon appears as <a>I. 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 <a>I 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.

		In successful David	
ile Edit View Project Design Process		Incremental Desig	gn Flo
	🗈 🖽 🖪 🥞 🧠 🔍 🔍 📗 🛛		
19 🖻 🖨 🚝 🔡 👫 🕂 👒 🕸 🖻 🖄 🔏] 🗄 🤮 🔇 🗢 🎇 🎲 🈕 🗍		
rocess &	K 🚯 Start Page 🗵 🛄 Re	ports 🗵	5
4 🛠 Synthesize Design	verilog_vhdl_edif		83
🛠 Synplify Pro		Lattice Mapping Report File for Design Module 'count16'	
💜 Translate Design	Design Summary	Lattice Mapping Report File for Design Module 'Countie'	
🔺 💜 Map Design	Project	Dealers To Grouphlan	8
📝 💜 Map Trace	Project Summary	Design Information	
📝 왷 Verilog Simulation File	Process Reports	Command line: map -inc -a LatticeECP3 -p LFE3-35EA -t FPBGA484 -s 8 -oc	
🔲 🥏 VHDL Simulation File	Synplify Pro	Commercial mixedcounter_verilog_vhdl_edif.ngd -icf mixedcounter verilog vhdl edif.icf -o	
a 🕏 Place & Route Design	🖻 🞽 Map	mixedcounter_verilog_vhdl_edif_map.ncd -pr	
📝 🥏 Place & Route Trace	Place & Route	mixedcounter_verilog_vhdl_edif.prf -mp mixedcounter_verilog_vhdl_edif.mrp	
📝 🦈 I/O Timing Analysis	Signal/Pad	C:/Diamond_tut/mixedcounter.lpf Target Vendor: LATTICE	
🔺 🥏 Export Files	Bitstream	Target Device: LFE3-35EAFPBGA484	
🔲 🥏 IBIS Model	🔺 💻 Analysis Reports	Target Performance: 8 Mapper: ep5c00, version: Diamond Version	
Verilog Simulation File	🖻 🖹 Map Trace	Mapped on:	
🗐 🥏 VHDL Simulation File	Place & Route Tra		
📝 🥏 Bitstream File	I/O Timing Analysis	Design Summary	
🔲 🥏 PROM File	🔺 💻 Tool Reports	Number of registers: 48	
	I/O SSO Analysis	PFU registers: 32 PIO registers: 16	
	Generate Hierarchy	Number of SLICEs: 18 out of 16632 (0%)	
	2 Run BKM Check	SLICEs(logic/ROM): 18 out of 13428 (0%) SLICEs(logic/ROM/RAM): 0 out of 3204 (0%)	
	PIO DRC	As RAM: 0 out of 3204 (0%)	
		As Logic/ROM: 0 out of 3204 (0%) Number of logic LUT4s: 1	
		Number of distributed RAM: 0 (0 LUT4s)	
		Number of ripple logic: 9 (18 LUT4s)	
		Number of shift registers: 0 Total number of LUT4s: 19	
		Number of PIO sites used: 19 out of 295 (6%)	
		Number of PIO FIXEDDELAY: 0 Number of PCS (SerDes): 0 out of 1 (0%) with bonded PIO sites	
		Number of DQSDLLs: 0 out of 2 (0%)	
		Number of PLLs: 0 out of 4 (0%) Number of DLLs: 0 out of 2 (0%)	
		Number of block RAMs. 0 out of 72 (0%)	
File List Process	× Find: design summary	Number of block BEMS: 0 out of 72 (08)	-
File List Process	× Find: design summary		5

Figure 17: MAP Report

6. Choose Tools > Spreadsheet View, or click 2.

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.

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

The Spreadsheet View is detached from the Diamond main window.

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

The PERIOD/FREQUENCY Preference dialog box appears.

9. Enter the following preference settings:

Type:FREQUENCYSecond Type:NetAvailable 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 don the Spreadsheet View toolbar.

