# Lattice Diamond User Guide



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.

Convention	Meaning or Use
Bold	Items in the user interface that you select or click. Text that you type into the user interface.
<italic></italic>	Variables in commands, code syntax, and path names.
Ctrl+L	Press the two keys at the same time.
Courier	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

#### Introduction 1 Lattice Diamond Overview 1 User Guide Organization 2

#### Getting Started 3

Prerequisites Running Lattice Diamond Creating a New Project Opening an Existing Project Importing an ispLEVER Project Next Steps Differences from ispLEVER

#### Design Environment Fundamentals 15

Overview 15 Project-Based Environment 16 Process Flow 17 Shared Memory 18 Context-Sensitive Data Views 18 Cross-Probing 20

#### User Interface Operation 23

Overview 23 Menus and Toolbars 24 Project Views 25 Tool View Area 26 Output and Tcl Console 26 Status Information 27 Basic UI Controls 27

Start Page 28 File List 28 Process 30 Hierarchy 30 Reports 32 Tool Views 34 Tcl Console 35 Output 35 Error, Warning, and Info 36 Find Results 36 Common Tasks 37 Controlling Views 37 Managing Layouts 38 Cross-Probing Between Views 42 Working with Projects 43 Overview 43 Implementations 45 Input Files 46 Synthesis Constraint Files 47 LPF Constraint Files 47 Debug Files 48 Script Files 49 Analysis Files 49 Programming Files 49 Strategies 50 Area 51 I/O Assistant 52 Quick 53 Timing 55 User-Defined 55 Common Tasks 56 Creating a Project 56 Changing the Target Device 56 Setting the Top Level of the Design 56 Editing Files 57 Saving Project Data 57 Lattice Diamond Design Flow 59 Overview 59 Design Flow Processes 60 Running Processes 61 Refreshing a Process State 63 Implementation Flow and Tasks 63 Run Management 64 Synthesis Constraint Creation 64 LPF Constraint Creation 66 Simulation Flow 67 I/O Assistant Flow 70

Summary of Changes from ispLEVER 72 Working with Tools and Views 73 Overview 73 Shared Memory 73 Cross Probing 73 View Menu Highlights 74 Start Page 74 Reports 75 Preference Preview 76 Tools 76 Spreadsheet View 77 Package View 81 Device View 82 Netlist View 83 NCD View 85 IPexpress 86 Platform Designer 87 Clarity Designer 88 Reveal Inserter 88 Reveal Analyzer 91 Floorplan View 96 Physical View 97 Logic Block View 98 Timing Analysis View 98 Netlist Analyzer 103 LDC Editor 104 Schematic Editor 106 Power Calculator 106 ECO Editor **111** Programmer 113 Soft Error Injection 114 Deployment Tool **116** Model 300 Programmer 116 Programming File Utility 118 Download Debugger 119 Partition Manager 119 Run Manager 120 Synplify Pro for Lattice 122 Active-HDL Lattice Edition 122 Simulation Wizard 123 Common Tasks 123 Controlling Tool Views 123 Using Zoom Controls 126 Displaying Tool Tips 126 Setting Display Options 126 Tcl Scripting 127 Overview 127 Tcl Console 127 External Tcl Console 128 Commands 129

Lattice Diamond Tcl Console 130 Project Manager 130 NCD 130 NGD 131 Reveal Inserter 131 Reveal Analyzer 131 Power Calculator 132 Programmer 132 Incremental Design Flow 132 Compile Lattice FPGA Simulation Libraries 133 Tcl Scripting From Command-Line Shells 133 Lattice Diamond Example Tcl Script 133 DOS Script for Running Lattice Diamond Tcl Script 133 Bash Script for Running Lattice Diamond Tcl Script on Windows 134 Bash Script for Running Lattice Diamond Tcl Script on Linux 134 Advanced Topics 135 Shared Memory Environment 135 Memory Usage 136 Clear Tool Memory 136 Environment and Tool Options 137 Pin Migration 138 Batch Tool Operation 139 Tcl Scripts 139 Creating Tcl Scripts from Command History 139 Creating Tcl Scripts from Scratch 140 Sample Tcl Script 140 Running Tcl Scripts 140

Project Archiving 141

File Descriptions 143

## Chapter 1



## Introduction

Lattice Diamond<sup>®</sup> software is the leading-edge software design environment for cost- sensitive, low-power Lattice FPGA architectures. It is the nextgeneration replacement for ispLEVER. Lattice Diamond's integrated tool environment provides a modern, comprehensive user interface for controlling the Lattice Semiconductor FPGA implementation process. Its combination of new and enhanced features allows users to complete designs faster, more easily, and with better results than ever before.

This user guide describes the main features, usage, and key concepts of the Lattice Diamond design environment. It should be used in conjunction with the Release Notes and reference documentation included with the product software. The Release Notes document is also available on the Lattice Web site and provides a list of supported devices.

### **Lattice Diamond Overview**

Lattice Diamond uses an expanded project-based design flow and integrated tool views so that design alternatives and what-if scenarios can easily be created and analyzed. The new *Implementations* and *Strategies* concepts provide a convenient way for users to try alternate design structures and manage multiple tool settings.

System-level information—including process flow, hierarchy, and file lists—is available, along with integrated HDL code checking and consolidated reporting features.

A fast Timing Analysis loop, ECO Editor, and Programmer provide new capabilities in the integrated framework. The cross-probing feature and the new shared memory architecture ensure fast performance and better memory utilization.

Lattice Diamond is highly customizable and provides Tcl scripting capabilities from its built-in console or from an external shell.

### **User Guide Organization**

This user guide contains all the basic information for using the Lattice Diamond software. It is organized in a logical sequence from introductory material, through operational descriptions, to advanced topics.

Key concepts and work flows are explained in "Design Environment Fundamentals" on page 15 and "Lattice Diamond Design Flow" on page 59.

Basic operation of the design environment is described in "User Interface Operation" on page 23.

Other parts of the book provide greater detail and practical usage information. The chapter "Working with Projects" on page 43 shows how to set up project implementations and strategies. "Working with Tools and Views" on page 73 describes the many tool views available.

## Chapter 2



## **Getting Started**

This chapter explains how to run Lattice Diamond and open or create a project. It includes instructions for importing a project from ispLEVER and explains key differences between Lattice Diamond and ispLEVER.

For more information about project fundamentals, see the chapters "Design Environment Fundamentals" on page 15 and "Working with Projects" on page 43.

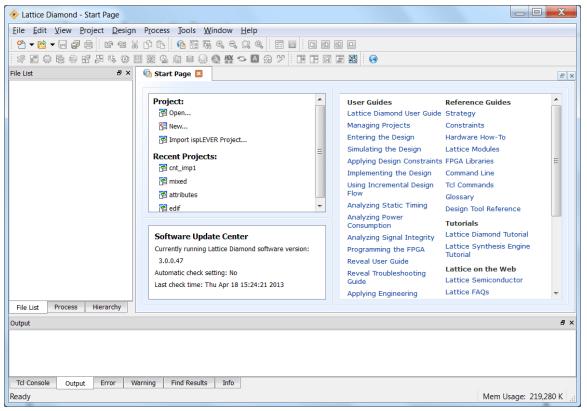
### Prerequisites

It is assumed you have already installed the Lattice Diamond software and product license. See the Lattice Diamond Installation Notice for complete information on product installation.

### **Running Lattice Diamond**

To run Lattice Diamond, select **Lattice Diamond** from the installation location. This opens the default Start Page.

#### Figure 1: Default Start Page



#### Note

If you are using Windows 7, you can use the "pin" command from the Start menu to place a Lattice Diamond shortcut on the Start menu or the Taskbar.

Do not use the "pin" command that is available from the Taskbar while Diamond is running. If you do so, the shortcut will fail when you try to use it to launch Diamond.

### **Creating a New Project**

The New Project wizard steps you through the process of creating a new project, allowing you to name the project and its implementation, add source files, and select a target device.

## You can open the New Project Wizard using one of the following methods:

- > On the Start Page, select **New** in the Project pane.
- From the File menu, choose New > Project

Several example project design files are included in Lattice Diamond. The following example procedure shows how to create a new project using the "mixedcounter" example.

On the Start Page, select **Project > New**.

Figure 2: Create a New Project

🚸 New Project	? 💌
	Create a Lattice Diamond Project
	This wizard will guide you through the creation of a new project.
	To create a Lattice Diamond project you will need to provide a name and location for your project. Next you will import source files. You may adjust implementation options. Finally you will process the source files to produce a programming file for a specific FPGA device.
	To continue, dick Next.
	<back next=""> Cancel</back>

Click **Next** to open the Project Name dialog box.

Click **Browse** to open the Browse for Folder dialog box. Navigate to the Lattice Diamond examples directory and select the **mixed\_mode** folder, as shown:

New Proje	t	2
	<b>ne</b> name for your project and specify a directory v is will be stored.	where the project
Project:		
<u>N</u> ame:		
Location:	C:/lscc/diamond/3.0/	▼ B <u>r</u> owse
Implementa Na <u>m</u> e:	tion:	
Location:	C:/lscc/diamond/3.0/	

#### Figure 3: Project Name

#### Figure 4: Project Browser

Organize   New folder			· · 0
-			<b>-</b>
lsc_env	Name	Date modified	Туре
> 📙 Iscc	📕 mixed	6/13/2013 10:28 AM	File folder
Mico_hold	k mixed_tcr.c	dir 6/13/2013 10:28 AM	File folder
MSOCache	🗼 source	6/11/2013 9:02 AM	File folder
b k my_designs			
b l new_cygwin	=		
L. News			
> 📙 opt			
L Outlook			
PerfLogs			
> 🗼 Perl			
Program Files	▼	III	
Folder: mixed_mo	de		

Enter a Project Name: Test.

Enter an Implementation Name: Test.

Figure 5:	Enter Pro	ject and Im	plementation	Name
-----------	-----------	-------------	--------------	------

🪸 New Proje	ct 🛛 🖓 📈
	me name for your project and specify a directory where the project es will be stored.
Project:	
<u>N</u> ame:	Test
Location:	C:/lscc/diamond/3.0/examples/mixed_mode
Implementa	
Na <u>m</u> e:	Test
Loca <u>t</u> ion:	C:/lscc/diamond/3.0/examples/mixed_mode/Test
	< <u>B</u> ack <u>N</u> ext > Cancel

Click Next to open the Add Source dialog box.

#### Figure 6: Add Source

New Project Add Source Add HDL, EDIF netlist, LPF constraints, or other files.	
Source files:	Add Source) Remove Source
Conv source to implementation directory	
Copy source to implementation directory	< Back Next > Cancel
	< Back

From this dialog box, you can add Verilog or VHDL source files, EDIF netlist files, LPF constraint files, schematic, debug and analysis files or any other project files. Diamond takes the source files and places them into the correct folders for the new project.

Click Add Source to open a file browser in the project example location.

Open the "source" folder, select all Verilog and VHD files and click **Open** to add the files to the project.

Figure 7: Source

Add HDL, EDIF netlis	t, LPF constraints, or other files.
<u>S</u> ource files:	Add Source
	amples/mixed_mode/source/count.v amples/mixed_mode/source/count_down.v
	amples/mixed_mode/source/count_down_2.v
	amples/mixed_mode/source/count_down_3.vhd amples/mixed_mode/source/count_up.v
C:/lscc/diamond/3.0/exa	• • • • • • • •
C:/lscc/diamond/3.0/exa	amples/mixed_mode/source/count_up.v

This process of browsing and adding source files can be done as many times as needed before going to the next step.

Notice the option to "Copy source to implementation directory." Select this option if you want to copy the external source files to the project's initial implementation. If you prefer to reference these files, clear this option. See "Implementations" on page 45 for further details.

Click **Next** to select the device.

**Figure 8: Select Device** 

Select Device: Family: LatticeSC LatticeSCM LatticeXP LatticeXP2 MachXO Platform Manager Diatform Manager Verformance grade:(_) 5 Operating conditions: Commercial Part Names: LFXP10C-SF388C	Device: LFXP6C LFXP6E LFXP10C LFXP10E LFXP15C LFXP15C LFXP20C I EYD20E ← Package type: FPBGA388		Device Informat Voltage: LUT: Registers: EBR Bits: EBR Blocks: Dist RAM: DSP: PLL: DLL: PCS: PIO Cells: PIO Pins:	tion: 1.8/2.5/3.3 9728 9728 216K 24 39K - 4 - 244 244 244
---	--	--	---	---

In this step you can select the target device, performance grade, package type, operating conditions and part name. For our example, we will use the default settings.

Click Next to open the Select Synthesis Tool dialog box.

Select the synthesis tool that you want to use.

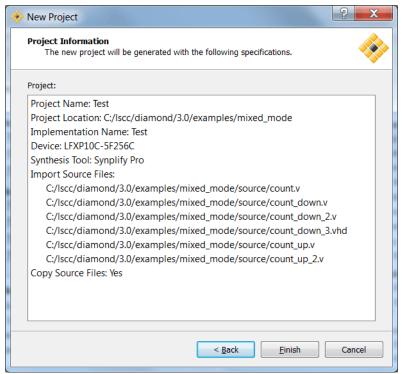
If you are designing for MachXO, MachXO2, MachXO3L, or Platform Manager, you have the option of using Lattice Synthesis Engine (LSE) as your synthesis tool instead of Synplify Pro for Lattice or another third-party synthesis tool. LSE is a synthesis tool custom-built for Lattice products and fully integrated with Diamond.

New Project	8
Select Synthesis Tool Specify a synthesis tool for the implementation.	4
Synthesis Tools:	
Synplify Pro	
Precision	
Lattice LSE <u>More information</u>	

Figure 9: Select Synthesis Tool

Click Next to open the Project Information summary.

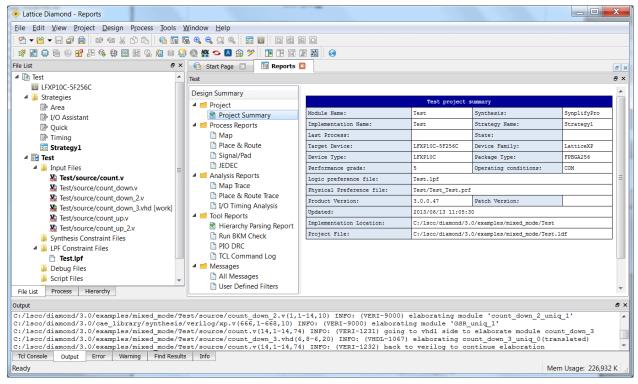
#### Figure 10: Project Summary



At this step or any other step in the process, you can click the **Back** button to review or change your selections.

Click **Finish**. The newly created project, shown in Figure 11, is now created and open.

Figure 11: Opened Project



Select the **File List** tab under the left pane, to view the Test project file list. Select the Process tab, to view the design flow processes and status.

To close a project, choose File > Close Project.

### **Opening an Existing Project**

Use one of the following methods to open an existing Lattice Diamond project:

- On the Start Page, select Open from the Project pane.
- From the File menu, choose **Open > Project**.
- On the Start Page, select the desired project from the Recent Projects list. Alternatively, choose a recent project from the File > Recent Projects menu.

You can use the Options dialog box to increase the number of projects that are shown in the Recent Projects list and to automatically load the previous project at startup. Choose **Tools > Options** to open the dialog box. To increase the number of recent projects listed, select **General** from the

Environment section, and then enter a number for "Maximum items shown in Recent items list." To automatically open the previous project during startup, select **Startup** from the Environment section, and then choose **Open Previous Project** from the "At Lattice Diamond startup" menu.

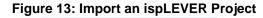
Figure 12: Open Recent Project

Project:		-
🚰 Ope	n	
🔁 New		
🚰 Impe	ort ispLEVER Project	=
Recent I	-	1
Recent I	-	
🚰 Test	-	
🚰 Test		

### Importing an ispLEVER Project

Use one of the following methods to import an ispLEVER project into Lattice Diamond:

- On the Start Page select Import ispLEVER Project from the Project pane.
- From the File menu, choose Open > Import ispLEVER Project



Project:	
🔁 Open	
월 New	
🚰 Import ispLEVER Project	=
Recent Projects:	
🔁 mixedcounter	
0-	

The file browser applies a **\*.syn** file filter to help you find ispLEVER project files. The ispLEVER project is converted to a Lattice Diamond project.

The import/conversion process has the following limitations:

- Any .ngo files will need to be manually copied into the Lattice Diamond project if these files were originally copied into the ispLEVER project; for example, .ngo files that were copied from Lattice IP generation.
- All .lpc files are replaced with .ipx files in Lattice Diamond. You will need to re-generate your IP by double-clicking the .lpc file name. The wizard will then open and help you generate the new .ipx file, replacing the .lpc file.

- If you select the "Copy source to Implementation directory" option, the following additional limitations will apply:
  - Verilog include files specified within the Verilog source files will not be copied.
  - Files associated with IPexpress Module .lpc files (such as .v, .txt, .ngo) will not be copied.
  - User-specified schematic symbols (.sym) will not be copied.

#### **Next Steps**

After you have a project opened in Lattice Diamond, you can go sequentially through the rest of this user guide to learn how to work with the entire design environment, or you can go directly to any topics of interest.

- The chapters "Design Environment Fundamentals" on page 15 and "Lattice Diamond Design Flow" on page 59 provide explanations of key concepts.
- "User Interface Operation" on page 23 provides descriptions of the functions and controls that are available in the Diamond environment.
- The chapters "Working with Projects" on page 43 and "Working with Tools and Views" on page 73 explain how to run processes and use the design tools.
- "Tcl Scripting" on page 127 provides an introduction to the scripting capabilities available, plus command-line shell examples.
- "Advanced Topics" on page 135 provides further details about environment options, shared memory, and Tcl scripting.

### **Differences from ispLEVER**

There are a number of differences between Lattice Diamond and ispLEVER. Key differences, especially regarding how projects are managed, include the following:

- ispLEVER has multiple project types, but there is only one Diamond project type. In ispLEVER, you need different projects types for each type of source; for example, one project for Verilog and a different project for VHDL. In Lattice Diamond, the project can include sources of different types. For example, one Lattice Diamond project can contain both Verilog and VHDL source files.
- Lattice Diamond includes implementations and strategies. These do not exist in ispLEVER.
- ispLEVER parses source file hierarchy when a project is opened, and it will question the existence of multiple top-level modules. Lattice Diamond does not display hierarchy by default (though it can be configured to do so), and you need to set the top-level design unit if multiple top-level modules exist.

- ispLEVER consists of a number of separate tools. Lattice Diamond is an integrated tool environment.
- All of the Lattice Diamond tool views share a common memory image of design data. This means that changes to the design data are seen by all tools.
- Lattice Diamond projects do not allow simulation test benches as source; only modules are contained within a Lattice Diamond project.
- Lattice Diamond 1.3 and later supports the ability to mark individual source files for simulation, synthesis, or both. This supports multiple file test benches and modules with different representations for simulation and synthesis. ispLEVER only supported a single test bench file for simulation and did not support different representations for the same module.

## Chapter 3



## Design Environment Fundamentals

This chapter provides background and discussion on the technology and methodology underlying the Lattice Diamond software design environment. Important key concepts and terminology are defined.

#### Overview

Understanding some of the fundamental concepts behind the Lattice Diamond framework technology will increase your proficiency with the tool and allow you to quickly come up to speed on its use.

Lattice Diamond is a next-generation software design environment that uses a new project-based methodology. A single project can contain multiple implementations and strategies to provide easily managed alternate design structures and tool settings.

The process flow is managed at a system level with run management controls and reporting. Context-sensitive views ensure that you only see the data that is available for the current state in the process flow.

The shared memory technology enables many of the advanced functions in Lattice Diamond. Easy cross-probing between tool views and faster process loops are among the benefits.