The INPUT_SETUP/CLOCK_TO_OUT Preference dialog box appears.

12. Enter the following preference settings:

Туре:	INPUT_SETUP
Second Type:	All Ports
Clock Ports/Nets:	clk
Time:	10ns

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:

Туре:	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 rightclick menu of the Spreadsheet View. You can also double click on a value in Timing Preferences tab and edit the value directly.

🎸 Sp	preadsheet View *		
File	Edit View Design Window Help		
H	🚍 🗶 🗅 🗈 📽 🐿		
ø	Preference Name	Preference Value	Preference Unit
nî	BLOCK		
	FREQUENCY NET "CLKOK c"		
\$	Frequency	50.000000	MHz
	Hold Margin	Auto	ns
ಕೆ	PAR ADJ	0.000000	115
	▲ NET "CLKOP c"	0.000000	
2	Frequency	500.000000	MHz
	Hold Margin	Auto	ns
where a	PAR ADJ	0.000000	
4	▲ NET "clk_c"		
1	Frequency	100.000000	MHz
	Hold Margin	0.000000	ns
⊿	PAR_ADJ	0.000000	
	PERIOD		
	▲ INPUT_SETUP		
₽J	ALLPORTS CLKPORT "clk"		
*	Time Hold Time	10.000000	ns
_	Clock Offset	Auto 0.000000	ns
p	PLL Phase Back	OFF	
	Input Delay	not Input Delay	
	Src Syn	OFF	
9	PORT "reset" CLKPORT "clk"	011	
	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		
	MULTICYCLE		
	MAXDELAY		
	CLKSKEWDIFF		
1	rce Route Priority Cell Mapping Globi	al Preferences Tir	ning Preferences

--+ \/:

- 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 LVCMOS33, 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.

Physical View			
	w <u>H</u> elp		[
0, 0, 0, 0,			
1			
3			
			· · · · · · · · · · · · · · · · · · ·
<u> </u>			
	in a second s		
►	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	<u> </u>
	······································	<u></u>	
	<mark>le li le l</mark> e le		
	· · · · · · · · · · · · · · · · · · ·		
A at m			
		x,y:(139029, 109632)	

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 ^Q button.

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

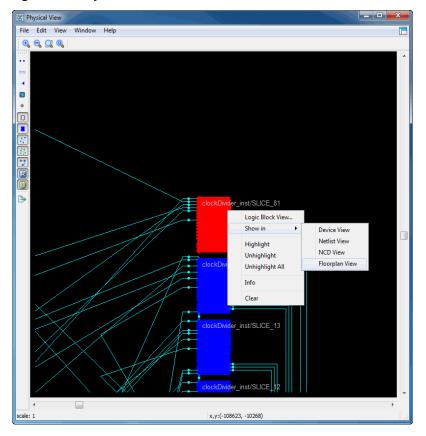


Figure 20: Physical View

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.

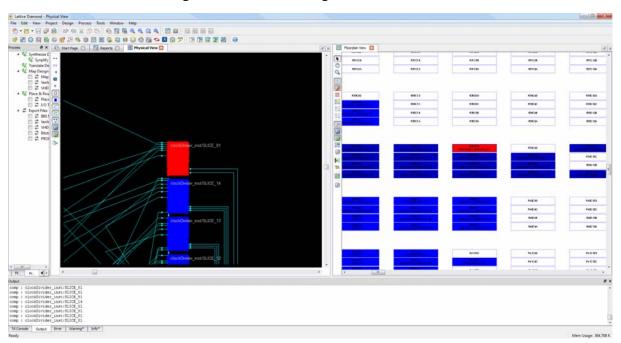


Figure 21: Cross Probing

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
heightain the Diamond toolbar.

The Timing Analysis view appears.

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.

3. From the Preference Reports tab (on the lower left of the Timing Analysis view), select INPUT_SETUP ALLPORTS 10ns 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.