#### Note

You can run multiple instances of Lattice Diamond, by loading Lattice Diamond multiple times. This enables you to run different Diamond projects at the same time. However, you must not load the same project in more than one Lattice Diamond software instance, because this can cause conflicts in the software.

You can also run Diamond remotely. Refer to the Lattice Diamond Installation Notice for more information.

### **Project-Based Environment**

A project in Lattice Diamond consists of HDL source files, EDIF netlist files, synthesis constraint files, LPF constraint files, Reveal debug files, script files for simulation, analysis files for power calculation and timing analysis, and programming files.

It also includes settings for the targeted device and the different tools. The project data is organized into implementations, which define the project structural elements, and strategies, which are collections of tool settings.

The following File List shows the items in a sample project.

File List ₽× mixedcounter LFE2-35E-5F672C A Description Strategies 🗒 Area I/O Assistant Quick Timing Strategy1 Strategy2 R verilog vhdl verilog\_vhdl\_edif 4 퉬 Input Files edif/rev\_1/count8.edn Source/topcount.v Source/typepackage.vhd [work] source/count16.vhd [work] Synthesis Constraint Files LPF Constraint Files mixedcounter v.lpf mixedcounter\_vv.lpf mixedcounter vve.lpf Debug Files reveal1.rvl Script Files Analysis Files hixedcounter.pcf hixedcounter.tpf Programming Files 🖪 vhdl File List Process Hierarchy

Figure 14: File List

Each item that is displayed in **bold** means that it has been selected as the active item for an implementation. An implementation displayed in bold means that it has been selected as the currently active implementation for the project. Your project must have one active implementation, and the implementation must have one active strategy. Optional items, such as Reveal hardware debugger files, can be set as active or inactive. This differs

from ispLEVER, where the existence of Reveal debugger files means that debug is active.

The project is the top-level organizational element in Lattice Diamond, and it can contain multiple implementations and multiple strategies. This enables you to try different design approaches within the same project. If you want to have a Verilog version of your design, you will make an implementation that consists of only the Verilog source files. If you want another version of the design with primarily Verilog files but an EDIF netlist for one module, you will create a new implementation using the Verilog and EDIF source files. It will be the same project and design, but with a different set of modular blocks.

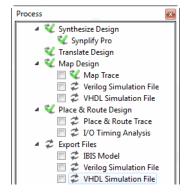
Similarly, if you want to try different implementation tool options, you can create a new strategy with the new option values.

You manage these multiple implementations and strategies for your project by setting them as active. There can only be one active implementation with its one active strategy at a time.

#### **Process Flow**

The Process View provides a system-level overview of the FPGA design flow. Each major step in the design process is shown, along with an icon that indicates its status. In the Map and Place & Route sections, you can select optional subtasks to be run every time. These selections are saved on a project basis. In the Export Files section, you can select the models or files that you want to be exported with the Export Files process. For example, if Bitstream File is checked, it will be generated and exported; if it is not checked, it will not be generated and exported.

#### Figure 15: Process Flow



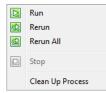
The process status icons are defined as follows:

- Process in initial state
- V Process completed successfully
- A Process completed with warnings, see Warning output

#### 獇 Process failed, see Error output

Right-clicking a process opens the controls for running, stopping, or cleaning up that process. The Run command runs the selected process if it is not upto-date. The Rerun command forces a process to run again, even if it is up-todate. The Rerun All command forces all processes to run again. The Clean Up Process command resets all processes and returns the status to the original state. To refresh a process state, see "Refreshing a Process State" on page 63.

#### Figure 16: Process Run Pop-up Menu



#### Shared Memory

Lattice Diamond uses a shared memory architecture. All tool and data views are looking at the same design data at any point in time. This means that when you change a data element in one view of your design, all other views will see the change, whether they are active or not.

When project data has been changed but not yet saved, an asterisk (\*) is displayed in the title tab of the view.

#### Figure 17: Title Tab with Changed Memory Indication

🌠 Spreadsheet View \* 🔀

Notice that the asterisks indicating changed data will appear in all views referencing the changed data.

If a tool view becomes unavailable, the Lattice Diamond environment will need to be closed and restarted.

### **Context-Sensitive Data Views**

The data in shared memory reflects the state or context of the overall process flow. This means that views such as Spreadsheet View will display only the data that is currently available, depending on process steps that have been completed.

For example, Figure 18 shows that the Process flow has been completed through Map Design but not through Place & Route Design. Therefore, Spreadsheet View shows no pin assignments.

Process &	×	Sta	art Page 🛛	🔃 Reports	🛛 🛛 🌠 Spre	eadsheet Vi	iew 🗵			e
<ul> <li>Synthesize Design</li> <li>Synplify Pro</li> </ul>	ø		Туре	Name	Group by	Pin	Bank	Vref	IO_TYPE	PULLMODE
V Translate Design	ហាំ	1	🐉 All Ports		N/A	N/A	N/A	N/A		
A 📽 Map Design	_⊈ 	2	Clock In…	clk	N/A			N/A	LVCMOS25	UP
🔲 🌮 Map Trace 🔲 🌮 Verilog Simulation File	ð	3	Input Port	reset	N/A			N/A	LVCMOS25	UP
	2	4	Input Port	direction	N/A			N/A	LVCMOS25	UP
✓ <sup>2</sup> Place & Route Design	2000	5	Output	count[7]	N/A			N/A	LVCMOS25	UP
Place & Route Trace I/O Timing Analysis		6	Output	count[6]	N/A			N/A	LVCMOS25	UP
🔺 🥏 Export Files		7	Output	count[5]	N/A			N/A	LVCMOS25	UP
☐ ⋨ IBIS Model		8	Output	count[4]	N/A			N/A	LVCMOS25	UP
VHUL SIMULATION FILE	₹J	9	Output	count[3]	N/A			N/A	LVCMOS25	UP
🔲 🥏 Bitstream File	2 1	10	Output …	count[2]	N/A			N/A	LVCMOS25	UP
🔲 🥏 PROM File	1000 1000 1000 1000	•	1	11						4
File Process HierarchyPost Map Res		Po	rt Assignments	Pin Assignme	nts Clock Re	source	Route Priority	Cell Mappir	ng Global Prefe	erences T 📢

Figure 18: Process Completed Through Map Design

After Place & Route Design has been completed, Spreadsheet View displays the pin assignments.

Figure 19: Process Completed Through Place & Route Design

Process	8 × 📢	Sta	art Page 🗵	🔃 Reports	🗵 🛛 🌠 Spre	adsheet Vie	w 🗵			æ
4 🛠 Synthesize Design	S.		Туре	Name	Group by	Pin	Bank	Vref	IO_TYPE	PULLMODE *
🛠 Synplify Pro 🛠 Translate Design	กกํ	1	Ports		N/A	N/A	N/A	N/A		E
4 🛠 Map Design	 	2	Clock In	clk	N/A	(AD15)	(4)	N/A	LVCMOS25	UP(UP)
🔲 🕏 Map Trace 🔲 🥏 Verilog Simulation File	5	3	Input Port	reset	N/A	(AD10)	(5)	N/A	LVCMOS25	UP(UP)
2 VHDL Simulation File	2	4	Input Port	direction	N/A	(AC10)	(5)	N/A	LVCMOS25	UP(UP)
4 💜 Place & Route Design		5	Output	count[7]	N/A	(AE5)	(5)	N/A	LVCMOS25	UP(UP)
Place & Route Trace I/O Timing Analysis		6	Output	count[6]	N/A	(AA7)	(5)	N/A	LVCMOS25	UP(UP)
✓ Z Export Files		7	Output	count[5]	N/A	(AB11)	(5)	N/A	LVCMOS25	UP(UP)
IBIS Model		8	Output	count[4]	N/A	(AF7)	(5)	N/A	LVCMOS25	UP(UP)
Verilog Simulation File           Image: Constraint of the second	<b>₽</b> ₽	9	Output	count[3]	N/A	(AC9)	(5)	N/A	LVCMOS25	UP(UP)
🔲 🕏 Bitstream File		10	Output	count[2]	N/A	(AF6)	(5)	N/A	LVCMOS25	UP(UP)
🔲 🥏 PROM File	2018 2018 10102	•	1	11						4
File Process HierarchyPost Map Res	5 🗵	Po	rt Assignments	Pin Assignm	ents Clock Re	source Ro	oute Priority	Cell Mapping	Global Prefe	rences T 📢 🕽

When you see a "Loading Data" message, it means that a process has been completed and that the shared memory is being updated with new data.

#### Figure 20: Loading Data



All tool views are dynamically updated when new data becomes available. This means that when you rerun an earlier process while a view is open and displaying data, the view will remain open but dimmed because its data is no longer available.

### **Cross-Probing**

Cross-probing is a feature found in most tool views. Cross-probing allows common data elements to be viewed in multiple tool views.

To see how this works, select a pin or signal in one view and right-click it. Select **Show In** to see a list of cross-probing views for the selected element.

#### Figure 21: Show In Pop-up Menu



For example, Figure 21 shows a tab group consisting of Floorplan View and Physical View. A signal is selected in Physical View.

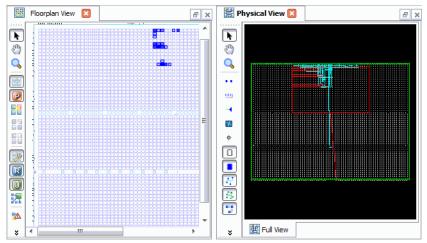


Figure 22: Physical View with Selected Signal

When you right-click the selected signal in Physical View and choose **Show In > Floorplan View**, the same selected signal is highlighted on the Floorplan View layout, as shown in Figure 23.

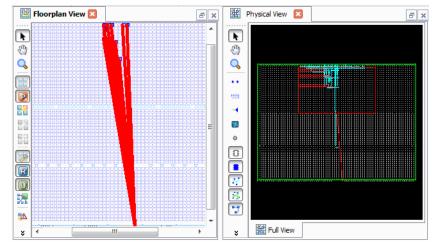


Figure 23: Selection in Floorplan and Physical Views





## **User Interface Operation**

This chapter describes the user interface features, controls, and basic operation. Each major element of the interface is explained. The last section in the chapter describes common user interface tasks.

#### Overview

The Diamond Lattice user interface (UI) provides a comprehensive, integrated tool environment. The UI is very flexible and configurable, enabling you to store layout preferences.

This chapter will take you through the operation of the main elements of the UI, but you should also explore the controls at your own pace. Figure 24 shows the Lattice Diamond main window in the default state.

Lattice Diamond - Start Page				
ile <u>E</u> dit <u>V</u> iew <u>Project</u> <u>D</u> esign P <u>r</u> ocess <u>T</u> ools <u>W</u> indow <u>H</u> elp				
		2 -		
Project: @ Open		User Guides	Reference Guides	
		Lattice Diamond User Guide		
8 New		Managing Projects	Constraints	
Market Import ispLEVER Project		Entering the Design	Hardware How-To	
Recent Projects:	=	Simulating the Design	Lattice Modules	
Ta cnt_imp1		Applying Design Constraints		1
Ta mixed		Implementing the Design	Command Line	
attributes		Using Incremental Design Flow	Tcl Commands	
	-	Analyzing Static Timing	Glossary	
P edif		Analyzing Power	Design Tool Reference	
		Consumption	Tutorials	
Software Update Center		Analyzing Signal Integrity	Lattice Diamond Tutorial	
Currently running Lattice Diamond software v	ersion:	Programming the FPGA	Lattice Synthesis Engine Tutorial	
3.0.0.47		Reveal User Guide		
Automatic check setting: No		Reveal Troubleshooting	Lattice on the Web	
Last check time: Thu Apr 18 15:24:21 2013		Guide	Lattice Semiconductor	
		Applying Engineering	Lattice FAQs	-
File List Process Hierarchy				
Dutput				
Tcl Console Output Error Warning Find Results Info				
Ready			Mem Usage: 21	9 280

### **Menus and Toolbars**

At the top of the main window is the menu and toolbar area. High-level controls for accessing tools, managing files and projects, and controlling the layout are contained here. All of the functionality in the toolbars is also contained in the menus. The menus also have functions for system, project and toolbar control.

Figure 25: Menu and Toolbar Area

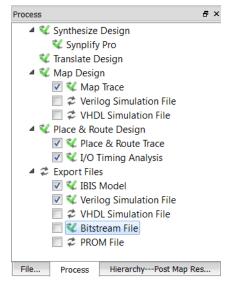
File	Edit	View	Project	Design	Process	Tools	Window	Help	
ł 🕐	- 🖻	• 🗐	j 🖨	12 11	¥ 🗅	Ē !	( <u>6</u> 🖭 🌆	⊕, ⊝,	
1 🚀	<b>e</b>	) 6	÷ 🖁	¢ 🔅 [	× * (	հ 🚈	III 😫 🔇	) 🔊 S	🖪 🎲 🏸 i 🏗 🖪 🖬 📰 i 🕥

The toolbars are organized into functional sets. The display of each toolbar is controlled in the **View > Toolbars** menu and also by right-clicking in the Menu and Toolbar area. Each toolbar can be repositioned by dragging and dropping it to a new location.

### **Project Views**

In the middle of the main window on the left side is the Project View area. This is where the overall project and process flow is displayed and controlled.

Figure 26: Project View Area



Tabs at the bottom of the Project View allow you to select between the following views:

- File List shows the files in the project organized by implementations and strategies. This is not a hierarchical listing of the design.
- Process shows the overall process flow and status for each step
- Hierarchy hierarchical design representation

### **Tool View Area**

In the middle of the main window on the right side is the Tool View area. This is where the Start Page, Reports View and all the Tool views are displayed.

Figure	27:	Tool	View	Area

Design Summary								
🔺 🔜 Project	Test project summary							
	dule Name:	Test	Synthesis:	SynplifyPro				
▲ Img	plementation Name:	Test	Strategy Name:	Strategy1				
🗋 Map	st Process:		State:					
Place & Route	rget Device:	LFXP10C-5F256C	Device Family:	LatticeXP				
	vice Type:	LFXP10C	Package Type:	FPBGA256				
	rformance grade:	5	Operating conditions:	COM				
	gic preference file:	Test.lpf						
Map Trace Phy Place & Route Trace	Physical Preference file: Test/Test_prf							
I/O Timing Analysis	oduct Version:	3.0.0.47	Patch Version:					
■ Tool Reports	dated:	2013/06/13 11:05:30						
	plementation Location:	C:/lscc/diamond/3	C:/lscc/diamond/3.0/examples/mixed_mode/Test					
	oject File:	C:/lscc/diamond/3	C:/lscc/diamond/3.0/examples/mixed_mode/Test.ldf					
<ul> <li>PIO DRC</li> <li>TCL Command Log</li> <li>Messages</li> <li>All Messages</li> <li>User Defined Filters</li> </ul>								

Multiple tools can be displayed at the same time. The Window toolbar includes controls for grouping the tool views as well as integrating all tool views back into the main window.

Each tool view is specific to its tool and can contain additional toolbars and multiple panes or windows controlled by additional tabs. The chapter "Working with Tools and Views" on page 73 provides more details about each tool and view.

### **Output and Tcl Console**

Near the bottom of the main window is the Output and Tcl Console area.

Tabs at the bottom of this area allow you to select between Tcl Console, Output, Error, Warning, Find Results, and Info. Tool output is automatically sent to the Output tab, and Errors and Warnings are automatically sent to their respective tabs.

(h	Sta	nt Page 🗵	E Reports	HDL HDL	Diagram 🗵	🛛 🌠 Spre	adsheet View 🛛	Package View
ø		Туре	Name	Group by	Pin	Bank	Vref	IO_TYPE
ហំ	1	🗦 All Ports		N/A	N/A	N/A	N/A	
_⊈ 	2	Clock In…	clk	N/A	(H12)	(2)	N/A	LVCMOS25
3	3	Input Port	rst	N/A	(J1)	(7)	N/A	LVCMOS25
2	4	Output	c_up[2]	N/A	(T2)	(5)	N/A	LVCMOS25
***	5	Output	c_up[1]	N/A	(T3)	(5)	N/A	LVCMOS25
M	6	Output	c_up[0]	N/A	(H6)	(7)	N/A	LVCMOS25
<u>~</u>	7	Output …	c_down[2]	N/A	(T6)	(5)	N/A	LVCMOS25
	8	Output	c_down[1]	N/A	(R6)	(5)	N/A	LVCMOS25
£₽	9	Output	c_down[0]	N/A	(H3)	(7)	N/A	LVCMOS25
	10	Output	c_up_2[2]	N/A	(M7)	(5)	N/A	LVCMOS25
翻翻	11	Output	c_up_2[1]	N/A	(T7)	(5)	N/A	LVCMOS25
9	12	Output	c_up_2[0]	N/A	(T10)	(4)	N/A	LVCMOS25
	•	111						
	Po	rt Assignments	Pin Assignmer	nts Clock Re	source Ro	oute Priority	Cell Mapping	Global Preferences

#### Figure 28: Multiple Tools

#### Figure 29: Output and Tcl Console Area

```
    Output

      F ×

        count_down_3_uniq_0(translated)
            C:/Projects/Design examples/mixed_mode/source/count.v(14,1-14,74) INFO: (VERI-1232) back to verilog to
            continue elaboration
            Design load finished with (0) errors, and (0) warnings.

        Tcl Console
        Output

        Error
        Warning

        Find Results
        Info
```

### **Status Information**

At the very bottom of the main window is status information. The information shown depends on the position of the mouse pointer, current memory usage, and whether unsaved preference changes are in memory. For more information on memory usage see "Advanced Topics" on page 135.

#### Figure 30: Status Information

Ready

Mem Usage: 515,896 K

### **Basic UI Controls**

The Lattice Diamond environment is based on modern industry standard user interface concepts. The menus, toolbars, and mouse pointer all behave in familiar ways. You can resize any of the window panes, drag and drop elements, right-click a design element to see available actions, and hold the mouse pointer over an object to view the tool tip.

Each of the Project and Tool views as well as the Outputs and Tcl Console items can be detached from the main window and operated as independent windows. Simply click the detach button in the upper-right corner.

After a view or item has been detached from the main window it can be reattached by one of two methods:

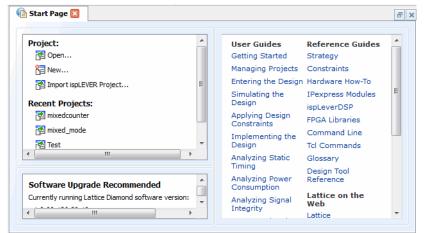
- For Project views, Outputs and the Tcl Console, double-click inside the window title bar, and the window will be re-attached to the main window.
- ► For Diamond Tool views, single-click the attach button 🖻 in the upperright corner, and the window will be re-attached to the main window.

Additionally, an **Integrate All Tools** button 📓 is available in the Window menu and toolbar. This control gathers all detached views and reintegrates them inside the main window.

### **Start Page**

The Start Page contains selections for opening projects, hyperlinks to product documentation, and software status and upgrade information. The Start Page appears in the Tool View area by default when Lattice Diamond is first launched.

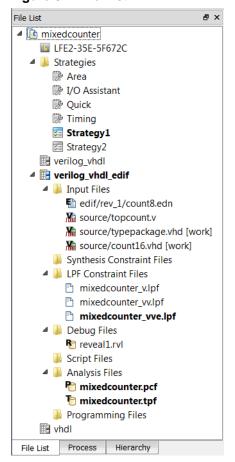
#### Figure 31: Start Page



The Start Page can be closed, opened, detached, and attached with the Attach button. See "Basic UI Controls" on page 27.

### **File List**

The File List is a project view that shows the files in the project, including implementations and strategies. It is not a hierarchical listing of the design, but rather a list of all the design source, configuration and control files that make up the project.



#### Figure 32: File List

At the top level in the File List is the project name. Directly below the project name is the target device, followed by the strategies, and then the implementations. There must be one active implementation, and it must have one active strategy. Active elements are indicated in **bold**.

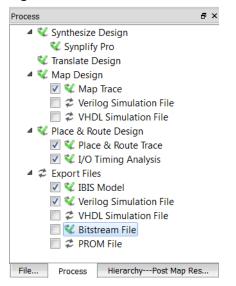
You can right-click any file or item in the File List to access a pop-up menu of currently available actions for that item. The pop-up menu contents vary, depending on the type of item selected.

The File List view can be closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

### Process

The Process View is a project view that displays the high-level process flow for the project.

Figure 33: Process View



The icons to the left of each step indicate the process status and are defined as follows:

- Process in initial state
- V Process completed successfully
- Process completed with warnings, see Warning output
- X Process failed, see Error output

You can right-click any step in the Process view to access a pop-up menu of currently available actions for that item, as explained in "Process Flow" on page 17.

The Process view can be selected, closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

### Hierarchy

The Hierarchy view is a project view that displays the design hierarchy.

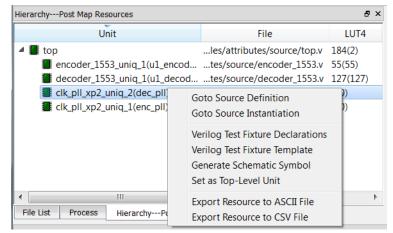
The Hierarchy View is displayed by default. If you would prefer that it not open by default, simply close Hierarchy View. The next time you launch Diamond, the Hierarchy View will not be opened. You can open it manually by selecting it from the View > Show View menu.

Figure	34:	Hierarc	hv	View
--------	-----	---------	----	------

HierarchyPost Map Resources		₽×
Unit	File	LUT4
🔺 🔳 top	les/attributes/source/top.v	184(2)
encoder_1553_uniq_1(u1_encod	tes/source/encoder_1553.v	55(55)
decoder_1553_uniq_1(u1_decod	tes/source/decoder_1553.v	127(127)
clk_pll_xp2_uniq_2(dec_pll)	tributes/source/clk_pll_all.v	0(0)
clk_pll_xp2_uniq_1(enc_pll)	tributes/source/clk_pll_all.v	0(0)
< <u> </u>		4
		P
File List Process HierarchyPost Map R	Resources	

Right-click any of the objects in the Hierarchy View to see the available actions.

#### Figure 35: Hierarchy Item Pop-up Menu



After synthesis, the Hierarchy view displays the calculated resource utilization data, such as LUT4, registers, and I/O registers. Device resources that would be consumed are shown to the right of each module. The first number in each column indicates the quantity of that resource that would be used by that module and all of its submodules. The second number, in parentheses, indicates the quantity of that resource that would be used by just that module, not including any of its submodules.

The numbers are updated after the synthesis and map stages of the implementation process.

The amount of information shown in the Hierarchy view can be controlled by right-clicking in the header row. In the drop-down menu, select the resource headers that you want displayed.

#### Figure 36: Post-Synthesis Resource Utilization

Unit	File	LUT4	PFU Registers	IO Registers	PFUMux	Carry Cells	Instantiated
📕 top	les/attributes/source/top.v	181(1)	85(0)	21(21)	6(0)	5(0)	61(51)
encoder_1553_uniq_1(u1_encoder	)tes/source/encoder_1553.v	54(54)	28(28)	0(0)	6(6)	0(0)	4(4)
decoder_1553_uniq_1(u1_decoder	r)tes/source/decoder_1553.v	126(126)	57(57)	0(0)	0(0)	5(5)	2(2)
clk_pll_xp2_uniq_2(dec_pll)	tributes/source/clk_pll_all.v	0(0)	0(0)	0(0)	0(0)	0(0)	2(2)
clk_pll_xp2_uniq_1(enc_pll)	tributes/source/clk_pll_all.v	0(0)	0(0)	0(0)	0(0)	0(0)	2(2)

#### Figure 37: Post-Map Resource Utilization

HierarchyPost Map Resources						×
Unit	File	LUT4	PFU Registers	IO Registers	Carry Cells	SLICE
🔺 📕 top	les/attributes/source/top.v	184(2)	85(0)	21(21)	5(0)	125(1.50)
encoder_1553_uniq_1(u1_encoder)	tes/source/encoder_1553.v	55(55)	28(28)	0(0)	0(0)	31(31)
decoder_1553_uniq_1(u1_decoder)	tes/source/decoder_1553.v	127(127)	57(57)	0(0)	5(5)	92.50(92.50)
clk_pll_xp2_uniq_2(dec_pll)	tributes/source/clk_pll_all.v	0(0)	0(0)	0(0)	0(0)	0(0)
clk_pll_xp2_uniq_1(enc_pll)	tributes/source/clk_pll_all.v	0(0)	0(0)	0(0)	0(0)	0(0)

Resource information for a selected module can be exported to a .csv file or to an ASCII text file. Right-click the desired module and choose the appropriate Export Resource command.

The Hierarchy view can be selected, closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

### Reports

The Reports View provides a centralized reporting mechanism in the Tools view area. The Reports View is automatically displayed and updated when processes are run. It provides a separate tab for each implementation, enabling you to quickly compare results.

Design Summary						_
🛚 🗾 Project			Test project sum	mary		
Project Summary		Module Name:	Test	Synthesis:	SynplifyPro	
Process Reports		Implementation Name:	Test	Strategy Name:	Strategy1	]
🖻 📓 Map		Last Process:	VHDL Simulation File	State:	Passed	1
🗋 Place & Route		Target Device:	LFXP10C-5F256C	Device Family:	LatticeXP	1
🗋 Signal/Pad		Device Type:	LFXP10C	Package Type:	FPBGA256	1
🗋 JEDEC		Performance grade:	5	Operating conditions:	СОМ	1
🛛 🗾 Analysis Reports	=	Logic preference file:	Test.lpf			1
Map Trace		Physical Preference file:	Test/Test_Test.prf			1
Place & Route Trace		Product Version:	3.0.0.47	Patch Version:		1
I/O Timing Analysis I/O Timing Analysis		Updated:	2013/06/13 14:08:59			1
Hierarchy Parsing R		Implementation Location:	C:/lscc/diamond/3.0/exa	mples/mixed_mode/Test		1
Run BKM Check		Project File:	C:/lscc/diamond/3.0/exa	mples/mixed_mode/Test.ldf		1
PIO DRC						-
TCL Command Log						
Messages						

The Design Summary pane on the left provides hyperlinks to the Project Summary, to report information for each step of the design process flow, and to messages from the implementation process. The right pane displays the report for the selected step.

When All Messages is selected in the Design Summary pane, all warning and error messages are displayed in the right pane. Click the down-arrow at the top right to display the filter options.

•	
	Show Filter Selection

The filter options and commands, shown in Figure 39, enable you to narrow the list by hiding certain categories of messages.

LEDtest Design Summary					
	General Filter (Hide Process: 📄 Synt Severity: 📄 🤇	hesis 🔲 Translate 📗	Map 🦳 Place & Route 📄 Export Warning 📄 🔀 Error	Filter D Save	Details Filter
Bitstream	ID	Process		Message	
<ul> <li>▲ Analysis Reports</li> <li>▷ Map Trace</li> <li>▷ Place &amp; Route Trace</li> <li>▷ Place &amp; Route Trace</li> </ul>	2019991	Synthesis	WARNING - CG921 :"C: \my_diamond_tutorial\LEDtest\s declared in this scope.	Find in Text	Ctrl+F
<ul> <li>I/O Timing Analysis</li> <li>Tool Reports</li> <li>I/O SSO Analysis</li> <li>Hierarchy Parsing Report</li> </ul>	2019991	Synthesis	WARNING - CG921 :"C: \my_diamond_tutorial\LEDtest\s already declared in this scope.	Filter Messages Exactly like This Filter Messages with This ID Filter Messages from This Process	
<ul> <li>Prietarchy Parsing Report</li> <li>Run BKM Check</li> <li>PIO DRC</li> <li>TCL Command Log</li> </ul>	2019991	Synthesis	WARNING - CG921 :"C: \my_diamond_tutorial\LEDtest\s declared in this scope.	Filter Messages with This Severity Filter Details List	
Messages     All Messages	2019991	Synthesis	WARNING - CG921 :"C: \my_diamond_tutorial\LEDtest\s already declared in this scope.	Promote Messages with This ID	
🖄 User Defined Filters	2019991	Synthesis	WARNING - CL118 :"C: \my_diamond_tutorial\LEDtest\s		• F1

#### Figure 39: Message Filtering

When you save a message filter, its name appears in the Messages section of the Design Summary pane under "User Design Filters." When you select the filter, all the filtered messages are hidden in the report pane. When you reselect All Messages, the filtered messages are again displayed.

You can right-click a report in the Design Summary list to access the Find in Text command. Selecting this command opens a text search area at the bottom of the Report View, shown in Figure 40.

The Report View can be selected, closed, opened, detached, and attached with the Attach button. See "Basic UI Controls" on page 27.

Design Summary	Level/ Cost [n	Numb cd] Unro		Worst Slack	Timing Score	Worst Slack(hold)	Timing Score (hold)	Run Time	NCI Sta
<ul> <li>Project</li> <li>Project Summary</li> <li>Process Reports</li> </ul>	5_1 *	0 ign saved		-0.075	219	-	-	16	Cor
Map Place & Route	Total (	real) rur	time .	for 1-seed:	16 secs				
<ul> <li>Signal/Pad</li> <li>JEDEC</li> <li>A analysis Reports</li> <li>Map Trace</li> </ul>					or Design "Te	st_Test_map.nc	d"		
<ul> <li>▶ Place &amp; Route Trace</li> <li>▶ I/O Timing Analysis</li> <li>■ Tool Reports</li> <li>♥ Hierarchy Parsing</li> <li>▶ Run BKM Check</li> </ul>	PAR: P1 Command Prefere Placeme	Line: pa	ir -w - Test_ -cost:	1 5 -i 6 -t Test.prf.	sion 3.0.0.47 : 1 -c 0 -e 0	-gui -exp par	UseNBR=1:par	CDP=0:par(	CDR=0:I
<ul> <li>PIO DRC</li> <li>TCL Command Log</li> <li>Messages</li> <li>All Messages</li> <li>User Defined Filters</li> </ul>	Design	name: cou sion: 3.2 LAI	INT P TTICE (P10C	lication pa	ar from file	Test_Test_map.	ncd.		

#### Figure 40: Report View Find in Text

### **Tool Views**

The Tool view area of the UI displays the active tools. Each tool that you have opened from the toolbar or the Tools menu is displayed. The Reports and Start page, which can be opened from the toolbar or the Windows menu, are also displayed. When multiple tools are active, the display can be controlled with the tab group functions in the Window toolbar. See "Common Tasks" on page 37 for more information on tab group functions.

Each tool view is specific to its tool and can contain additional toolbars, multiple panes, or multiple windows controlled by additional tabs. See "Working with Tools and Views" on page 73 for descriptions of each tool and view, plus details on controlling their display.

In Figure 41, You will notice an asterisk (\*) in the tool view tab title when there has been a change to the shared memory.

The Tool views can be selected, closed, opened, detached, and attached using the Attach button. See "Basic UI Controls" on page 27.

🖽 Reports 🗵 🖉 Spreadsheet View * 🗵 📴 Netlist View * 🔀	×
🐉 🚅 🐾 🖓 Find 🍸 Filter	
Ports	*
🔺 👭 Design Name: topcount	
🔺 📚 Ports	
4 📄 Input	
🖻 🗾 Clock	Ξ
direction	
reset	
Bidir	
4 🥌 Output	
4 [ count2[15:0]	
count2_0 (B9)	
count2_1	
— count2_10	
count2_11	
count2_12	
count2_13	Ŧ

Figure 41: Tool View Tab Title Showing Changed Data

### **Tcl Console**

The Tcl Console is an integrated console for Tcl scripting. You can enter Tcl commands in the console to control all of the functionality of Lattice Diamond. Use the Tcl help command (help) to display a listing of the groups of Lattice Diamond extended Tcl commands.

Figure 42: Tcl Console

> help		
For more	e information on a specific command, type "help <command/> ":	
dtc	Lattice Diamond Tcl Console extended Tcl commands.	
prj	Project Manager extended Tcl commands.	
ncd	NCD extended Tcl commands.	
ngd	NGD extended Tcl commands.	
hle	HDL Explorer extended Tcl commands.	
rvl	Reveal Inserter extended Tcl commands.	
rva	Reveal Analyzer extended Tcl commands.	
pwc	Power Calculator extended Tcl commands.	
pgr	Programmer extended Tcl commands.	
psb i	Pojo2 extended Tcl commands.	

The Tcl Console can be selected, closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

### Output

The Output View is a read-only area where tool output is displayed.

The Output view can be selected, closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

#### Figure 43: Output View



### Error, Warning, and Info

The Error, Warning, and Info views are read-only areas where messages of a specific severity are displayed. Right-clicking a message provides a menu of commands, including Location > Text Editor, which opens the source file in the Text Editor and highlights the location of the problem.

Figure 44: Error and Warning Display

	ID	Message			
⚠	2019991	WARNING - CL190 :"C:\lscc\diamond\3.0\examples\mixed_mode\source\count	_down_2.v":7:0:	7:5 Optimi	izing register bit c_down[0] to a constant 1
⚠	2019991	WARNING - CL260 :"C:\lscc\diamond\3.0\examples\mixed_mode\source\count	_down_2.v":7:0:	7:5 Pruning	g register bit 0 of c_down[2:0]
A	2019991	WARNING - CL190 :"C:\lscc\diamond\3.0\examples\mixed_mode\source\count			register bit c_up[0] to a constant 0
⚠	2019991	WARNING - CL260 :"C:\lscc\diamond\3.0\examples\mixed_mode\source\co	Сору	Ctrl+C	pister bit 0 of c_up[2:0]
⚠	2019993	WARNING - MT420  Found inferred clock count clk with period 5.00ns. Please	Select All Clear All	Ctrl+A	k on object "p:clk"
_			Find in Text		

The Error, Warning, and Info views can be selected, closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

## **Find Results**

The Edit > Find in Files command enables you to search for information in the files within your project directory. The search results are then displayed in the Find Results view.

Figure	45:	Find	Results	View
--------	-----	------	---------	------

Find Results		₽×
C:/lscc/diamond/3.0/examples/mixed_mode/source/count.v(7):output [2:0]c_down_2;		
C:/lscc/diamond/3.0/examples/mixed_mode/source/count.v(8):output [2:0]c_down_3;		
C:/lscc/diamond/3.0/examples/mixed_mode/source/count.v(11):count_down count_down_inst (.clk(clk), .rst(rst), .c_down(c_down));		
C:/lscc/diamond/3.0/examples/mixed_mode/source/count.v(13):count_down_2 count_down_2_inst (.clk(clk), .rst(rst), .c_down(c_down_2));		
C://scc/diamond/3.0/examples/mixed_mode/source/count.v(14):count_down_3 count_down_3_inst (.clk(clk), .rst(rst), .c_down(c_down_3));		_
C:/lscc/diamond/3.0/examples/mixed_mode/source/count_down.v(1):module count_down (clk,rst,c_down)/* synthesis GSR="DISABLED" */;		•
< III		4
Tcl Console Output Error Warning Find Results Info		
Ready	Mem Usage:	: 235,292 K

The Find Results view can be selected, closed, opened, detached, and attached using the double-click method. See "Basic UI Controls" on page 27.

### **Common Tasks**

The Lattice Diamond UI controls many tools and processes. The following sections describe some of the more commonly performed tasks.

## **Controlling Views**

All of the views in Lattice Diamond are controlled in a similar manner, even though the information they contain varies widely. Here are some of the most common operations:

- Open Use the View > Show Views menu selections or right-click in the menu or toolbar areas to select a view from the pop-up menu.
- Select If a view is already open you can select its tab to bring it to the front.
- Close Click the x in the upper right corner of the view, or right-click in the menu or toolbar area and select the view from the pop-up menu to clear the check mark.
- Detach Click the detach button a in the upper right corner of the view
- Attach Use one of the two following methods to re-attach a view to the main window:
  - For project views, Output and the Tcl Console, double-click in the window title bar.
  - For tool views, click the attach button <a>[</a>]
- Attach All tool views To re-attach all tool views that have been detached, click the Integrate All Tools button an the Window toolbar.
- Move Click and hold a view's tab, and then drag and drop the view to a different position among the open views.

**Using a Tab Group** You can use the Window menu or the Window toolbar to split off a view and control it as a separate tab group. This allows you to examine two open views side by side. The controls work as follows:

- TR Split Tab Group displays two views side by side
- Merge Tab Group merges a split tab group back into the primary view
- Move to Another Tab Group moves the selected tab to the other tab group
- Switch Tab Group Position switches the positions of the two tab groups

Figure 46 shows the display after selecting **Split Tab Group**.

#### Figure 46: Split Tab Group

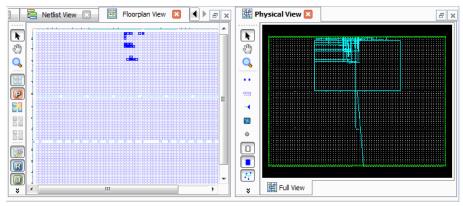
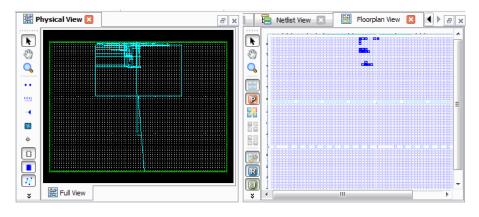


Figure 47 shows the display after selecting Switch Tab Group Position.

Figure 47: Switch Tab Group Position



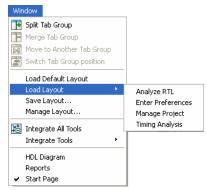
# **Managing Layouts**

Diamond's layout management utilities allow you to load a predefined layout of views and to create your own customized layouts. Predefined and customized layouts enable you to get to work immediately on a specific task, such as analyzing your source code or setting design constraints.

### **Using Predefined Layouts**

Four predefined layouts are included with Diamond and are available from the Window menu.