4. Select Row 1 of the Path Table.

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

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

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

6. 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

Setup Performance Grade Current: Default Current: Default Current: Defau				
Check unconstrained connections Check unconstrained paths Report asynchronous timing loops Full name Report style: Verbose Timing Report Worst-Case paths: 20 +	Setup Performance	Grade	Hold Performance G	Grade
Check unconstrained paths Check unconstrained paths Report asynchronous timing loops Full name Report style: Verbose Timing Report Worst-Case paths:	Current:	Default 💌	Current:	Default 🔻
 □ Report asynchronous timing loops □ Full name Report style: Verbose Timing Report ▼ Worst-Case paths: 	Check unconstrai	ned connections		
Full name Report style: Verbose Timing Report ▼ Worst-Case paths: 20 +	Check unconstrai	ned paths		
Report style: Verbose Timing Report 💌	Report asynchror	nous timing loops		
Worst-Case paths:	Eull name			
	rannane			
Number of unconstrained paths: 0		Verbose Timing Report 🔻		
	Report style:			
	Report style: Worst-Case paths:	20		

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

The Detailed Path Tables are updated.

9. 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.

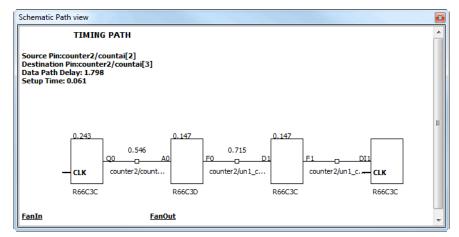
Edit View Window	Help												
	alues		Path Table - "FR	EQUENCY N	ET "dk_c" 100.00	0000 MHz * ((setup)						
	atticeECP3 FE3-35EA												
	PE3-35EA PBGA484		Source Filt	er:				Destination filter	1			📝 C	ase sensiti
Setup Performance G D	efault												
Hold Performance Gr D Check Unconstrained N	efault			Source		ination	Weighted Slack		Required	Data Delay	Route %	Levels	Clock
Check Unconstrained N				2/countai[2			8.141	1.859	10	1.798	70.1	3	0
ReportAsynchronous N			2 counter				8.35	1.65	10	1.589	66.2	3	0
Report Style Vi Full Name N	erbose Timing Report		3 counter				8.986	1.014	10	0.953	59.1	2	0
Worst-Case Paths 2			4 counter				8.986	1.014	10	0.953	59.1	2	0
Number of Unconstrai 0			5 counter				9.011	0.989	10	0.928	58.0	2	0
			6 direction		counter2/c	ountai(3)	9.013	0.987	10	2.918	81.6	3	-0.819
				2/countai(2			9.016	0.984	10	0.923	57.7	2	0
			8 counter	2/countai[3] counter2/c	ountai(3)	9.029	0.971	10	0.91	57.1	2	0
			9 direction	nR	counter2/c	ountai(2)	9.108	0.892	10	2.823	86.2	2	-0.819
			10 counter	2/countai(2	counter2/c	ountbi[2]	9.125	0.875	10	0.741	67.2	1	0
			11 counter	2/countai(3	counter2/c	ountbi(3)	9.125	0.875	10	0.741	67.2	1	0
			12 counter	2/countai(1] counter2/c	ountai(1)	9.196	0.804	10	0.743	47.5	2	0
Preference Name	*	Analysis Type	13 counter	2/countai(1] counter2/c	ountai(2)	9.212	0.788	10	0.727	46.4	2	0
Analysis Results		Analysis type	14 direction	ıR	counter2/c	ountai(1)	9.228	0.772	10	2.703	85.6	2	-0.819
	IET "CLKOK_c" 50.000000 MHz	setup	15 counter	2/countbil[2] counter2_c	countio[2]	9.276	0.724	10	0.843	71.2	1	-0.114
	IET "CLKOK_c" 50.000000 MHz	hold	16 counter	2/countbil0	counter2_0	countio[0]	9.276	0.724	10	0.843	71.2	1	-0.114
	IET "CLKOP_c" 500.000000 MHz	setup	17 counter	2/countbi(3] counter2_c	countio[3]	9.276	0.724	10	0.843	71.2	1	-0.114
	IET "CLKOP_c" 500.000000 MHz IET "dk.c" 100.000000 MHz	hold setup	18 counter	2/countbi[1] counter2_c	countio[1]	9.276	0.724	10	0.843	71.2	1	-0.114
	IET "dk_c" 100.000000 MHz	hold	19 counter	2/countai(0] counter2/c	ountbi(0)	9.293	0.707	10	0.573	57.6	1	0
	ALLPORTS 10.000000 ns CLKPORT "dk"	setup	20 counter	2/countai(1] counter2/c	ountbi(1)	9.298	0.702	10	0.568	57.2	1	0
	ALLPORTS 10.000000 ns CLKPORT "dk"	hold											
	ALLPORTS 10.000000 ns CLKPORT "clk" PORT "reset" 10.000000 ns CLKPORT "clk"	hold_window setup	•										
	PORT "reset" 10.000000 ns CLKPORT "clk"	hold	Detailed Path Ta	bles									
NPUT_SETUP	PORT "reset" 10.000000 ns CLKPORT "clk"	hold_window	Data Path De	als Sou	rce Clock Des	tination Clo	dk .						
			Name	Fanout	t Delay	Source	Destination		Resource				
			REG_DEL	0	0.243	R66	R66C3C.Q0	counter2/SLICE	_51				
			ROUTE	3	0.546	R66	R66C3D.A0	counter2/count	ai(2)				
			CTOF_DEL	0	0.147	R66	R66C3D.F0	counter2/SLICE	_94				
			ROUTE	1	0.715	R66	R66C3C.D1	counter2/un1_c	ountai_axbxc3_	1			
			CTOF_DEL	0	0.147	R66	R66C3C.F1	counter2/SLICE	51				