#### Figure 48: Diamond Predefined Layouts



Each of these predefined layouts is described in the Manage Layout dialog box, which is also available from the Window menu. The Information pane

#### **Figure 49: Predefined Layouts**

Туре		Layout Nam	e		
Predefined	Analyze RTL				
Predefined	Enter Preferences				
Predefined	Manage Project				
Predefined	Timing Analysis				
Information:		Preview:			
This layout a - Timing An - Spreadshe		Preview:			
This layout a - Timing An	alysis	Preview:	©Lattice Diamond • Timing To Be yes high four buy	ann Taols Window Halp	53
This layout a - Timing An - Spreadshe - Reports	alysis eet (TPF)	Preview:	Bie Bit View Broject Design Proce	am Dok Wadov Belp 1 S. s. s. c. s. B. B. D. C. C. C. C. C. C.	5)
This layout a - Timing An - Spreadshe - Reports	alysis :et (TPF) s: Place & Route process	Preview:	Bie Edit View Project Design Proce Project Bie Bie al Constant Constant of 200 Bie of the State State Bie Strayers Dit of Synastemer State	am Dok Watov Help 2 S S S G S N B D C C C C C C C C C S S S C Z Z B C	51
This layout a - Timing An - Spreadshe - Reports Prerequisites	alysis :et (TPF) s: Place & Route process	Preview:	Bie Balt View Broject Design From	ac Dob Wade Belo 5 4 4 0 4 0 8 0 0 0 0 0 0 0 1 5 4 4 0 7 2 8 0 1 5 5 7 7 7 8 0 1 5 5 7 7 8 0 1 5 7 7	
This layout a - Timing An - Spreadshe - Reports Prerequisites successfully To run Place	alysis set (TPF) s: Place & Route process completed. & Route Design:	Preview:	Bie Edit View Project Design Proce Project Bie Bie al Constant Constant of 200 Bie of the State State Bie Strayers Dit of Synastemer State	ana Taola Wandow Hado Sana a a a a Bara 22 a a a a a Sana ana a a 2 a a a a a a a a a a a a a	
This layout a - Timing An - Spreadshe - Reports Prerequisites successfully To run Place 1. In the Pro	alysis set (TPF) s: Place & Route process completed. & Route Design: cess view, right click on Place	Preview:	Bie Edit View Project Design Proce Project Bie Bie al Constant Constant of 200 Bie of the State State Bie Strayers Dit of Synastemer State	en Dek (Medor Bel) 2 S - S - S - S - S - S - S - S - S - S	
This layout a - Timing An - Spreadshe - Reports Prerequisites successfully To run Place 1. In the Pro & Route Des	alysis eet (TPF) s: Place & Route process completed. & Route Design: cess view, right click on Place ign.	Preview:	Pro Die You Project Design Person Pro Die 1992 (Provincial Configuration Pro Die 1992 (Provincial Configuration Pro Die 1992 (Provincial Configuration Setting Provincial Configuration Provincial	en Dek (Medor Bel) 2 S - S - S - S - S - S - S - S - S - S	
This layout a - Timing An - Spreadshe - Reports Prerequisites successfully To run Place 1. In the Pro & Route Des 2. In the pop	alysis set (TPF) s: Place & Route process completed. & Route Design: cess view, right click on Place	Preview:	Bis Bit the Project Design Prov Profession Lines (1994) in the Profession Lines (1994) in th	en Dek (Medor Bel) 2 S - S - S - S - S - S - S - S - S - S	

provides prerequisites and instructions for viewing all the windows of the layout. For example, if you load the Timing Analysis predefined layout and your design is still in the pre-route stage, you will need to run Place & Route in order to view the Timing Analysis View.

You cannot modify or delete any of the predefined layouts from the Manage Layout dialog box. However, you can load one of the predefined layouts into Diamond's main window, modify it by opening and closing or detaching views, and then save the arrangement as a customized layout.

**Loading a Predefined Layout** You can load a predefined layout by selecting it from the Window menu or from the Manage Layout dialog box. When you load a predefined layout, Diamond does not close any of the tool views that are already open. Any currently open tool views that are part of the predefined layout will be arranged according to the layout when it is loaded. Other open tool views will remain in their current positions. For example, if you have Power Calculator detached as a separate window when you load the "Analyze RTL" predefined layout, Power Calculator will remain open and in its detached position.

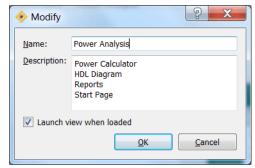
### **Creating Customized Layouts**

When you have the UI set up the way you like, you can save the arrangement as a customized layout by using the Window > Save Layout command. You can save as many of these customized layouts as you want, giving each a unique name. The names of saved customized layouts are added to the Window > Load Layout menu, and they appear as "Customized Layouts" in the Manage Layouts dialog box.

You can use the Manage Layout dialog box to delete a customized layout, and you can open the Modify dialog box to make minor changes. The Modify dialog box enables you to change the name of the customized layout and to select or clear the "Launch view when loaded" option. However, it does not allow you to add or delete tool views.

**Using the "Launch view when loaded" Option** The "Launch view when loaded" option will cause all of the layout's tool views to open when you load the customized layout. This gives you quick access to the tools and is very useful for layouts that include only a few tool views. For more complex layouts, this option can cause a long delay while each tool view gets loaded with the layout. If your customized layout includes a lot of tool views, you should clear this option.

#### Figure 50: Modify Layout



**Loading a Customized Layout** You can load a customized layout by selecting it from the Window menu or from the Manage Layout dialog box. When you load a customized layout, Diamond does not close any of the tool views that are already open. If the customized layout is set to "Launch view when loaded," the Load Layout command will open all of the layout's tool views that are currently unopened. Any of the layout's tool views that were already open will be arranged according to the layout. Any other open tool views will remain in their current positions.

If the customized layout is not set to "Launch view when loaded," the Load Layout command will not open any of the layout's tool views. For these types of layouts, first open the layout's tool views that you need to work with, and then choose the Load Layout command. The tool views will be arranged according to the customized layout. You can later open any or all of the remaining tool views that are part of the customized layout and choose the Load Layout command again. These tool views will also be arranged according to the layout.

### Layout Management Commands

All of the commands for layout management are available from the **Window** menu:

- Load Default Layout Loads the predefined or customized layout that has been specified as the default in the Manage Layout dialog box. This command gives you quick access to your default layout after you have opened your project in Diamond.
- Load Layout Provides quick access to the predefined and customized layouts.
- Save Layout Enables you to save the arrangement of all open views in your current session as a customized layout. The Save Layout dialog box shows a preview of the layout and lists all the views that will be included in the customized layout.
- Manage Layouts Opens the Manage Layout dialog box, which allows you to preview and load a predefined or customized layout, delete a customized layout, and select a layout as the default so that it will open with the Load Default Layout command. The Layout Name list displays the default layout in bold type. If a lot of views are listed for a customized

layout, you should leave the "Launch view when loaded" option cleared to avoid a long wait as the layout is loaded.

		Layout Name	Load
Predefined	Analyze RTL		Modify
Predefined	Enter Preferences		Mouny
Predefined	Manage Project		Delete
Predefined	Timing Analysis		
Customized	Floorplanning		Set as Defa
Customized	Power Analysis		
			<u>C</u> lose
Floorplan Vie Netlist View Spreadsheet Reports Start Page	(detached)		

Figure 51: Customized Layout as Default

# **Cross-Probing Between Views**

It is possible to select a data object in one view and see that same data object in a different view or views. Right-click a selected object; if cross-probing is available for that object you will see a **Show In** sub-menu with available views listed. If you select a view that is not yet open, Diamond will open it automatically. Cross-probing is available for most tool views.

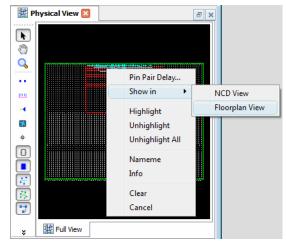


Figure 52: Show In

# Chapter 5



# **Working with Projects**

This chapter covers projects and their elements. Implementations and strategies are explained and some common project tasks are shown.

## Overview

A project is the top organizational element in the Lattice Diamond design environment. Projects consist of design, constraint, configuration and analysis files. There is only one project open at a time, and a single project can include multiple design structures and tool settings.

You can create, open, or import a project from the Start Page. See the chapter "Getting Started" on page 3 for instructions on creating a new project.

🕒 Start Page 🛛 ъ× Project: User Guides **Reference Guides** 🔁 Open... Getting Started Strategy Managing Projects Constraints 🔁 New... Entering the Design Hardware How-To Mark Import ispLEVER Project... Simulating the IPexpress Modules Ξ Recent Projects: Design ispLeverDSP Applying Design FPGA Libraries 🔁 mixedcounter Constraints 🚰 mixed\_mode Implementing the Command Line 🔁 Test Design Tcl Commands Analyzing Static Glossary Timing Design Tool Analyzing Power Reference . Software Upgrade Recommended Consumption Lattice on the Currently running Lattice Diamond software version: Analyzing Signal Web Integrity . . . . . . 4 Lattice

Figure 53: Opening a Project from the Start Page

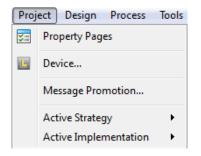
The File List view shows a project and its main elements.

File List ₽× Image: A mixed counter LFE2-35E-5F672C A B Strategies 🗒 Area P I/O Assistant Duick 🖗 🗒 Timing Strategy1 Strategy2 verilog\_vhdl verilog\_vhdl\_edif 4 퉬 Input Files edif/rev\_1/count8.edn Source/topcount.v Source/typepackage.vhd [work] Source/count16.vhd [work] Synthesis Constraint Files LPF Constraint Files mixedcounter v.lpf mixedcounter\_vv.lpf mixedcounter\_vve.lpf 4 퉬 Debug Files reveal1.rvl Script Files 4 퉬 Analysis Files hixedcounter.pcf T mixedcounter.tpf Programming Files 🖪 vhdl File List Process Hierarchy

Figure 54: Project Files in the File List

The Project menu commands enable you to examine the project properties, change the target device, change the severity level of warning messages, set the synthesis tool, show the active strategy tool settings and set the top level design unit.

#### Figure 55: Project Menu



### Implementations

An implementation is the structure of a design and can be thought of as *what* is in the design. For example, one implementation might use inferred memory while another implementation uses instantiated memory. Implementations also define the constraint and analysis parameters for a project.

There can be multiple implementations in a project, but only one implementation can be active at a time. And there must be one active implementation. Every implementation has an associated active strategy. Strategies are a shared pool of resources for all implementations and are discussed in the next section. An implementation is created whenever you create a new project.

Implementations consist of the following files:

- Input files
- Synthesis constraint files
- LPF constraint files
- Debug files
- Script files
- Analysis files
- Programming files

To add a new implementation to an existing project, right-click the project name in the File List project view and select **Add > New Implementation**. In the New Implementation dialog box, you can set the implementation name, directory, default strategy, and you can add source files. When you select **Add Source** you have a choice of browsing for the source files or using a source from an existing implementation.

Notice that you have the option to "Copy source to implementation directory." If this option is selected, the source files will be copied from the existing implementation to the new implementation, and you will be working with different source files in the two implementations. If you want the two implementations to share the same source files and stay in sync, make sure that this option is not selected.

To make an implementation active, right-click its name in the File List and choose **Set as Active Implementation**.

To add a file to an implementation, right-click the implementation name or any file folder in the implementation and choose **Add > New File**, or choose **Add > Existing File**.

lame:	verilog_vhdl2	Synthesis Tool:	Synplify Pro	
Directory:	verilog_vhdl2	Default Strategy:	Area	
ocation:	C:/lscc/diamond/3.2/examples/mixed_mode_SC/verilog	g_vhdl2		
Source fi	iles:		Add Source 🔻	Remove Source
C:/Iscc/				
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/	'count_down.v 'count_down_2.v 'count_down_3.vhd 'count_up.v		
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/ 'diamond/3.2/examples/mixed_mode_SC/Test/source/	'count_down.v 'count_down_2.v 'count_down_3.vhd 'count_up.v		

# **Input Files**

Input files are the design source files for the project. Input files can be any combination of Verilog, VHDL, and EDIF files.

Right-click an input file name to open a pop-up menu of possible actions for that file.

#### Figure 57: Input File Actions

4 🗄 mixed	
a 🌗 Input Files	
Source/count	Open With Open Containing Folder Add Clone Implementation
mixed.lpf mixed/source,	Exclude from Implementation Remove
Debug Files Script Files Analysis Files	Select Synthesis Tool Set Top-Level Unit
Programming File Image: Programming File Programming File	Include for  Properties

You can use the "Include for" commands to specify that a source file be included for both synthesis and simulation, synthesis only, or simulation only.

# **Synthesis Constraint Files**

Synthesis constraint files are constraint files that are directly interpreted by the synthesis engine. Starting with the G-2012.09 (Diamond 2.1) release, Synplify Pro replaced the old Synplify-style constraints (.sdc) with the Synopsys standard timing constraints (.fdc). New constraints created in Synplify Pro are in the new FDC format, by default, but the legacy SDC is still supported. Synopsys has provided a script, sdc2fdc, which does a one-time conversion of SDC files to the new FDC format. More information about this script can be found in the Synplify Pro release notes.

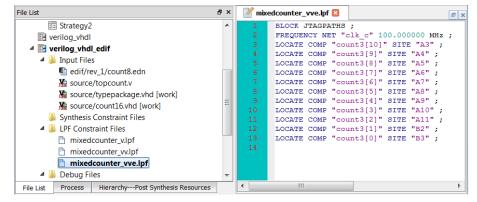
An .sdc file or .fdc file can be added to an implementation if the selected synthesis tool is Synplify Pro or Precision. If the selected synthesis tool is the Lattice Synthesis Engine (LSE), which is available for MachXO/2 and Platform Manager devices, a Lattice design constraint (.ldc) synthesis file can be added. Constraints in the .ldc file use the Synopsys constraint format.

An implementation can have multiple synthesis constraint files. Multiple .sdc or .fdc files can be active at the same time, but only one .ldc file can be active at a time. Unlike LPF constraints, a synthesis constraint file must be set as active by the user.

# **LPF Constraint Files**

LPF constraint files are logical preference files (.lpf), source files for storing logical constraints called preferences. Preferences that you add and edit using Diamond's preference-editing views, such as Spreadsheet View, are saved to the active .lpf file. The active logical preference file is then used as input for post-synthesis processes.

An implementation can have multiple .lpf files, but only one can be active at a time.



#### Figure 58: LPF Constraint File

# **Debug Files**

The files in the Debug folder are project files for the Reveal Inserter. They are used to insert hardware debug into your design. There can be multiple debug files, and one or none can be set as active. To insert hardware debug into your design, right-click a debug file name and choose **Set as Active Debug File** from the pop-up menu. The debug file name becomes bold, indicating that it is active.



File List			5	1,	×
	redcounter	_			7
	LFE2-35E-5F672	С			
⊿ 鷆	Strategies				
[	Area				
[	I/O Assistant				
[	濴 Quick				
[	🔅 Timing				
	Strategy1				
	Strategy2				
	verilog_vhdl				
	verilog_vhdl_ed	lif			
4	Input Files		at0 a da		
	edif/rev_1				
		•	ckage.vhd [work]		
	Source/co		-		
	Synthesis Cor				
	LPF Constrain				
	🖹 mixedcou				
	🖹 mixedcou	nter_	vv.lpf		
	🖹 mixedcou	inter	_vve.lpf		
4	Debug Files				
	R reveal1.rv	u	0		
	Script Files		Open		
4	Analysis File		Open With		
	P mixedco		Open Containing	) F	older
	T mixedco		Add		۱.
	Programmin vhdl		Clone Strategy		
Eile	Process Mo		Set as Active De	bu	ua File
1 IIC	1100000 1110		Remove		
			Set Top-Level Ur	nit	
			Include for		Þ
	1		Properties		
	L				

## **Script Files**

The Script Files folder contains the scripts that are generated by the Simulation Wizard. After you run the Simulation Wizard, the steps are stored in a simulation project file (.spf), which can be used to control the launching of the simulator.

#### Figure 60: Simulation Script File

Summary	
Simulator : Active-HDL Project Name : simulationfile Project Location : C:/lscc/diamond/3.2/examples/mixed_mode_SC Simulation Stage : RTL Simulation Files : C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_down_3.vhd C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_down_2.v C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_down.v C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_down.v C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_up.v C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_up.v C:/lscc/diamond/3.2/examples/mixed_mode_SC/simulationfile/count_v Simulation Libraries :	
Run simulator     Add top-level signals to waveform display     Run simulation 	Cancel

## **Analysis Files**

The Analysis Files folder contains Power Calculator files (.pcf) and Timing Preference files (.tpf). The folder can contain multiple analysis files, and one or none can be set as active. The active or non-active status of an analysis file affects the behavior of the associated tool view.

# **Programming Files**

Programming files (.xcf) are configuration scan chain files used by the Diamond Programmer for programming devices in a JTAG daisy chain. The .xcf file contains information about each device, the data files targeted, and the operations to be performed.

An implementation can have multiple .xcf files, but only one can be active at a time.The file must be set as active by the user.

# **Strategies**

Strategies are collections of all the implementation-related tool settings in one convenient location. Strategies can be thought of as recipes for how the design will be implemented. An implementation defines *what* is in the design, and a strategy defines *how* that design will be run. There can be many strategies, but only one can be active at a time. There must be one active strategy for each implementation.

Lattice Diamond provides four predefined strategies. It also enables you to create customized strategies. Predefined strategies cannot be edited, but they can be cloned, modified, and saved as customized user strategies. Customized user strategies can be edited, cloned, and removed. All strategies are available to all of the implementations, and any strategy can be set as the active one for an implementation.

To create a new strategy from an existing one, right-click the existing strategy and choose **Clone <strategy name> Strategy**. Set the new strategy's ID and file name.

Strategy ID:	Strategy3		
File name:	Strategy3	File extension:	sty
Save to location			
	: 'ON/Work/ActisDesign/Projects/Lattice Diamond/mixedcounter/mi	xedcounter	<b>.</b>

Figure 61: Cloning to Create a New Strategy

To make a strategy active, right-click the strategy name and choose **Set as Active Strategy**.

To change the settings in a strategy:

- Double-click the strategy name in the File List view
- Select the option type to modify
- Double-click the Value of the option to be changed

#### Figure 62: Strategy Settings

Process	Synplify P	ro		
<ul> <li>Synthesize Design</li> <li>Synplify Pro</li> </ul>	Display catalog: All		▼ Defau	lt
🔄 Precision	Name	Туре	Value	-
Translate Design	Allow Duplicate Modules	T/F	False	
Map Design Map Trace	Area	T/F	False	
✓ Place & Route Desig	Arrange VHDL Files	T/F	True	=
📰 Place & Route Tr	Clock Conversion	T/F	True	
🔄 IO Timing Analy	Command line Options	Text		
Timing Simulation	Default Enum Encoding	List	Default	
🔄 Bitstream	Disable IO Insertion	T/F	False	
	Export Diamond Settings to Synplify Pro GUI	List	No	
	FSM Encoding	T/F	True	
	Fanout Guide	Num	100	
	Force GSR	List	False	
	Frequency (MHz)	Num	200	-
< III >>				

The default values are displayed in plain blue font. Modified values are displayed in italic bold font.

Strategies are design data independent and can be exported and used in multiple projects.

## Area

The Area strategy is a predefined strategy for area optimization. Its purpose is to minimize the total logic gates used while enabling the tight packing option available in Map. It is commonly used for low-density devices such as MachXO.

Applying this strategy to large and dense designs might cause difficulties in the place and route process, such as longer time or incomplete routing.

Process	Synplify Pro				
<ul> <li>Synthesize Design</li> <li>Synplify Pro</li> <li>Precision</li> <li>Translate Design</li> <li>Map Design</li> <li>Map Trace</li> <li>Place &amp; Route Design</li> </ul>	Display catalog: All	▼ Default	▼ Default		
	Name	Туре	Value		
	Allow Duplicate Modules	T/F	False		
	Area	T/F	True		
	Arrange VHDL Files	T/F	True		
📰 Place & Route Trace	Clock Conversion	T/F	True		
📰 IO Timing Analysis	Command line Options	Text			
Timing Simulation	Default Enum Encoding	List	Default		
🗺 Bitstream	Disable IO Insertion	T/F	False		
	Export Diamond Settings to Synplify Pro GUI	List	No		
	FSM Encoding	T/F	True		
	Fanout Guide	Num	100		
	Force GSR	List	False		
	Frequency (MHz)	Num	1		
	Number of Critical Paths	Num		l	
	Number of Start/End Points	Num			
	Output Netlist Format	List	None		
	Output Preference File	T/F	True		
	Pipelining and Retiming	List	Pipelining Only		
	Push Tristates	T/F	True		
	Specifies optimization preference for area reduction	over timing (	delay reduction.		

Figure 63: Area Predefined Strategy

# I/O Assistant

The I/O Assistant strategy is a predefined strategy that is useful for I/O design. It helps you select a legal device pinout and produce LOCATE and IOBUF preferences for optimal I/O placement.

The benefit is that you will get results in I/O placement information early on, without having a complete design or any long run times after finishing place and route. However, applying this strategy to your design might take extra runtime, because it executes logic synthesis, translation, map, and I/O placement processes.