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.





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** Mutton in the Timing Analysis view.

The TPF Spreadsheet View appears.

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

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

13. Enter the following preference settings:

Туре:	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

Type PERIOD FREQUENCY	Available Clock F	Ports:	
Second Type			
None			
Net			
Port	T Filter		
	Frequency:	200	MH
	Frequency: Hold margin:	200	MH: ns
		200	

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.
- 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.

Figure 26: System Jitter

Edit View Design Window Hel	p
⊜ X B î e ≊	
Preference Name	Preference Value
Junction Temperature (Tj)(C)	85.000
Voltage (V)	1.140
SYSTEM_JITTER(ns)	0.100
Block Path	
BlockAsynchpaths	ON
Block Resetpaths	ON
Block RD During WR Paths	OFF
Block InterClock Domain Paths	OFF
Block Jitter	OFF
▲ sysConfig	
PERSISTENT	OFF
CONFIG_MODE	SPI
DONE_OD	ON
DONE_EX	OFF
MCCLK_FREQ	2.5
CONFIG_SECURE	OFF
WAKE_UP	21
WAKE_ON_LOCK	OFF
COMPRESS_CONFIG	OFF
ENABLE_NDR	OFF 2.5
CONFIG_IOVOLTAGE STRTUP	2.5 EXTERNAL
⊿ User Code	EXTERNAL
UserCode Format	Binary
UserCode	000000000000000000000000000000000000000
Global Set/Reset Net	000000000000000000000000000000000000000
Giobal Genteset Net	
<	
nments Clock Resource Route Priorit	y Cell Mapping Global Preferences

19. Close Spreadsheet View. In the Confirm dialog box, click **Discard** to discard the change.

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 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 ^(G) 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 CLKPORT "clk" Setup**, right-click and choose **TPF Preferences**.