If you use the I/O Assistant strategy for your project, the generated .ncd file will be incomplete. Running the Export Files > Bitstream File process or the Export Files > JEDEC File process will fail. If you want to implement a complete design, you will need to choose another strategy and rerun all processes again. See "Lattice Diamond Design Flow" on page 59 for more information.

Synthesize Design Synplify Pro Precision			e Design splay catalog: Al Vefa		
Advanta					
	Name Type Value				
😒 Translate Design	Auto Hold-Time Correction	List	Off		
Map Design	Clock Skew Minimization	List	Off		
Map Trace	Command line Options	Text	-io		
Place & Route Design	Congestion-Driven Placement	List	Auto		
	Congestion-Driven Routing	List	1		
Timing Simulation	Create Delay Statistic File	T/F	False		
	Disable Timing Driven	T/F	False		
	Generate TRACE report for each iteration	T/F	False		
0	Guided PAR Matching Factor	Num			
0	Guided PAR Report Matches	T/F	False		
I	Ignore Preference Errors	T/F	True		
L.	Multi-Tasking Node List	File			
1	NCD Guide File	File			
F	Path-based Placement	List	Off		
F	Placement Effort Level [1-5]	Num	5		
F	Placement Iteration Start Pt	Num	1		
F	Placement Iterations [0-99]	Num	1		
Le	D1 . C D + D (1.003	A.1			

Figure 64: I/O Assistant Predefined Strategy

# Quick

The Quick strategy is a predefined strategy for doing an initial quick run. This strategy uses very low effort level in placement and routing to get results with minimum run time. If your design is small and your target frequencies are low, this is a good strategy to try. Even if your design is large, you might want to start with this strategy to get a first look at place-and-route results and to tune your preference file with minimum runtime.

The Quick strategy will give you results in the least possible time. However, the quality of these results in terms of achieved frequency will probably be low, and large or dense designs might not complete routing.

Process	Place & Route Design							
<ul> <li>Synthesize Design</li> <li>Synplify Pro</li> </ul>	Display catalog: All 🔍 D							
Precision	Name	Туре	Value	^				
📰 Translate Design	Disable Timing Driven	T/F	False					
4 🔚 Map Design	Generate TRACE report for each iteration	T/F	False					
Map Trace	Guided PAR Matching Factor	Num						
Place & Route Design	Guided PAR Report Matches	T/F	False					
IO Timing Analysis	Ignore Preference Errors	T/F	True	1				
Timing Simulation	Multi-Tasking Node List	File						
Bitstream	NCD Guide File	File						
	Path-based Placement	List	Off					
	Placement Effort Level [1-5]	Num	1					
	Placement Iteration Start Pt	Num	1	_				
	Placement Iterations [0-99]	Num	1					
	Placement Save Best Run [1-99]	Num	1					
	Remove previous design directory	T/F	True					
	Routing Delay Reduction Passes [0-100]	Num	0					
	Routing Passes [1-1000]	Num	1					
	Routing Resource Optimization [0-6]	Num	0					
	Routing method	List	Default					
	n ni .o.i	<b>T</b> ( <b>F</b>	P.1					

#### Figure 65: Quick Predefined Strategy

# Timing

The Timing strategy is a predefined strategy for timing optimization. Its purpose is to achieve timing closure. The Timing strategy uses very high effort level in placement and routing. Use this strategy if you are trying to reach the maximum frequency on your design. If you cannot meet your timing requirements with this strategy, you can clone it and create a customized strategy with refined settings for your design. This strategy might increase your place-and-route run time compared to the Quick and Area strategies.

#### Figure 66: Timing Predefined Strategy

Process	Timing Sim	Timing Simulation						
<ul> <li>Synthesize Design</li> <li>Synplify Pro</li> </ul>		isplay catalo						
Precision	Name	Туре	Value					
📰 Translate Design	Generate PUR in the Netlist	T/F	True					
4 📴 Map Design	Generate X for Setup/Hold Violation	T/F	False					
Map Trace	Hold Check Min Speed Grade	T/F	False					
Place & Route Design Place & Route Trace	Multichip Module Prefix	Text						
IO Timing Analysis	Negative Setup-Hold Times	T/F	True					
Timing Simulation	Retarget Speed Grade	Num						
Sa Bitstream	Timing Simulation Max Delay between buffers (ps)	Num						
	Verilog Hierarchy Separator	Text						
	Write Verbose Netlist	T/F	False					

# **User-Defined**

You can define your own customized strategy by cloning and modifying any existing strategy. You can start from either a predefined or a customized strategy.

## **Common Tasks**

Working with projects includes many tasks: creating the project, editing design files, modifying tool settings, trying different implementations and strategies, saving your data.

# **Creating a Project**

See "Creating a New Project" on page 5 in the "Getting Started" chapter for step-by-step instructions.

# **Changing the Target Device**

There are two ways to access the Device Selector dialog box for changing the target device:

- Double-click the device in the project File List view or right-click it and choose Edit.
- Choose Project > Device

# Setting the Top Level of the Design

If multiple top levels exist in the hierarchy of your HDL source files, you will need to set the top-level design unit. After generating the hierarchy, choose **Project > Active Implementation > Set Top-Level Unit**. Alternatively, right-click the implementation and chose this command from the pop-up menu

#### Figure 67: Top-Level Design Unit

i mixed	Name:	Timing	Category: Implementation		
LFSC3GA15E-5F256C	Location:	C:/lscc/diamond/2.2	/examples/mixed_mode/Timing		
Timing source/count_up_2.v					
source/count_up_2.v		Name	Value		
source/count_down.v	Top-Lev	el Unit	count		
source/count_down_2.v	HDL Par	ameters			
source/count_down_3.vhd	VHDL L	brary Name	work		
source/count_up.v mixed.lpf	Verilog	Include Search Path			
mixed					
🖗 Area					
I/O Assistant					
🗒 Quick					
朦 Timing					
🔄 Strategy1					

In the Project Properties dialog box, select **Value** next to **Top-Level Unit** and select the desired top level from the list.

You can also use the Hierarchy View to set the top-level. Right-click any level in the Hierarchy View and choose **Set as Top-Level Unit**.

## **Editing Files**

You can open any of the files for editing by double-clicking or by right-clicking and choosing **Open** or **Open with**.

# **Saving Project Data**

In the File menu are selections for saving your design and project data.

- Save saves the currently active item
- Save As opens the Save As dialog to save the active item
- Save All saves all changed documents
- Save Project saves the current project
- Save Project As opens the Save Project As dialog to save the active project
- Archive Project creates a zip file of the current project in a location you specify

# Chapter 6



# **Lattice Diamond Design Flow**

This chapter describes the design flow in Lattice Diamond. Running processes and controlling the flow for alternate what-if scenarios is explained. A summary of the major differences from the ispLEVER flow is included.

### Overview

The FPGA implementation design flow in Lattice Diamond provides extensive what-if analysis capabilities for your design. The design flow is displayed in the Process view.

#### 

#### Figure 68: Design Flow

# **Design Flow Processes**

The design flow is organized into discrete processes, where each step allows you to focus on a different aspect of the FPGA implementation.

**Synthesize Design** This process runs the selected synthesis tool (Synplify Pro is the default) in batch mode to synthesize your HDL design.

**Translate Design** This process converts the EDIF file output from synthesis to the Native Generic Database (NGD) format. If the design utilizes Lattice NGO netlist files, such as generated Lattice IP, the netlist will also be read into the design. The Translate Design process is not present when the Lattice Synthesis Engine (LSE) has been selected as the synthesis tool.

**Map Design** This process maps the design to the target FPGA and produces a mapped Native Circuit Description (.ncd) design file. Map Design converts a design's logical components into placeable components.

The Map Design process also generates timing analysis and simulation files when you have selected them before running Map Design:

- Map Trace creates a post-map timing report file (.tw1) that helps determine where timing constraints will not be met. This report can be viewed in Report View. In post-map timing analysis, Trace determines component delays and uses estimated routing delays. The estimation method used is based on the setting for "Route Estimation Algorithm" in the Map Trace section of the active strategy. This can be used to detect severe timing issues, such as deep levels of logic, without incurring the runtime of PAR.
- Verilog Simulation File generates a Verilog netlist of the mapped design that is back annotated with estimated timing information. This generated \_map\_vo.vo file enables you to run a simulation of your design.
- VHDL Simulation File generates a VHDL netlist of the routed design that is back annotated with estimated timing information. This generated \_map\_vho.vho file enables you to run a simulation of your design.

The timing analysis and simulation files can also be generated separately by double-clicking each one.

**Place & Route Design** This process takes a mapped physical .ncd design file and places and routes the design. The output is an .ncd file that can be processed by the design implementation tools. Auto Hold Time Correction, which is enabled by default in the active strategy, causes Place & Route to process setup and hold time requirements in a single flow and concurrently. This option improves timing closure without increasing run time.

The Place & Route Design process also generates timing and SSO analysis files when you have selected them before running Place & Route Design:

Place & Route Trace – creates a timing report (.twr) that enables you to verify timing. This report can be viewed in Report View. In post-route timing analysis, Trace analyzes path delays and reports where these occur in the design. I/O Timing Analysis – runs I/O timing analysis and generates an I/O Timing Report that can be viewed in Report View. The report is an analysis of inputs and outputs across all potential silicon to help ensure that the board design is compatible; it shows the constraints to which the board design will need to adhere.

For each input port in the design, this report shows the worst case setup and hold time requirements. For each output port, it shows the worst case min/max clock-to-out delay. The computation is performed over all performance grades available for the device and at the voltage and temperature specified in the preference file. I/O timing analysis also automatically determines the clocks and their associated data ports.

**Export Files** This process generates the IBIS, simulation, and programming files that you have selected for export:

- IBIS Model generates a design-specific I/O Buffer Information Specification model file (.ibs). IBIS models provide a standardized way of representing the electrical characteristics of a digital IC's pins (input, output, and I/O buffers).
- Verilog Simulation File generates a Verilog netlist of the routed design that is back annotated with timing information. This generated .vo file enables you to run a timing simulation of your design.
- VHDL Simulation File generates a VHDL netlist of the routed design that is back annotated with timing information. This generated .vho file enables you to run a timing simulation of your design.
- JEDEC File generates a JEDEC file for programming the device. JEDEC is the industry standard for PLD formats. In the Lattice Diamond software, JEDEC refers to the fuse map of your design for the selected device. This option is applicable only to non-volatile FPGAs such as LatticeXP, LatticeXP2, MachXO, MachXO2, MachXO3L.
- Bitstream File generates a configuration bitstream (bit images) file, which contains all of the design's configuration information that defines the internal logic and interconnections of the FPGA, as well as devicespecific information from other files. This option is applicable only to volatile SRAM-based FPGAs, such as LatticeECP/2/M, LatticeECP3, LatticeSC/M.
- PROM File generates an output file in one of several programmable read-only memory (PROM) file formats. This option is applicable only to volatile SRAM-based FPGAs, such as LatticeECP/2/M, LatticeECP3, LatticeSC/M

The files for export can also be generated separately by double-clicking each one.

## **Running Processes**

For each step in the process flow you can perform the following actions:

- Run runs the process, if it has not yet been run
- Rerun reruns a process that has already been run

- Rerun All reruns all processes from the start to the selected process
- Stop stops a running process
- Clean Up Process clears the process state and puts a process into an initial state as if it had not been run

The state of each process step is indicated with an icon to the left of the process:

- Process in initial state
- V Process completed successfully
- A Process completed with warnings, see Warning output
- X Process failed, see Error output

The Reports View displays detailed information about the process results, including the last process run. In the Tool Reports section, it shows the results of SSO analysis that has been set up in Spreadsheet View.

The Messages section shows warning and error messages and allows you to filter the types of messages that are displayed. See "Reports" on page 32.

Figure 69: Reports View of Last Process Run

Reports							
ixed				e			
	]						
Design Summary	mixed project summary						
Project	Module Name:	mixed	Synthesis:	SymplifyPro			
Project Summary	Implementation Name:	mixed	Strategy Name:	Timing			
Process Reports	Last Process:	I/O Timing Analysis	State:	Passed			
Synplify Pro                Map							
Place & Route	Target Device:	LFSC3GA15E-5F256C	Device Family:	LatticeSC			
Place & Noule Signal/Pad	Device Type:	LFSC3GA15E	Package Type:	FPBGA256			
Bitstream	Performance grade:	5	Operating conditions:	COM			
Analysis Reports	Logic preference file:	mixed.lpf					
Map Trace	Physical Preference file:	mixed/mixed_mixed.prf					
Place & Route Trace	Product Version:	3.0.0.47	Patch Version:				
I/O Timing Analysis	Updated:	2013/06/13 15:54:06					
Tool Reports	Implementation Location:	C:/lscc/diamond/3.0/examples/mixed_mode/mixed					
I/O SSO Analysis	Project File:	C:/lscc/diamond/3.0/examples/mixed_mode/mixed.ldf					
<ul> <li>Hierarchy Parsing Report</li> <li>Run BKM Check</li> <li>PIO DRC</li> <li>TCL Command Log</li> </ul>							
<ul> <li>Messages</li> <li>All Messages</li> <li>User Defined Filters</li> </ul>							
mixed timing							

## **Refreshing a Process State**

You can choose **Design > Refresh Design** to refresh the process state. For example, after running a certain process, you have made changes to a source file using a source editor outside Lattice Diamond. Then you can use Refresh Process to check the most recent state of the processes. This will also update the Hierarchy view and Reveal Inserter with any design changes.

### **Implementation Flow and Tasks**

Implementations organize the structure of your design and allow you to try alternate structures and tool settings to determine which ones will give you the best results.

You might want to try different implementations of a design using the same tool strategy, or try running the exact same implementation with different strategies to see which scenario will best meet your project goals. Each implementation has an associated active strategy, and when you create a new implementation you must select its active strategy.

To try the same implementation with different strategies you will need to create a new implementation/strategy combination. Right-click the project name in the File List and choose **Add > New Implementation**. In the dialog box, the Add Source selection allows you to use source from an existing implementation. The Default Strategy selection allows you to choose from the currently defined strategies.

lame:	verilog_vhdl2		Synthesis Tool:	Synplify Pro		
irectory:	verilog_vhdl2		Default Strategy:	Area		
ocation:	C:/lscc/diamond/3.2/examples/n	nixed_mode_SC/verilog_v	/hdl2			
Source fi	les:			Add Source 🔻	Remove Source	
	diamond/3.2/examples/mixed_n	node_SC/Test/source/co	unt.v			
C:/Iscc/	diamond/3.2/examples/mixed n	node SC/Test/source/cou	unt down.v			
C:/Iscc/	'diamond/3.2/examples/mixed_n 'diamond/3.2/examples/mixed_n 'diamond/3.2/examples/mixed_n	node_SC/Test/source/co	unt_down_2.v			
C:/lscc/ C:/lscc/ C:/lscc/	diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n	node_SC/Test/source/cou node_SC/Test/source/cou node_SC/Test/source/cou	unt_down_2.v unt_down_3.vhd unt_up.v			
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n	node_SC/Test/source/con node_SC/Test/source/con node_SC/Test/source/con node_SC/Test/source/con	unt_down_2.v unt_down_3.vhd unt_up.v			
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n	node_SC/Test/source/con node_SC/Test/source/con node_SC/Test/source/con node_SC/Test/source/con	unt_down_2.v unt_down_3.vhd unt_up.v			
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n	node_SC/Test/source/con node_SC/Test/source/con node_SC/Test/source/con node_SC/Test/source/con	unt_down_2.v unt_down_3.vhd unt_up.v			
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n	node_SC/Test/source/coi node_SC/Test/source/coi node_SC/Test/source/coi node_SC/Test/source/coi node_SC/Test.lpf	unt_down_2.v unt_down_3.vhd unt_up.v		Properties	
C:/lscc/ C:/lscc/ C:/lscc/ C:/lscc/	diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n diamond/3.2/examples/mixed_n	node_SC/Test/source/coi node_SC/Test/source/coi node_SC/Test/source/coi node_SC/Test/source/coi node_SC/Test.lpf	unt_down_2.v unt_down_3.vhd unt_up.v		Properties	

#### Figure 70: Adding a Source to a New Implementation

If you want to use the exact same source for the new and the existing implementations, make sure that the "Copy source to implementation"

directory" option is not selected. This will ensure that your source is kept in sync between the two implementations.

## **Run Management**

Use Run Manager to run different implementations. Each implementation will use its active strategy. Choose **Tools > Run Manager** or click the Run Manager button 💮 on the toolbar.-

#### Figure 71: Run Manager

ile	Window												
3	0 implementations are hidde	n. <u>Cli</u>	k to configure visible	implementations									
	Implementation < St	trateg	y> Current Step	Status	Next Step	Worst Slack	Timing Score	Worst Slack (Hold)	Timing Score (Hold)	Slice	Unrouted Net	Number Of Signals	Number Of Connection
	4 mixed <strategy1></strategy1>	•	N/A	Completed	N/A	5.366	0	0.169	0	14/7584	0	N/A	N/A
3	N/A		N/A	N/A	N/A	5.366	0	0.169	0	14/7584	0	N/A	N/A
8	N/A		N/A	N/A	N/A	5.552	0	0.141	0	14/7584	0	N/A	N/A
۶	N/A		N/A	N/A	N/A	5.486	0	0.169	0				N/A
	N/A		N/A	N/A	N/A	5.392	0	0.154	0	14/7584	0	N/A	N/A
	N/A		N/A	N/A	N/A	5.381	0	0.154	0	14/7584			N/A
	mixed250 < Strategy		N/A	Completed	N/A	2.659	0	0.181	0	14/7584			N/A
	N/A		Select All		VA	2.659	0	0.181	0	14/7584			N/A
	N/A		Unselect All		/A	2.849		0.154	0	14/7584			N/A
	N/A		Run		VA	2.697		0.167	0	14/7584			N/A
	N/A	_			VA	2.819		0.159	0				N/A
	N/A	_	Rerun		VA	2.726	0	0.153	0	14/7584	0	N/A	N/A
			Pause										
Ì			Stop										
		С	Show/Hide Imple	mentation									
		₽	Export										
			Set As Active										
			Show Report										

Run Manager runs the entire process flow for each selected implementation. If you are running multiple implementations on a multi-core system, Run Manager will distribute them so that they are executed in parallel.

You can use the Run Manager list to set an implementation as active. Rightclick the implementation/strategy pair and choose **Set as Active**.

For an implementation that uses multiple iterations of place-and-route, you can select the run that you want to use as the active netlist for further processes. Right-click the desired run, and choose **Set as Active**. Multiple place-and-route (PAR) iterations are specified in the Place & Route section of the active Strategy.

To examine the reports from each process, make an implementation active, and then select the Reports View. For multiple Place & Route iterations, the PAR report is not updated when the iteration is made active.

# **Synthesis Constraint Creation**

Synthesis constraints, in the format of the Synopsys® Design Constraint (SDC) language, can be added to a design implementation. Constraints can also be added in the Synopsys standard timing constraints format in the form of FPGA Design Constraint (FDC) files.

If you are using Synplify Pro or Precision for synthesis, the constraints will be included in an .sdc file or .fdc file. If you are using the Lattice Synthesis Engine, which is available for MachXO, MachXO2, MachXO3L, and Platform Manager designs, the synthesis will be included in an .ldc file.

To create a new synthesis constraint file, right-click the Synthesis Constraint Files folder in the File List pane and choose **Add > New File**. In the New File dialog box, select LDC Constraint Files or SDC Files from the Source Files list and give the file a name. If you selected SDC Files, the .fdc extension will be added by default, but you have the option of selecting the .sdc extension from the drop-down menu.

a <u>t</u> egories:		Source Files:	
Source		LDC Files	
Other Fi	lles	Power Calculator Files	
		Schematic Files	
		Yerilog Files	
		M VHDL Files	
		EDIF Netlist Files	
		Preference Files	
		SDC Files	
		🖹 System Verilog Files	
		Timing Preference File	
DC Files			
Na <u>m</u> e:			Ext: fdc
Location:	C:/lscc/diamo	nd/3.0/examples/attributes	Browse
[	✓ <u>A</u> dd to Imp	olementation attributes	

Figure 72: New Synthesis Constraint File

If you selected SDC Files, open the file in the Source Editor to add the constraints. If you selected LDC Constraint Files, the LDC Editor will open displaying the spreadsheet and six tabs for creating and editing synthesis constraints.

#### Figure 73: LDC Editor

ſ	S	tart Page	Reports 🗵	💰 attributes_ldc.ldc * 🛛	e e e e e e e e e e e e e e e e e e e
1		Туре	Port	Clock	Value(ns)
	1	Input	[all_inputs]	clock	1.200000
	2	Output	[all_outputs]	clock	1.200000
	3				

For detailed information about setting SDC constraints, see *Applying Lattice Synthesis Engine Constraints* and the *Constraints Reference Guide* in the Lattice Diamond online Help.

## **LPF Constraint Creation**

The logical preference file (.lpf) is the source file for storing constraints called preferences. Logical preferences in the .lpf file are used as input to post-synthesis processes. These preferences can be created using Diamond's preference-editing views. The following steps illustrate how you might assign and edit logical preferences in Lattice Diamond and implement them at each stage of the design flow.

- If desired, define some constraints at the HDL level using HDL attributes. These attributes from source files will be included in the EDIF netlist, and they will be displayed in Lattice Diamond after you run the Translate Design process.
- Open one or more of the following views to create new constraints or to modify existing constraints from the source files and save them as preferences.
  - Spreadsheet View This is the primary view for setting constraints. Set timing objectives such as fMAX and I/O timing, define signaling standards, and make pin assignments. Assign clocks to primary or secondary routing resources. Set parameters for simultaneous switching outputs for SSO analysis. Define groups of ports, cells, or ASIC blocks. Create UGROUPs from selected instances to guide placement. Establish REGIONs for UGROUPs or for reserving resources.
  - Package View Examine the pin layout of the design. Modify signal assignments and reserve pin sites that should be excluded from placement and routing. Examine SSO analysis by pin. Run PIO design rule check to check for legal placement of signals to pins.
  - Device View Examine FPGA device resources. Reserve sites that should be excluded from placement and routing.
  - Netlist View Examine the design tree by the element names of ports, instances, and nets so that the names can be used when defining preferences. Assign selected signals by dragging them to Package View. Set timing and location constraints. Create UGROUPs of logical instances to guide placement and routing. Set BLOCK preferences for selected nets.
  - Floorplan View Examine the device layout of the design. Draw bounding boxes for UGROUPs. Draw REGIONs for the assignment of groups or to reserve an area. Reserve sites and REGIONs that should be excluded from placement.

When you modify preferences in Diamond, the "Preferences Modified" indicator appears in the status bar on the right.

3. Save the preferences to the logical preference file (.lpf).

The "Preferences Modified" indicator is cleared from the status bar.

- 4. Run the Map Design process (map). This process reads the .ngd and .lpf files and produces a native circuit description file (.ncd) and a physical preference file (.prf). The .prf file is an internal file generated by the Map engine that contains preferences used by the PAR engine. The .prf file is not intended to be edited by the user because edits will be lost when it is regenerated.
- 5. Run the Map TRACE process and examine the timing analysis report. This is an optional step, but it can be a quick and useful way to identify serious timing issues in design and/or preference errors (syntax & semantic). Modify preferences as needed and save them.
- 6. Run the Place & Route Design process (par). This process reads the post-MAP .ncd file and the .prf file, and it appends placement and routing to a post-PAR .ncd file.
- 7. Open views directly or by cross-probing to examine timing and placement and create new UGROUPs. Also examine the Place & Route Trace report.
  - Timing Analysis View Examine details of timing paths. Cross probe selected paths to Floorplan and Physical Views. Create one or more timing preference files (.tpf) and experiment with sets of modified preferences for the purpose of timing analysis, using the TPF Spreadsheet View. Copy the best results to the .lpf file.
  - NCD View– Examine placement assignments. Cross probe to Physical View and open detailed views of selected blocks. Create new UGROUPs, as needed, from selected instances in NCD View.
- 8. Modify preferences or create new ones using any of the views. Save the preference changes and rerun the Place & Route Design process.

### **Simulation Flow**

The simulation flow in Diamond 1.3 or later has been enhanced to support source files that can be set in the File List view to be used for the following purposes:

- Simulation & Synthesis (default)
- Simulation only
- Synthesis only

This allows the use of test benches, including multiple file test benches. Additionally, multiple representations of the same module can be supported, such as one for simulation only and one for synthesis only.

An enhancement in Diamond 2.1 is the ability to automatically add top level signals to the waveform display in the simulator and to automatically start the simulator running.

The Simulation Wizard has been enhanced to support these changes. The Simulation Wizard will automatically include any files that have been set for simulation only or for both simulation and synthesis. The user can select the top of the design for simulation independent of the implementation design top.

This allows easy support for test bench files, which are normally at the top of the design for simulation but not included for implementation. The implementation wizard will export the design top to the simulator, along with source files, and set the correct top for the SDF file if running timing simulation.

The File List view shows an implementation's input files for simulation. This is a listing of source files and does not show design hierarchy.

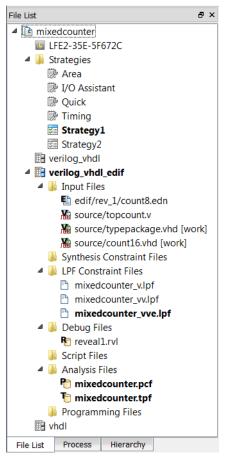


Figure 74: Input Files for Simulation

After you add a module, use the **Include For** menu to specify how the module file is to be used in the design.

When you are ready to simulate, you can export the design using the Simulation Wizard. Choose **Tools > Simulation Wizard** or click the Simulation Wizard button  $\stackrel{\text{W}}{\longrightarrow}$  on the toolbar. The wizard will lead you through a series of steps: selecting a simulation project name and location, specifying the simulator to use (if you have more than one installed), selecting the process stage to use (from RTL to Post-Route Gate-Level + Timing), and

_			
File List			
▲ It Test			
IIII LFXP10C-5F388C			
A Brategies			
🖗 Area			
🔅 I/O Assistant			
🖗 Quick			
Timing			
Strategy1			
⊿ 📑 Test			
4 퉬 Input Files			
Test/source/cour	nt.v		
Test/source/	Open		
Test/source/	Open With		
Test/source/	Open Containing Folder		
Test/source/	open containing rolder		
Test/source/	Add 🕨		
🎍 Synthesis Consti	Clone Implementation		
4 퉬 LPF Constraint F	Clone Strategy		
📑 Test.lpf	Clone strategy		
🍌 Debug Files	Exclude from Implementation		
🔺 🎍 Script Files	Remove		
iii simulationfil	Remove		
Analysis Files	Set Top-Level Unit		
Programming Fi	Select Synthesis Tool		
E verilog_vhdl2	-		
	Include for	۲	Synthesis and Simulation
	Properties		Synthesis
			Simulation
		_	Simulation

Figure 75: Include For Commands

selecting the language (VHDL or Verilog) and source files. You can also run the simulation directly from the wizard.

### Figure 76: New Simulation Project

🤔 Simulation Wize	Simulation Wizard							
Simulator Proje Enter a name stored.	ect Name e for your simulator project and specify a directory where the project file will be							
Project name:	mixed_sim							
Project location:	C:/Projects/Design examples/mixed_mode							
Simulator								
Active-HDL								
ModelSim								
8								
	< <u>B</u> ack <u>N</u> ext >	Cancel						

After you have set up the simulator project and specified the implementation stage and source files to be included, the Simulation Wizard parses the HDL and test bench. The last step is to specify the simulation top module.

#### Figure 77: Simulation Top Module

🥍 Simulation Wizard	X
Parse HDL files for simulation Parse HDL files for simulation.	
Simulation top parsing finished.	
<ul> <li>Analyzing Verilog file C:/Projects/Design examples/mixed_mode/source/count_up_2.v (VERI-1482)</li> <li>Analyzing Verilog file C:/Projects/Design examples/mixed_mode/source/count_down.v (VERI-1482)</li> <li>Analyzing Verilog file C:/Projects/Design examples/mixed_mode/source/count_down.v (VERI-1482)</li> <li>Analyzing Verilog file C:/Projects/Design examples/mixed_mode/source/count_down_2.v (VERI-1482)</li> <li>Analyzing Verilog file C:/Projects/Design examples/mixed_mode/source/count_dvn_2.v (VERI-1482)</li> <li>Analyzing Verilog file C:/Projects/Design examples/mixed_mode/source/count_up.v (VERI-1482)</li> <li>Analyzing VHDL file C:/Iscc/diamond/3.0/cae_library/synthesis/vhdl/sc.vhd (VHDL-1481)</li> <li>Restoring VHDL parse-tree ieee.std_logic_1164 from C:/Iscc/diamond/3.0/cae_library/vhdl_packages/vdbs/std/s</li> <li>Analyzing VHDL file C:/Projects/Design examples/mixed_mode/source/count_down_3.vhd (VHDL-1481)</li> <li>Restoring VHDL parse-tree ieee.std_logic_s/mixed_mode/source/count_down_3.vhd (VHDL-1481)</li> <li>Restoring VHDL parse-tree ieee.numeric_std from C:/Iscc/diamond/3.0/cae_library/vhdl_packages/vdbs/std/s</li> </ul>	
۰ III ۲	
Simulation Top Module: count	•
< <u>B</u> ack <u>N</u> ext > Cance	el

In some designs, the compile order of the HDL files passed to the simulator might result in compilation warnings. In most cases, these compilation warnings can be safely ignored. The warnings can be eliminated in one of two ways:

- The correct compilation order for the HDL files can be set in the File List view. After the correct order for the files is set manually, the files will be sent to the simulator, which will eliminate any compilation warnings.
- The correct compilation order for the HDL files can be set in the Simulation Wizard during the "Add and Reorder" step. After the correct order for the files is set manually, the files will be sent to the simulator, which will eliminate any compilation warnings.

### I/O Assistant Flow

Defining a device pinout can be a complicated process because of constraints in the PC board layout and the FPGA architecture, and it is typically done long before the entire FPGA design is complete. The I/O Assistant, a special predefined strategy within Diamond, assists you with this task, enabling you to produce an FPGA-verified pinout early on based upon PC board layout requirements.

The I/O Assistant strategy helps you select a legal device pinout and produce LOCATE and IOBUF preferences for optimal I/O placement. The only design content required to validate an I/O plan is an HDL model of the I/O ports. Details of the internal logic can be treated as a black box. The primary output of the I/O Assistant flow is a validated placement of I/O signals that can be back annotated to the logical preference file.

The I/O Assistant strategy is a read-only predefined set of properties for the design flow. The following sequential steps are typical for the I/O Assistant design flow:

- 1. Create a top-level module in HDL that describes all of the ports in the design. You can do so manually or use the I/O modules generated by IPexpress.
- 2. Make the I/O Assistant strategy active for your project. From the Strategies folder in the File list pane, right-click I/O Assistant and choose Set as Active Strategy.
- 3. Synthesize your HDL as you would normally.

If you are using Synplify Pro, Lattice Diamond will automatically pass the required attributes and header files for I/O Assistant flow when you run the Translate Design process. It will also automatically pass the required attributes and header files if you are using the integrated Lattice Constraint Engine (LSE) for a MachXO, MachXO2, MachXO3L, or Platform Manager design.

If you are running synthesis in stand-alone mode, you will need to include these attributes and header library files in the source code before synthesis. See the synthesis tool documentation for more information.

- 4. Constrain your design to add banking location preferences, I/O types, I/O ordering, and minor customizations. You can set these preferences using Spreadsheet View or you can do this manually.
  - To set the preferences in Spreadsheet View, choose Tools > Spreadsheet View and edit the I/Os.
  - To set I/O preferences manually, double-click the name of the project's logical preference file (.lpf) from the Constraints folder in the File List view.

When implementing DDR interfaces, it is recommended that you generate the required DDR modules using IPexpress along with the port definitions. This will enable the tool to check for any DDR-related rules that are being violated.

5. Run the Place & Route Design process.

The process maps and places the I/Os based on the preferences, the I/O Assistant strategy, and the architectural resources. The output is a pad report (.pad) to guide future placement and a placed and routed native circuit description .ncd that contains only I/Os.

- 6. Examine the I/O Placement results by doing one or more of the following:
  - From the Process Reports folder in the Reports window, select Signal/Pad to open the PAD Specification File and examine the pinout.
  - Choose Tools > Package View, and then choose View > Display IO Placement to view the pin assignments on the layout to cite areas for minor customization.

To view the results of timing constraints:

Run the Place & Route Trace process and open the Place & Route Trace report from the Reports window.

- Run the I/O Timing Analysis process and open the I/O Timing Report from the Reports window.
- Make any needed adjustments to the I/O preferences, as you did in Step 4.
- 8. Rerun Place & Route Design.
- 9. Repeat steps 5 through 7 as necessary to achieve your I/O placement objectives.
- From Package View, choose Design > Backannotate Assignments to copy the I/O preferences to the logical preference file, and then choose File > Save.

I/O placement preferences are written to the end of the .lpf file and will take precedence over any existing preferences that may conflict with them.

11. Create a new strategy or add an existing one. Set the strategy as the active one, and take your design through the regular flow.

### Summary of Changes from ispLEVER

Lattice Diamond is the next-generation FPGA design environment, replacing the ispLEVER tool. Although the design processes are very similar between the two environments, there are a number of improvements and differences to be aware of if you are an experienced ispLEVER user.

- Synthesize Design and Translate Design steps in Lattice Diamond replace the Build Database step in ispLEVER.
- Exporting designs to simulation is done with the Simulation Wizard in Lattice Diamond.
- Timing analysis can be performed using the Timing Analysis View in Lattice Diamond without having to re-implement the design. See "Timing Analysis View" on page 98 for more details.
- Reports from the design process steps are viewed independently of the process state. Therefore, viewing a process report will not cause a process to be rerun. In ispLEVER, viewing a report causes a process to be rerun.

# Chapter 7



# **Working with Tools and Views**

This chapter covers the tools and views controlled from the Lattice Diamond framework. Tool descriptions are included and common tasks are described.

### Overview

The Lattice Diamond design environment streamlines the implementation process for FPGAs by combining the tool control and data views into one common location. Two main features of this design environment make it easy to keep track of unsaved changes in your design and examine data objects in different view.

## **Shared Memory**

Lattice Diamond uses shared memory that is accessed by all tools and views. As soon as design data has been changed, an asterisk \* appears in the tab title of the open views, notifying you that unsaved changes are in memory.

# **Cross Probing**

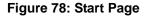
Shared design data in Lattice Diamond enables you to select a data object in one view and display it in other views. This cross-probing capability is especially useful for displaying the physical location of a component or net after it has been implemented.

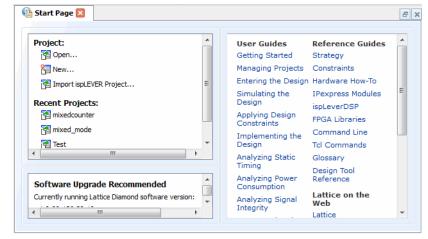
## **View Menu Highlights**

The View menu and toolbar control the display of toolbars, project views and display control. Also included in the View menu are the important project-level features Start Page, Reports and Preference Preview.

# **Start Page**

The Start Page is displayed by default when you run Lattice Diamond. The three panes within the Start Page enable you to open projects, read product documentation, and view the software version and updates. You can modify startup behavior by choosing **Tools > Options**.





The Start Page gives you quick access to recent projects and to product documentation. It can be opened, closed, detached and attached (using the icon method).

### **Reports**

The Reports view provides one central location for all project and tool report information. It is displayed by default when a project is open. Separate tabs are provided for each implementation, enabling you to view the results from multiple implementations at the same time.

#### Figure 79: Reports View

ile Edit View <u>W</u> indow			
nixed			
Design Summary			
🛛 🗾 Project		mixed project	-
Project Summary	Module Name:	mixed	Synthesis:
🛛 르 Process Reports	Implementation Name:	mixed	Strategy Name:
Synplify Pro	Last Process:		State:
🖻 🖹 Map	Target Device:	LFSC3GA15E-5F256C	Device Family:
🖻 🖹 Place & Route	Device Type:	LFSC3GA15E	Package Type:
🖻 🖹 Signal/Pad	Performance grade:	5	Operating conditions:
🗋 Bitstream	Logic preference file:	mixed.lpf	
Analysis Reports	Physical Preference file:	mixed/mixed_mixed.prf	
Map Trace	Product Version:	2.2.0.85	Patch Version:
<ul> <li>Place &amp; Route Trace</li> <li>I/O Timing Analysis</li> </ul>	Updated:	2013/04/12 08:31:55	
I Tool Reports	Implementation Location:	C:/lscc/diamond/2.2/exa	amples/mixed_mode/mixed
I/O SSO Analysis	Project File:	C:/lscc/diamond/2.2/exa	amples/mixed_mode/mixed.ldf
Hierarchy Parsing Report			
Run BKM Check			
PIO DRC			

The Design Summary pane of the Reports view is organized into Project, Process Reports, Analysis Reports, Tool Reports, and Messages. The different file icons indicate whether a report has been completed (green check mark), has never been generated (blank note paper), or is out of date (orange question mark). Select any item to see its report.

The Reports view also allows you to filter the messages from the implementation processes so that you can find the most useful messages out of what may be a very large number. To set up a filter, choose All Messages on the left. All the messages from the implementation processes are shown. Select message types, such as process, severity level, or message ID, that you want to hide, leaving a much smaller number to study. You can also save the settings for later use. See "Reports" on page 32 for more details about message filtering.

The Reports view is the primary, central view for all process report information. It can be opened, closed, detached and attached (using the icon method).

## **Preference Preview**

Preference Preview shows the design's logical preferences as they exist in shared memory, which includes unsaved preferences as well as those in the logical preference file (.lpf). When you first open Lattice Diamond, Preference Preview shows the contents of the active .lpf file. As you use preference views such as Spreadsheet View and Package View to add or modify constraints, Preference Preview reflects those changes. What you see in Preference Preview will be reflected in the .lpf file when you use the Save command.

### Figure 80: Preference Preview

🏭 Preference Preview 🗵	B >
BLOCK RESETPATHS ; BLOCK ASYNCHPATHS ;	
FREQUENCY PORT "dock" 120.000	)0 MHz ;

The Preference Preview is a read-only view of logical preferences and can be opened, closed, detached and attached (using the attach button).

## Tools

The entire FPGA implementation process tool set is contained in Lattice Diamond. You can run a tool by selecting it from the Tools menu or toolbar.

This section provides an overview of each of these tools. More detailed information is available in the user guides, which you can access from the Start Page or from the Lattice Diamond Help. Detailed descriptions of external tools can be found in their product documentation.

## **Spreadsheet View**

Spreadsheet View provides an interactive spreadsheet format for viewing and assigning design constraints. Its collection of preference sheets enables you to assign preferences such as PERIOD, FREQUENCY, I/O timing, and LOCATE to optimize placement and routing. Preferences can also be set for SSO Analysis and clock jitter.