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

- 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

le	preadsheet View - TPF * Edit View Window Help		
	Preference Name	Preference Value	Preference Unit
r: n			
£	✓ NET "clk c"		
ø	Frequency	100.000000	MHz
2	Hold Margin	0.000000	ns
ŝ	PAR_ADJ	0.000000	
-	▲ NET "CLKOK_c"	50,000000	10.15
•	Frequency Hold Margin	50.000000 Auto	MHz ns
	PAR ADJ	0.000000	115
	NET "CLKOP_c"		
	Frequency	500.000000	MHz
	Hold Margin	Auto	ns
	PAR_ADJ	0.000000	
	PERIOD INPUT SETUP		
	ALLPORTS CLKPORT "clk"		
	Time	15.000000	ns
	Hold Time	Auto	ns
	Clock Offset	0.000000	
	PLL Phase Back	OFF	
	Input Delay	not Input Delay	
	Src Syn PORT "reset" CLKPORT "clk"	OFF	
	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 MULTICYCLE		
	MAXDELAY		
	CLKSKEWDIFF		
	< []		
1	Global Preferences Timing Preferences	Group	

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

The Update button 📓 on the toolbar is now rotating.

7. 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 inmemory 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:

- 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.
- 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.

Run Manager							l	
ile Window								Ī
0 implementations are hidden. <u>Click to con</u>	figure visible implem	entations						
Implementation <strategy></strategy>	Current Step	Status	Next Step	Timing Score	Slice	Unrouted Net	Run Time	Level/Cost
LEDtest < Strategy1 >	N/A	Completed	N/A	0	756/16632	0	00:01:22	5_1
verilog_vhdl <strategy1></strategy1>	Synthesis	Running	RevealCore	N/A	N/A	N/A	N/A	N/A
3								
*								

Figure 28: Run Manager

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 the Maximum number of the Maximum number of the Box in front of the Maximum number in the box in front of the Maximum number of the Same number in the box in front of the Maximum number of the Box in front of the Maximum number of the Box in front box in front of the Box in front of the Box in front box in front of the Box in front of the Box in front box in front of the Box in front box in fro

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.

3 🖨 🔍								
ttice Po	wer Ca	lculator					Software Mode:	Calculation
ver Summary	Logic Bl	ock Clocks	I/O I/O	Term Block RAM	DSP PLL	DLL DQSDLL	SERDES Graph Re	eport
evice					Enviror	iment		
amily:	LatticeECP3	▼ Per	formance grade	: 8	•		Thermal Profile	
evice:	LFE3-35EA	• Op	erating condition	s: Commercial	- Ambie	nt Temperature(°C):	25	
ackage type:	FPBGA484	▼ Par	t Names:	LFE3-35EA-8FN484C	• Effectiv	/e Theta-JA:	8.79	
evice Power F	arameters				Junctio	n Temperature(°C):	26.23	
rocess Type:	Typical	•	Power File Revis	sion: Final	Maxim	um Safe Ambient(°C):	82.85	
Voltage/Dyna	mic Power M	ultiplier	Current by Po	wer Supply	Power by Pow	er Supply	Power by Block (W)	Peak Sta 🔍 🕨
	Voltage	DPM	Static (A)	Dynamic (A) Ital (,	Static (W)	Dynamic (W) :al (Logic Block	0.0140
Vcc	1.200	1.00	0.0207	0.0000 0.0	0.0248	0.0000 0	Clocks	0.0014
Vccaux	3.300	1.00	0.0165	0.0000 0.0	0.0545	0.0000 0	I/O	0.0893
Vccj	1.200	1.00	0.0025	0.0000 0.0	0.0030	0.0000 0	Block RAM	0.0030
Vccpll	3.300	1.00	0.0046	0.0000 0.0	0.0150	0.0000 0	DSP	0.0011
Vccio 3.3	3.300	1.00	0.0079	0.0000 0.0	0.0259	0.0000 0	PLL	0.0150
Vccio 2.5	2.500	1.00	0.0000	0.0000 0.0	0.0000	0.0000 0	DLL	0.0000
Vccio 1.8	1.800	1.00	0.0000	0.0000 0.0	0.0000	0.0000 0	DQSDLL	0.0000
Vccio 1.5	1.500	1.00	0.0000	0.0000 0.0	0.0000	0.0000 0	SERDES	0.0084
Vccio 1.2	1.200	1.00	0.0079	0.0000 0.0	0.0095	0.0000 0	Misc	0.0077
Vccib	1.200	1.00	0.0000	0.0000 0.0	0.0000	0.0000 0	Total	0.1400
	1.200	1.00	0.0000	0.0000 0.0	0.0000	0.0000 0		
Vccob	1.200	1.00	0.0060	0.0000 0.0	0.0072			
Vccob Vcca			0.0660	0.0000.0660	0.1400	0.00001400		