#### Figure 81: Spreadsheet View

Eile Edit View Design Window Help									
1 <b>6</b> 1 X 6 6 16 91									
Preference Name	Preference Value	9							
Junction Temperature (Tj)(C) Voltage (V) SYSTEM_JITTER(ns) <b>4 Block Path</b>	85 1.140 Default								
Block Asynchpaths Block Resetpaths Block RD During WR Paths Block InterClock Domain Paths Block Jitter	ON OFF OFF OFF								
✓ sysConfig PERSISTENT	OFF								
CONFIG_MODE DONE_PULL DONE_EX	NONE ON OFF								
MCCLK_FREQ CONFIG_SECURE WAKE_UP COMPRESS_CONFIG	1 OFF 21 OFF								
✓ User Code	011								
UserCode Format UserCode	Binary 000000000000000000000000000000000000	000000000000000000000000000000000000000	00000						
▲ Derating     VCC     VCC1P2     VCCAUX     VCCIO     Global Set/Reset Net	-5% -5% -5% -5%								
Port Assignments Pin Assignments	Clock Resource	Route Priority	Cell Mapping	Global Preferences	Timing Preferences	Group	Misc Preferences		

The Port and Pin Assignments sheets allow you to view I/Os by signal or pin attributes and use the Assign Pins or Assign Signals functions to make assignments. Custom columns are also available on the Port and Pin Assignments sheet. Custom columns enable you to add your own information, such as notes for specific signals or pins or design data for third-party tools. You can create an unlimited number of custom columns, and you can include the column when you export to a Lattice CSV file or a Pin Layout File. Right-click any column heading to add a custom column.

#### Figure 82: Adding a Custom Column

🖉 Spread	sheet View 🛛	5 ×
D_TYPE	PULLMODE	Fit Column Hide Column
CMOS25	UP	Insert Custom Column

As soon as the target device has been specified, Spreadsheet View enables you to set global preferences. After synthesis and translation, it allows you to explore other devices of the same family for possible pin migration, as explained in "Pin Migration" on page 138.

After synthesis and translation, all of the preference sheets become available for editing.

Diamond enables you to export all Spreadsheet View preferences to a single LPF file for later reference. Use the command **File > Export > All SSV Preferences to LPF**. The LPF file of exported preferences includes not only the assignments you have made, but all default preferences as well. It also includes any assignments that you have not yet saved.

### **Port Assignments**

The Port Assignments sheet provides a signal list of the design and shows any pin assignments that have been made. It enables you to assign or edit pin locations and other attributes by entering them directly on the spreadsheet. It also enables you to assign pins in the Assign Pins dialog box by right-clicking selected signals and selecting **Assign Pins** from the pop-up menu.

Spreadshe										. 🗆 🔜
	<u>/</u> iew <u>D</u> esign <u>W</u> indow <u>I</u>	<u>H</u> elp								
	Nâme	Group By	Pin	BANK	VREF	ΙΟ ΤΥΡΕ	PULLMODE	SLEWRATE	IMPEDANCE	DRIVE
1	All Ports	N/A	N/A	N/A	N/A	10_1112	TOLLMODE	N/A	INTEDATIVE	Diave
1.1	🔺 🖿 Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1.1	🗈 clk	N/A			N/A	LVCMOS25	UP	NA	OFF	NA
1.1.2	rst	N/A			N/A	LVCMOS25	UP	NA	OFF	NA
1.2	🖌 🦪 Output	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	<pre></pre> c_down[0]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.2	<pre></pre> c_down[1]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.3	<pre></pre> c_down[2]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.4	c_down_2[0]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.5	c_down_2[1]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.6	c_down_2[2]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.7	c_down_3[0]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.8	c_down_3[1]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
1.2.9	c_down_3[2]	N/A			N/A	LVCMOS25	UP	SLOW	OFF	8
•	III						115	0.011	077	•
Port Assi	gnments Pin Assignments	Clock Resource	Rout	e Priority	Cell Mapp	ing Global Pr	eferences Tin	ning Preferences	Group Mi	sc Preference

Figure 83: Spreadsheet View Port Assignments

By default, the ports are grouped by direction—Input, Output, and Bidirectional—plus an "Others" category that includes user-defined port groups. This allows you to quickly focus, for example, on Output ports by closing the display of Input ports or vice versa. If you prefer to display the port rows in a single un-grouped alphabetical list, clear the "Show Group Row(s)" selection from the View > Display Group Row(s) menu

### **Pin Assignments**

The Pin Assignments sheet provides a pin list of the device and shows the signal assignments that have been made. It enables you to edit signal assignments or assign new signals by right-clicking selected pins and selecting **Assign Signals** from the pop-up menu.

	preadsheet View *								
ile 🔲	Edit View Design Window Help								
	140004 1 6	Pin	Pad Name	Dual Function	Polarity	IO_TYPE	Signal Name	Signal Type	•
ň	1	Bank1	N/A	N/A	N/A	N/A	N/A	N/A	=
94 92	1.1	A10	FIO:PT27B	PCLKC1_0	Ν				
	1.2	A11	FIO:PT33B	D0	Ν				
	1.3	A12	FIO:PT35B	D2	Ν	LVCMOS25	c_down[0]	Output	
	1.4	A13	FIO:PT36D	D4	Ν	LVCMOS25	c_down[1]	🥌 Output	
	1.5	A14	FIO:PT43C	LDCN	Р	LVCMOS25	c_down[2]	Output	
	1.6	A15	FIO:PT43D	HDC	Ν	LVCMOS25	c_down_2[0]	🥌 Output	
	1.7	B11	FIO:PT33A	QOUT	Р	LVCMOS25	c_down_2[1]	🥌 Output	
- 111	1.8	B12	FIO:PT36C	D3	Р	LVCMOS25	c_down_2[2]	🥌 Output	
A- 100	1.9	B14	FIO:PT41A	CS1		LVCMOS25	c_down_3[0]	🥌 Output	
	1.10	C10	FIO:PT27A	MPICLK/PCLKT1_0	Р	LVCMOS25	c_down_3[1]	🥌 Output	
2	1.11	C11	FIO:PT37B	D6	Ν	LVCMOS25	c_down_3[2]	Output	
	1.12	C12	FIO:PT35A	D1	Р	LVCMOS25	c_up[0]	Output	
	1.13	D8	FIO:PT28C	RDY		LVCMOS25	c_up[1]	Output	
	1.14	D10	FIO:PT37A	D5	Р	LVCMOS25	c_up[2]	Output	
	1.15	D12	FIO:PT37D	WRN	Ν	LVCMOS25	c_up_2[0]	Output	
	1.16	D13	FIO:PT39A	RDN	Ρ				~
	Port As	signments	Pin Assignments	Clock Resource R	oute Priority	Cell Mapping	Global Preference	s Timing Pref	e ∢

**Figure 84: Spreadsheet View Pin Assignments** 

By default, the rows of pins are grouped by bank number. You can also view the differential pin pair groupings within each bank by selecting "Show Differential Group Row(s)" from the View > Display Group Row(s) menu. If you prefer to display the rows of pins in a single un-grouped alphabetical list, clear the "Show Group Row(s)" selection from the View > Display Group Row(s) menu.

### **Clock Resource**

The Clock Resource sheet enables you to apply a clock domain to the device's primary or secondary clock or prohibit the use of primary and secondary clock resources to route the net. For LatticeECP2 devices, it

enables you to use edge clock resources. For LatticeECP3 devices, it enables you to assign a secondary clock to a clock REGION that has already been defined.

### **Route Priority**

The Route Priority sheet enables you to set the PRIORITIZE preference, which assigns a weighted importance to a net or bus. To set this preference, drag the desired nets from Netlist View to the Route Priority sheet. You can then select a priority value for each net. Values range from 0 to 100.

### **Cell Mapping**

The Cell Mapping sheet enables you to set the USE DIN and USE DOUT cell preferences for flip-flops in your design. The PIO Register column allows you to set the register to True or False. The True setting moves registers into the I/Os. The False setting moves registers out of the I/Os. To set these preferences, drag the desired registers from Netlist View to the Cell Mapping sheet.

### **Global Preferences**

The Global Preferences sheet enables you to set preferences that affect the entire design, such as junction temperature and voltage; BLOCK preferences applied to all paths of a particular type; and USERCODE. Also included in the Global sheet are sysCONFIG preferences for FPGA devices that support the sysCONFIG configuration port.

### **Timing Preferences**

The Timing Preferences sheet displays all timing preferences that have been set in the design, including BLOCK preferences for specific nets, FREQUENCY, PERIOD, INPUT\_SETUP, CLOCK\_TO\_OUT, MULTICYCLE, and MAXDELAY. You can create a new timing preference by double-clicking the preference name, which opens the dialog box. To modify an existing timing preference, double-click the preference name, edit the information in the dialog box, and click the Update button.

### Group

The Group sheet displays any groups that have been created and enables you to define a new cell, port, or ASIC group or create a new universal group (UGROUP). Double-click the group type to open the dialog box and create a new group preference. To modify an existing group preference, double-click the group name, edit the information in the group dialog box, and click the Update button.

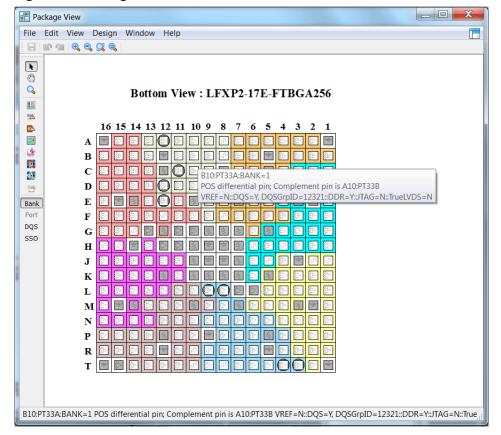
### **Misc Preferences**

The Miscellaneous sheet enables you to define REGIONs, assign Vref locations, and reserve resources by setting a PROHIBIT preference. To set a new miscellaneous preference, double-click the preference type to open the dialog box. To modify an existing miscellaneous preference, double-click the preference name, edit the information in the dialog box, and click the Update button.

# **Package View**

Package View shows the pin layout of the target device and displays the assignments of signals to device pins. Package View interacts with Netlist View for assigning pins, enabling you to drag selected signals to the desired locations on the pin layout to establish LOCATE preferences. Each pin that is assigned with a LOCATE preference is color-coded to indicate the port direction of the related signal port. Package View allows you to edit these assignments, and it allows you to reserve sites on the layout that you want to exclude from placement and routing.

After synthesis and translation, Package View allows you to explore other devices of the same family for possible pin migration, as explained in "Pin Migration" on page 138.



#### Figure 85: Package View

As you move your mouse pointer over the layout, pin descriptions and locations are displayed in tool tips and in the status bar. The **View > Show Differential Pairs** command displays fly wires between differential pin pairs and identifies the positive differential pins.

To filter the display of banks and pin types, click the Pin Display Selection button is on the toolbar. The Pin Display Selection dialog box allows you to clear the selections of banks and pin types so that only those that you want to view are displayed.

Pin Display Selection		P X
Visible Banks          V       Bank0         V       Bank1         V       Bank2         V       Bank3         V       Bank4         V       Bank5         V       Bank7         V       Bank8         V       Others	General Purpose          Image: Constraint of the second state o	Test and Programming/Configuration JTAG SysCONFIG SERDES Block SERDES Power Supplies VCC/VCCIO/VCCAUX/SERDES GND/GNDIO/VSS
Select	All Clear All Close	Help

Figure 86: Pin Display Selection Dialog Box

Package View is available as soon as the target device has been specified.

## **Device View**

Device View provides a categorized index of device resources based on the target device. Statistics cover System PIOs, User PIOs, PFUs, PFFs, sysDSP blocks, sysMEM blocks, IOLOGIC, PLL/DLLs, and other embedded ASIC blocks. Device View enables you to use the Prohibit command to reserve sites that you want to exclude from placement and routing.

From Device View, you can cross probe selected resources to their sites in Package View, Floorplan View, and Physical View.

Device View is available as soon as the target device has been specified.

#### 💭 Device View - • • File Edit View Design Window Help 🔚 🗠 🖄 📩 👗 Find... Filter... Design ٠ ▲ 💭 Device LatticeXP: LFXP3C-PQFP208 DQS: 10 IOL: 160 PCS Blocks: 0 PFF Slices: 1152 PFU Slices: 384 PLL/DLL: 2 Clock Dividers DLLDELA PLL3\_R10C1 PLL3\_R10C26 DCC/DCS: 8 Image: SysDSP Blocks: 0 Image: SysMEM Blocks: 6 🖻 🗾 Others PIO Cells: 160 APIO: 0

#### Figure 87: Device View

## **Netlist View**

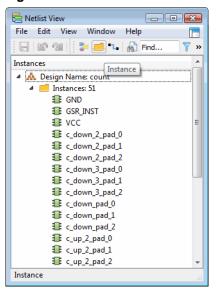
Netlist View displays the design elements of the post-synthesis native generic database (NGD) netlist. The NGD is a binary speed-optimized data structure that is used by the system to browse the logical netlist.

Netlist View organizes the netlist by ports, instances, and nets, and it provides a toolbar button and design tree view for each of these categories to make it easier to create timing or location preferences.

#### 🔚 Netlist View - • • File Edit View Window Help 🕞 🕼 🚳 🕴 🛃 🖬 😘 🚮 Find... 🍸 » Ports Ports 4 🔥 Design Name: count 🔺 🐉 Ports 🔺 📄 Input Clock 垫 clk 📄 rst Bidir Output a [2 c\_down[2:0] c\_downZ0Z\_2 c\_down\_0 c\_down\_1 c\_down\_2[2:0] c\_down\_3[2:0] c\_down\_3\_0 c\_down\_3\_1 c\_down\_3\_2 Ports

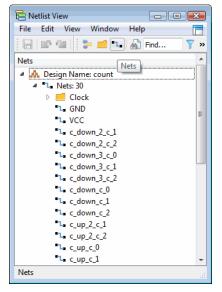
Figure 88: Netlist View Ports Design Tree

Each design tree view is equipped with utilities for filtering the list and searching for elements.



#### Figure 89: Netlist View Instances Design Tree





From Netlist View, you can drag selected signals to Package View to assign them, drag selected nets to Spreadsheet View's Route Priority sheet to prioritize them, and drag registers to the Cell Mapping sheet to specify registers for flip-flops. You can use the right-click menu to set timing preferences for selected nets and to create logical groups from selected instances.

Netlist View is available after synthesis and translation.

## **NCD** View

NCD View provides a categorized index of synthesized design resources and consumption based on the target device and the in-memory native circuit description (NCD) database. Statistics cover System PIOs, User PIOs, PFUs, PFFs, sysDSP blocks, sysMEM blocks, IOLOGIC, PLL/DLLs, and other embedded ASIC blocks.

NCD View is organized by nets and instances and provides a toolbar button and design tree view for each of these categories.

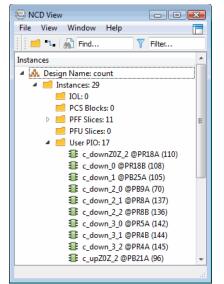
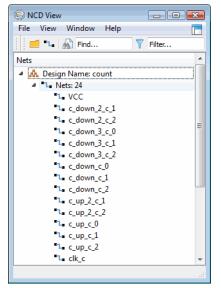


Figure 91: NCD View Instances Design Tree





Each design tree view is equipped with utilities for filtering the list and searching for elements.

From NCD View, you can create a new UGROUP preference from selected instances. You can also access schematic or tabular detailed views for selected instances.

NCD View is available after a successful run of Place & Route.

## **IPexpress**

IPexpress is a collection of functional modules that can be used to generate Verilog or VHDL source for use in your design. Modules are functional blocks of design that can be reused wherever that function is needed. They are optimized for Lattice device architectures and can be customized. Use these modules to speed your design work and to get the most effective results.

Many basic modules are included in IPexpress. They provide a variety of functions including I/O, arithmetic, memory, and more. A recommended way to use IP express is to select the import module option which will include an .ipx file in your source list. This file can be used to regenerate the module for any changes.

Choose **Tools > IPexpress** to see the full list of available modules.

Hexpress				- 0 🔀
File View Help				
🖳 🌽 🖏 🎓 😫 🛍				
Name	DDR 4.2			
<ul> <li>Module</li> <li>Architecture_Modules</li> </ul>	Macro Type:	Module	Version:	4.2
4 🖄 IO	Module Name:	DDR	, 	
録 DDR 録 DDR_GENERIC	Project Path:	diamond/1.0/examples/mixe	ed_mode	Browse
碑 DDR_MEM 碑 DQS	File Name:			
🐺 GDDR_7:1	Module Output:			<b>*</b>
_ 🙀 SDR	Device Family:			
疑定 DLL 疑定 Digital CDR	Part Name:			
🐙 Dynamic Bank Controller	Please enter a file	name!		
好 EFB 好 ORCAstra				Customize
PCS				
Power Controller				
System_Bus				
Tag Memory				
4 資 Arithmetic_Modules 🔹	Configuration	About		
A Basely	Configuration	About		
Ready				H.

#### Figure 93: IPexpress

In addition to the included modules, an IP Server button is provided that enables you to connect to the Lattice IP server, explore the available IP, and select those that you would like to download and install. In Diamond 2.1, IPexpress indicates whether an IP from the IP server is compatible with the running software release. The three states are compatible, incompatible, or unsupported. Lattice IP can be purchased or used in a trial mode. Refer to the Lattice Web site for a list of the available IP functions. For more information, see the *IPexpress Module Reference Guide*.

## **Platform Designer**

The Lattice Diamond Platform Designer enables you to use the Platform Manager 2 device, or a MachXO2 device with external analog sense and control (ASC) devices, to create a complete hardware system. Each Platform Manager 2 device is made up of an analog sense and control block and an FPGA block. These two blocks provide the necessary programmable building blocks to enable integration of various combinations of hardware management functions into a single chip.

Platform Designer's integrated design environment allows you to configure the device, implement the hardware management algorithm, simulate, assign pins, and finally generate the JEDEC files required to program and configure the device on the circuit board. It also allows you to import other HDL files to integrate other desired functions.

	21   6	R, Q, Q,							
Global Components Tan Controller		ASC Device	Pins Monitor / Trim	VMON Schematic Net Name	Nominal Profile 0 (V)	Logical Name	Type Over V/ Under V	Trip Point Selection (Volts : %)	Glitch Filter
Fault Logger Hot Swap	1	ALL	VMONs / Trims				OV/UV	+/- 0.00%	64 us 🔹
PMBus Adapter	2	ASC0	VMON1/TRIM1	ASC0_VMON1	0.668	ASC0_VMON1_A	ov 👻	0.668 : +0.0% -	64 us
Analog	3	ASC0	VMON1/TRIM1	ASC0_VMON1	0.668	ASC0_VMON1_B	UV 👻	0.668 : +0.0% 🔹	64 us
Current	4	ASC0	VMON2 / TRIM2	ASC0_VMON2	0.668	ASC0_VMON2_A	ov 🗸	0.668 : +0.0% 🔹 🗸	64 us 🕓
Temperature	5	ASC0	VMON2 / TRIM2	ASC0_VMON2	0.668	ASC0_VMON2_B	UV 👻	0.668 : +0.0% 🔹	64 us
Control	6	ASC0	VMON3	ASC0_VMON3	0.668	ASC0_VMON3_A	ov 🗸	0.668 : +0.0% 🔹	64 us 🕙
Ports & Nodes	7	ASC0	VMON3	ASC0_VMON3	0.668	ASC0_VMON3_B	UV 👻	0.668 : +0.0% 🔹 👻	64 us
Logic Build	8	ASC0	VMON4	ASC0_VMON4	0.668	ASC0_VMON4_A	ov 🗸	0.668:+0.0% -	64 us 🕓
bulla	9	ASC0	VMON4	ASC0_VMON4	0.668	ASC0_VMON4_B	UV 👻	0.668 : +0.0% 🔹 👻	64 us
	10	ASC0	VMON5	ASC0_VMON5	0.672	ASC0_VMON5_A	ov 🗸	0.672:+0.0% -	64 us
	11	ASC0	VMON5	ASC0_VMON5	0.672	ASC0_VMON5_B	UV 👻	0.672:+0.0% -	64 us
	12	ASC0	VMON6	ASC0_VMON6	0.672	ASC0_VMON6_A	ov 🗸	0.672:+0.0% -	64 us 🔹
	13	ASC0	VMON6	ASC0_VMON6	0.672	ASC0_VMON6_B	UV -	0.672:+0.0% -	64 us
	14	ASC0	VMON7	ASC0_VMON7	0.672	ASC0_VMON7_A	ov 🗸	0.672:+0.0% -	64 us
	15	ASC0	VMON7	ASC0_VMON7	0.672	ASC0_VMON7_B	UV 👻	0.672:+0.0% -	64 us
	•								÷.