Figure 29: Power Calculator

- In the Device Power Parameters section select the following parameter: Process Type: 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, select the following parameter:

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. 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 **D** 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.
- 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 ¹/₂ 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.

- Ensure that the proper device is selected by clicking the cell in the Device column and selecting LFE3-35EA from the drop-down menu.
- 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

LatticeECP3 - LFE3-35	EA - Device Properties
Device Operation	
Access Mode:	JTAG 1532 Mode
Operation:	Fast Program 🔹
Programming Options Programming File: :/r	my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit
Device Options	
Reinitialize part on p	rogram error
	OK Cancel

- 8. Click OK.
- 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 <a> 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:

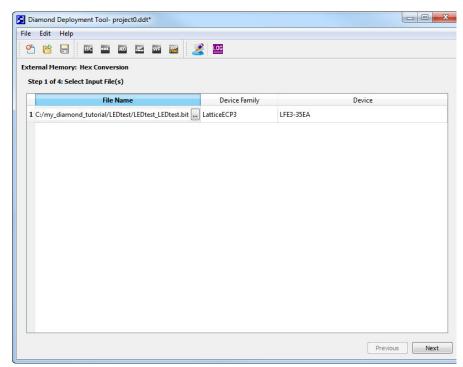
- Choose Programs > Lattice Diamond <version_number> > Accessories > Deployment Tool from the Windows Start menu.
- 2. In the Diamond Deployment Tool Getting Started dialog box, choose **Create New Deployment**.
 - a. In the Function Type dropdown, choose External Memory.
 - b. In the Output File Type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

Figure 31: Deployment Tool Getting Started Dialog Box

Create New I	Deployment
Function Type:	External Memory 🔹
Output File Type:	Hex Conversion 💌
🔘 Open an Exis	ting Deployment
Recent Files:	· · · · · · · · · · · · · · · · · · ·

- 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

Diamond Deployment	ool- project0.ddt*	
ile Edit Help		
20 🔂 🚰	🚥 📧 🚾 🟙 🛃 🔟	
xternal Memory: Hex (onversion	
Step 2 of 4: Hex Conve	rsion Options	
Ouput Format:	Intel Hex 🔻	
Program Security Bit:	Default 🔻	
Verify ID Code:	Default 🔻	
Frequency:	Default 🔻	
Compression:	Default 👻	
CRC Calculation:	Default 🔹	
Starting Address:	0x000000 -	
Byte Wide Bit Mirror		
🔲 Retain Bitstream Heade	r -	
		Previous Next

- 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.

Diamond Deploy	ment Tool- project0.ddt*	
ile Edit Help		
🕈 🖻 🔒	ISC 🚥 JED 🛲 SVF 🌃 🏄 LOG	
xternal Memory:	Hex Conversion	
Step 3 of 4: Sele	ct Output File(s)	
Output File 1:	C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs	•
ouput ne 1.	e.my_damond_adonal/ecorest/ecorest_ecorestances	•
		Previous Next
		INEXIOUS

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.

ile Edit Help	
🗠 💽 🔤 🔤 🔤 🗠	
xternal Memory: Hex Conversion	
Step 4 of 4: Generate Deployment	
	View File
Deployment Tool Summary	2
Input File: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit	
Options:	=
Output Format: Intel Hex	
Program Security Bit: Default	
Verify ID Code: Default Frequency: Default	
Compression: Default	
CRC Calculation: Default	
"C: //scc /diamond/bin /nt /ddtcmd" -oft -int -dev LEE3-35EA -if "C: /mv_diamond_tutorial//EDtest/A	EDtest LEDtest hit" -address 0x000000 -of
"C://scc/diamond/bin/nt/ddtcmd" -oft -int -dev LFE3-35EA -if "C://my_diamond_tutorial/LEDtest/L "C://my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs" Reading Input File: C:/my_diamond_tutorial/LEDtest/LEDtest/LEDtest_LEDtest.bit Format: Single Intel Hex Output File: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs Starting address: 0x000000 Generating Intel Hex Processing file C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit Fromats: 0x000000 Generating Intel Hex Processing file C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest_LEDtest.bit File C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit processfully. The G: \my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit processfully.	EDtest_LEDtest.bit* -address 0x000000 -of
C:/my_damond_tutorial/LEDtest/LEDtest_LEDtest.mcs Reading Input File: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit Format: Single Intel Hex Output File: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs Starting address: 0x00000 Generating Intel Hex Processing file C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest_bit File C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit processed successfully. The file C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs was generated successfully.	-
C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs Reading Input File: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit Format: Single Intel Hex Output File: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.mcs Starting address: 0x000000 Generating Intel Hex Processing file: C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit File C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit processed successfully.	-

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.



View Debug Window Help Dataset 6 ×	Trace	Ī
Dataset 🗗 🗸	M Tare	
Dataset B' X	Trace	
A 🏝 Datasata	WR Trace	
 Udases topcount_LA0 		
Design Tree 🛛 🗗 🗙		
At topcount IdcxLDivider_inst(clockDivide		
Signal Search	Sample Clock Implementation EBR	
Search	Buffer Depth 256 Timestamp 9 Bits	
Trigger Output & X	Sample Enable Data Capture Mode Sample Enable Active High Multiple Trigger Capture	
	POR Debug Minimum samples per trigger * Trigger Enable Active High * Number of triggers for POR 1 *	
	Indude trigger signals in trace data	
	Trace Signal Setup Trigger Signal Setup	

- 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.

🥵 Reveal Inserter - C:/my_diamond_tutorial/LEDtest/LED	test.rvl	
File View Debug Window Help		—
Dataset 8 ×		
Design Tree Image: X Image: X Image:		
Signal Search Trigger Output	Sample Clock dk Implementation EBR ▼ 5 EBRs Buffer Depth 4096 Timestamp 13 ♥ Bits Sample Enable Data Capture Mode Single Trigger Capture POR Debug Minimum samples per trigger 32 ♥ Trigger Enable Active High ♥ Number of triggers for POR 1 ♥ V Indude trigger signals in trace data Trace Signal Setup Trigger Signal Setup	

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.
- 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 should now resemble Figure 38.

Reveal Inserter - Untitled *									
File View Debug Window Help									
Dataset B X	Trigger Unit								
topcount_LA0	name	Signals (MSB:LSB)	Operator	Radix	Value				
	1 countbi counter1/cou	ntbi[7:0]	<= ▼	Hex 🔻	88				
	2 dir directionR		<= ▼	Bin 🔻	1				
Design Tree	Add Remove Trigger Expression			Default Trigger R	adx Bin 🔹				
 counter2(count4_uniq_1) counter3(count4_uniq_2) 	name Expre	Max Sequence Depth	Max Event Counter						
 my_LEDtest(LEDtest_uniq_1) my_pll_inst(my_pll_uniq_1) 	1 TE1 dir THEN cou	ntbi 1 EBR 🔻	Depth	· · · · · · · · · · · · · · · · · · ·	32 •				
CLKOK CLKOP Signal Search									
Search									
Trigger Output & X	Add Remove								
	Event Counter								
	Enable final trigger counter	Event Counter Value 8	•						
	Trigger Out								
	Enable Trigger Out Net	ET 🔻 reveal_debug_topcoun	t_LA0_net						
	Polarity Active High 💌 Minimu	n pulse width 0							
	Trace Signal Setup Trigger Signal Setu	p							
					h.				

Figure 38: Trigger Signal Setup Tap

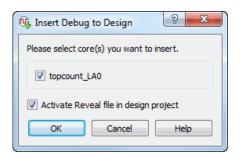
Inserting the Debug Logic

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

To insert the debug logic:

- 1. Choose **Debug > Insert Debug** or click p.
- 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

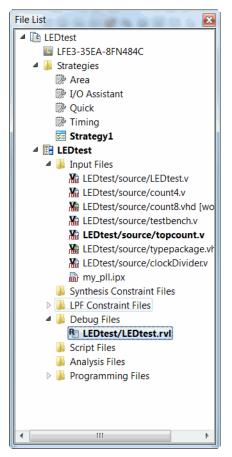


- 3. Click OK. Name the file <project_directory>/LEDtest/LEDtest.rvl.
- 4. Click Save.

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 **D** 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 Methods 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.
- 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 <a>[*] 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:

 In the Diamond main window, choose Tools > Reveal Analyzer or click the is 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.

- 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.

Create a new file	Test	HW-USBN-2B (FTDI)
e a cale a nen ne	Multiple Device in JTAG Chain	
USB port:	FTUSB-0	▼ Detect
XCF source:		Browse
Debug device:	UELFE3-35EA: 16851011	▼ Scan
RVL source:	C:/my_diamond_tutorial/LEDtest.rvl	Browse
Open an existing file	2	
File name:		Browse

Figure 41: Reveal Analyzer Startup Wizard

8. 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.

View	Design Window Help	
	eady	
Trigger Un	it	
Name	Signals (MSB:LSB)	Operator Radix Value
countbi	counter1/countbi[7:0]	<= • He • 88
dir	directionR	<= • Bin • 1
Trigger Ex	pression	
Name	Expression	Sequence Depth Max Sequence Depth Max Event Counter
V TE1	dir THEN countbi	2 4 32
	otions	Trigger Position: 2/32 Pre-selected: Pre-Trigger
Enable TE:		
Samples Pe	OR All	Pre-selected: Pre-Trigger
Enable TE: Samples Pe	OR All	O Pre-selected: Pre-Trigger ✓ User-selected: ✓ Ⅲ ▶

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.

	Design Window Help										
⊕, ⊝	, Q.										
R	eady 🔯 🔟 🗹 topcount_LA0										
Trigger Unit											
Name Signals (MSB:LSB) Operator Radix Value											
countbi	counter1/countbi[7:0]	<= •	He 🔻	88							
dir	directionR	<= •	Bin 🔻	1							
Name	Expression dir THEN countbi	Sequence Depth	Max S		Max Event Counter						
Frigger O		- Trigger Pos	_		1						
nable TE		Pre-sele	ected:	Post-Trigger 🔹 🔻]						
nable TE amples P		 Pre-sele User-sele 	ected: F	Post-Trigger 🔹 🔻)						

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 <a>D 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 witton.

You now see the waveforms displayed, as shown in Figure 44.

Figure 44: Reveal Analyzer Waveform

🚯 Reve	al Analyzi	er - C:/my	_diamond_tutorial/LEC	itest/Test.rva *													00	х
File V	iew De	sign W	indow Help															
	B, Q, Q	2																
	Comple	ted	🔁 🔝 📝 topcour	it_LA0														
Bu	s/Signal	Data	0:32	0:64	0.96	0:128	0:160	0:192	0:224	0:256	0:288	0:320	0:352	0:384	0:416	0:448	0:480	
Þ	ountbi	88 -																
	dir 👘	1							ſ									
P (ountai	89 -																
4																		F
LATI	gger L	A Wavefo	m															

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*.