**Figure 94: Platform Designer** 

You can launch Platform Designer by creating a new project in Diamond that targets either a Platform Manager 2 device or a MachXO2 device with external ASCs. You can also launch Platform Designer by opening an existing Platform Designer project file (.ptm).

For more information, see "Designing with Lattice Diamond Platform Designer" in the Diamond online Help or refer to the *Platform Designer User Guide*.

# **Clarity Designer**

Like IPexpress, Clarity Designer is an easy way to use a collection of functional blocks from Lattice Semiconductor. But Clarity Designer doesn't just customize intellectual property (IP). It actually creates a package of modules, including wiring the modules together and creating placement constraints for DDR and PCS pins. This package can contain a single module or the entire design.

The resulting Clarity Designer module can be used to aid with developing the board design and firmware while developing the FPGA design, cutting the total development time.

Clarity Designer is used with ECP5 only.

Clarity Designer has three major tabs:

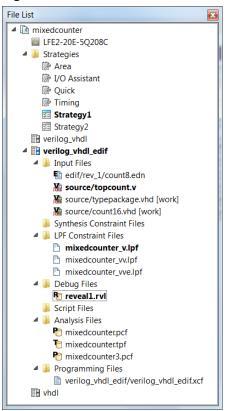
- Catalog provides a list of the available modules and IP. Use Catalog to select the modules you want in the project and to customize them. You can also download more IP from the Lattice Web site.
- Builder helps you make connections between your customized modules.
- Planner provides a graphic way to place DDR pins and PCS components before synthesis. You can run design rule checks on the placement while still in the earliest stages of designing.

## **Reveal Inserter**

Reveal Inserter allows you to add debug information to your design to allow hardware debugging using Reveal Analyzer. Reveal Inserter enables you to select the design signals to use for tracing, triggering, and clocking. Reveal Inserter will automatically generate the debug core, and insert it into a modified design with the necessary debug connections and signals. Reveal Inserter supports VHDL, Verilog, and EDIF sources. Mixed-HDL designs are represented by the synthesis EDIF netlist. After the design has been modified for debug, it is mapped, placed and routed with the normal design flow in Lattice Diamond.

The File List Debug file folder contains the debug files for Reveal Inserter.

One or none of the debug files can be active at a time. If no debug file is active, hardware debug will not be inserted into the design when it is implemented.



#### Figure 95: File List View with Reveal Debug Project File

### Launching Reveal Inserter

To launch Reveal Inserter, choose **Tools > Reveal Inserter** or click the Reveal Inserter button in on the toolbar. When Reveal Inserter is launched, the debug file it uses will depend on the following conditions:

- If an active debug file exists, it will be used. An inactive debug file will be used if it is the only one available.
- If an active or inactive debug file in the File List Debug folder is doubleclicked, it will be used.
- If multiple debug files exist and no debug file is set as active, a dialog box will enable you to select one of the inactive debug files.
- If no debug files exist, Reveal Inserter will use a default configuration.

The sections of Reveal Inserter include Trace Signal Setup and Trigger Signal setup views.

### Setting up and Inserting Debug

A Debug menu becomes available after Reveal Inserter is launched. It includes controls for managing cores, managing trace signals, running DRC, and inserting debug.

<u>V</u> iew <u>D</u> ebug <u>W</u> indo	w <u>H</u> elp		
Trigger Output	₽ ×		
		Sample Clock clk Implementation EBR   1 EBR Buffer Depth 1024 Timestamp 11 Bits Sample Enable Sample Enable Sample Enable Sample Capture Mode Sample Capture	
		Sample Enable  Active High  Include trigger signals in trace data	•
III	4	Trace Signal Setup Trigger Signal Setup Wishbone Master Setup	

### Figure 96: Reveal Inserter Trace Signal Setup

### Figure 97: Reveal Inserter Trigger Signal Setup

Reveal Inserter - C:/lscc/diamond/2.0/e	oles/counta_reveal/reveal_rvl/co	unta.rvl *		
<u>File View Debug Window H</u> elp				
Trigger Output	Trigger Unit			
count_rvl_LA0/reveal_debug_count_				_
		(MSB:LSB) Operator	Radix Value	_
	1 TU1 c[2:0]	== *	Bin • 110	
	Add Remove	Default	t Trigger Radix Bin	• E
•	Trigger Expression			
		RAM Sequence	ıx Sequer Max Even	
	name Expressi	Type Depth	Depth Counter	
	1 TE1 TU1	1 EBR 🔻 1	1 • 1 •	
		_		
	Add Remove			
	Event Counter			
< III				
	Trace Signal Setup Trigger S	ignal Setup Wishbone Ma	ster Setup	

See the Reveal User Guide for more information on setting up debug information with Reveal Inserter.

After you have your debug set up, choose **Debug > Insert Debug** or click the Insert Debug button on the vertical toolbar into your design. This will set the current debug file as active.

Ŷ • 🖄 • 🔒 🗿 🖨 🛛 🖄 🖇 Ĉ Ĉ.	Debug Tools Window Help
	Au ≈ ↔ ☆ ₩ ₩ ₩ ₩ 10 10 10 10 10 10 10 10 10 10 10 10 10
Trigger Output & X	Trigger Unit
count_rvl_LA0/reveal_debug_count_rvl	name     Signals (MSB:LSB)     Operator     Radix     Value       1     TU1     c[2:0]     == • Bin • 110       Add     Remove     Default Trigger Radix     Bin •
Insert Debug to Design         Please select core(s) you want         Image: Count_rvl_LA0         Image: Count_rvl_LA0 <td>gn project Help RAM Sequence &amp; Sequer Jax Even Type Depth Depth Counter 1 1 EBR • 1 1 • 1 • Remove</td>	gn project Help RAM Sequence & Sequer Jax Even Type Depth Depth Counter 1 1 EBR • 1 1 • 1 • Remove
	Event Counter  Event Counter Value  Trace Signal Setup  Triager Signal Setup  Vishbone Master Setup

#### Figure 98: Insert Debug

When the design is fully implemented and programmed, you can run Reveal Analyzer to debug your design.

## **Reveal Analyzer**

After you generate the bitstream or JEDEC file, you can use Reveal Analyzer to debug your FPGA circuitry. Reveal Analyzer gives you access to internal nodes inside the device so that you can observe their behavior. It enables you to set and change various values and combinations of trigger signals. After the specified trigger condition is reached, the data values of the trace signals are saved in the trace buffer. After the data is captured, it is transferred from the FPGA through the JTAG ports to the PC.

To launch Reveal Analyzer, choose **Tools > Reveal Analyzer** or click the Reveal Analyzer button in on the toolbar. The Reveal Analyzer Startup Wizard allows you to use an existing Reveal Analyzer file or create a new

one. If there is only one Reveal Analyzer project file in your implementation, it will automatically bypass the wizard and open that file. To make a new file, use the menu New File command.

Figure 99:	Reveal	Analyzer	Setup	Wizard
------------	--------	----------	-------	--------

Create a new file	untitled	USB
	Multiple Device in JTAG Chain	
USB port:	· · · · · · · · · · · · · · · · · · ·	Detect
Debug device:	· · · · · · · · · · · · · · · · · · ·	Scan
RVL source:		Browse
Open an existing file		
File name:	d/2.0/examples/counta_reveal/reveal_rvl/counta.rva 💌	Browse

The Reveal Analyzer view consists of a Trigger Setup view and a Waveform view. In the Trigger Setup view are areas displaying the Trigger Unit, Trigger Expression, Trigger Options and Trigger Position. Also in this view are controls to select which core to use to enable for triggering the analyzer.

When you choose **Run**, the analyzer connects to the hardware, configures the debug logic, and waits for the trigger conditions. Once triggered, the data is uploaded to the analyzer. The Run command also switches the display to the Waveform view.

🥨 Re	eveal Ar	nalyzer - C:/lscc/diam	nond/2.0/examples/	counta_reveal/re	veal_rvl/counta.	va			X
		<u>D</u> esign <u>W</u> indow	<u>H</u> elp						
	⊕, ⊖,								
	Ready	🔯 🔟 📝 count	_rvl_LA0						
	Trigger I	Unit							
	Name	Si	gnals (MSB:LSB)	C	perator Radix		e	1	
	TU1	c[2:0]	-	=	= • Bin •	110			-
ſ	Trigger l	Expression							
	Name		Expression		Sequence Dept	n Max Sequenc	e Depth	Max Event Counter	
	🔽 TE1	TU1			1	1	1		
14									
	Trigger	·				osition: 64/1024			
		E: OR All	n n			elected: Pre-Trig			
		Per Trigger: 1024 •	5			selected: •		•	
	Number	of Trigger: 1 🍦 (n	128)		E Final	Event Counter: 1			
LA	A Trigger	LA Waveform	SERDES Registers	Wishbone Registe	ers				
_					and the second				3

### Figure 100: Reveal Analyzer

The Waveform view has controls for running, zooming and window controls in the menu and toolbar areas.



0, 0, <u>0</u> ,											
Completed 🔝 🔟 🗹 top_LA	0										
Bus/Signal	Data	3710 0:896	3774 0:960	3838 0:102	3902 4 0:1088	3966 0:1152	4030 0:1216	4094 0:1280	62 0:1344	126 0:1408	190 0:1472
🖩 stimstate	pk_ready	(	startpk	X				pk_finish	_end		
start	1										
≅ hand_type	zero 💙			zen	0				one	ġ.	
pb_done	0							1			
sort_done	0		I				1				
E- stimcount	1 💙										
🗄 gmstate	wait_cards	wa	it_hand2	<u> </u>	x						
e chstate	idle	sort	*		dle 👔 🗶	sort	×.	X		idle	
€ uut/gamesm/nxstate	0101 🖌	(	0100	X	101 🐰			1111			
uut/ch/sort/sort_loop_done	1 🗸	XXXXXXX	X	1	1		010		1		
uut/gamesm/bet_amount_int		(									
game_ready	1			1							
- bet	0										
deal	0			ļ							
finish	0										
discard	0										
uut/ch/bj_mux_done	0			- į							
uut/ch/phm_ready	1										
uut/ch/dhm_ready	1										
uut/hand_ready	1		mmm	ານແມ	- 11		mm	mm			
U						111					

Right-clicking in the waveform area brings up a pop-up menu with selections for changing the cursor mode for panning, zooming or selecting plus selections for managing cursors, changing the clock period and exporting waveforms.

Completed 🛛 🔝 🔟 🗹 top_LA												
Bus/Signal	Data	3710 0:896	3774 0:960	3838 0:1024	3902 0:1088	3966 0:1152	4030 0:1216	4094 0:1280	62 0:1344	126 0:1408	190 0:1472	
e stimstate	pk_ready	s	tartpk	<u> </u>		1		pk_finish	end			
start	1											
≅ hand_type	zero 👻			zero					one			
pb_done	0							1				
sort_done	0		1			Pan Mode						
E stimcount	1 💙					Select Mode						
e gmstate	wait_cards	wai	t_hand2		8	Zoom Mode						
e chstate	idle	sort	XX	X idle	XXX	Add Cursor Remove Cur	sor	X		idle		
🗄 uut/gamesm/nxstate	0101 🖌	(	0100	0101	XX	Clear All Cur		11111				
i uut/ch/sort/sort_loop_done	1 👻			1		Zoom	•		1			
uut/gamesm/bet_amount_int						Set Clock Pe						
game_ready	1			1		Export Wave	eform	ſ				
bet	0											
deal	0			Ļ								
finish	0											
discard	0											
uut/ch/bj_mux_done	0			ļ								
	1											
	1											
uut/hand_ready	1		-000000					เบบบบ				

#### Figure 102: Reveal Analyzer Waveform Cursor Controls

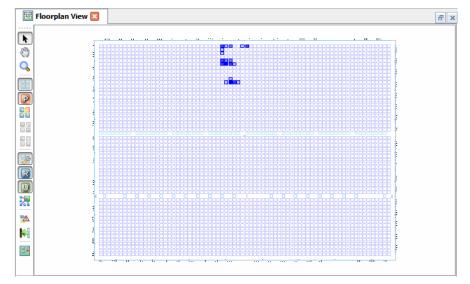
The Data column in the view shows the data for the active cursor. The Reveal Analyzer supports multiple cursors which can be added, removed and position changed within the waveform. Selecting the cursor and dragging it will produce a rubber band effect which can be used for measuring time intervals.

See the Reveal User Guide for more information on using Reveal Analyzer.

## **Floorplan View**

Floorplan View provides a large-component layout of your design. It displays user constraints from the logical preference file (.lpf) and placement and routing information. All connections are displayed as fly-lines.





Floorplan View allows you to create REGIONs and bounding boxes for UGROUPs and specify the types of components and connections to be displayed. As you move your mouse pointer over the floorplan layout, details are displayed in tooltips and in the status bar:

- the number of resources for each UGROUP and REGION
- the number of utilized slices for each PLC component
- the name and location of each component, port, net, and site

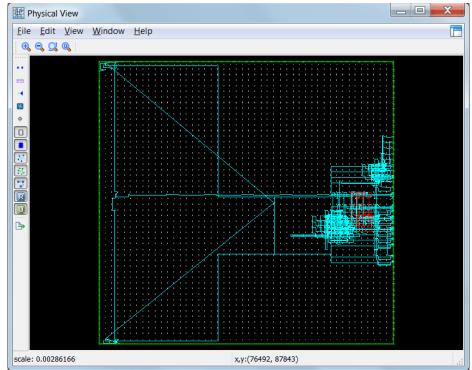
If your design uses the incremental design flow, Floorplan View will display partitions that are included in the design and allow you to edit them. For more information, refer to "Using Incremental Design Flow" in the Diamond online Help.

Floorplan View is available as soon as the target device has been specified.

### **Physical View**

Physical View provides a read-only detailed layout of your design that includes switch boxes and physical wire connections. Routed connections are displayed as Manhattan-style lines, and unrouted connections are displayed as fly-lines.





As you move your mouse pointer slowly over the layout, the name and location of each REGION, group, component, port, net, and site are displayed as tool tips and also appear in the status bar. The tool tips and status bar also display the group name for components that are members of a group.

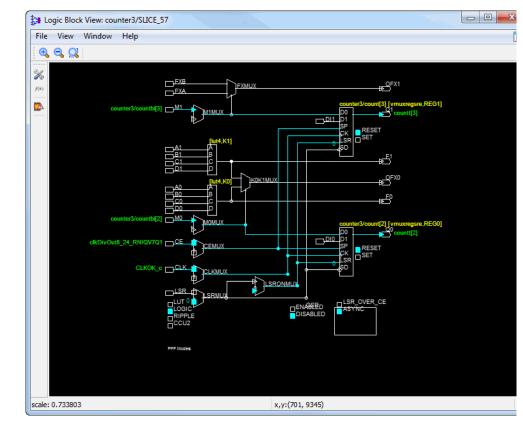
The Physical View toolbar allows you to select the types of elements that will be displayed on the layout, including components, empty sites, switch boxes, switches, pin wires, routes, and timing paths.

If your design uses the incremental design flow, Physical View will display partitions that are included in the design. For more information, refer to "Using Incremental Design Flow" in the Diamond online Help.

Physical View is available after placement and routing.

## **Logic Block View**

Logic Block View is available from either Physical View or Floorplan View, and enables you to examine logic details of one or more placed and routed components. It provides either a schematic or tabular view, depending on the type of component selected.



#### Figure 105: Logic Block Schematic View

Schematic views of PIO and PFU/PFF components can be accessed from NCD View, Floorplan View, and Physical View. Tabular views of PLL, EBR, and DSP blocks can be accessed from Floorplan View and Physical View. Right-click one or more selected components and choose **Logic Block View**.

Multiple Logic Block Views can be opened at one time, up to the maximum that has been set in the Tool > Options dialog box for Physical View.

# **Timing Analysis View**

Timing Analysis View provides a graphical way to navigate timing information and a fast timing analysis loop that allows dynamic path changes without having to re-implement or remap your design. The File List view of your project contains the Timing Analysis preference files (.tpf) in the Analysis folder. One or none of the .tpf files can be active.

- File List X Image: Image: Image: Amage: IFE2-35E-5F672C 4 퉬 Strategies 🗒 Area I/O Assistant 🗒 Quick 🗒 Timing 📰 Strategy1 Strategy2 verilog\_vhdl ▲ I verilog\_vhdl\_edif 4 퉬 Input Files edif/rev\_1/count8.edn Source/topcount.v Source/typepackage.vhd [work] Source/count16.vhd [work] Synthesis Constraint Files 4 🌗 LPF Constraint Files mixedcounter\_v.lpf mixedcounter\_vv.lpf mixedcounter\_vve.lpf 4 퉬 Debug Files R reveal1.rvl Script Files 4 퉬 Analysis Files P mixedcounter.pcf 🛅 mixedcounter.tpf Programming Files 🗄 vhdl
- Figure 106: Timing Analysis Preference File

### Launching Timing Analysis View

When Timing Analysis View is launched, it uses the active .tpf file. If there is no active .tpf file, it will read timing preferences from the logical preference file (.lpf) for its initial timing calculations. You can also double-click any active or inactive .tpf file in the File List to launch Timing Analysis View using the settings from the selected .tpf file.



s P eXP2 2-17E 6A256 It It	Path Table - "FREQUENCY NET "enc_p Source Filter: Source	Destin	z " (setup) ation filter:			Ð
eXP2 2-17E \$A256 It	Source Filter:	Destin				Ð
eXP2 2-17E \$A256 It	Source Filter:	Destin				đ
2-17E \$A256 It			ation filter:			
	Source				Case s	ensitive
		Destination	Weighted Slack	Arrival	Required	Di 1
	1 u1_encoder/busy_cnt[0]	u1_encoder/tx_data	-0.149	5.149	5	5.05e
	2 u1_encoder/busy_cnt[4]	u1_encoder/tx_data	-0.135	5.135	5	5.042
se Timing	3 u1_encoder/data_reg[2]	u1_encoder/tx_data	0.046	4.954	5	4.861
	4 u1_encoder/busy_cnt[4]	u1_encoder/tx_data	0.056	4.944	5	4.851
	5 u1_encoder/data_reg[6]	u1_encoder/tx_data	0.062	4.938	5	4.845
	•	1 1			1	•
alysis Type	Report: setup					ć
ір –	30 items scored	l, 2 timing errors	detected. ts by 0.149ns Cell/ASIC name ul encoder/busy of	<u>cnt[0]</u> (from	enc pll clk	<u>c</u> +)
	•			4 logic level	3.	ŀ
aly ip i ip i ip	sis Type	sis Type Report: setup Preference: FREQUENCY NET ' 30 items scored Error: The following path e Logical Details: Cell typ Source: FF Destination: FF Delay: 5.0	sis Type Report: setup Preference: FREQUENCY NET "enc_pll_clk_c" 20 30 items scored, 2 timing errors Error: The following path exceeds requiremen Logical Details: Cell type Pin type Source: FF Q Destination: FF Data in Delav: 5.056ns (32.3% logi	S       d_encoder/busy_cnt[4]       u1_encoder/tx_data       0.056         4       u1_encoder/busy_cnt[4]       u1_encoder/tx_data       0.056         5       u1_encoder/data_reg[6]       u1_encoder/tx_data       0.062         4       III       III         sis Type         Report: setup         Preference: FREQUENCY NET "enc_pl1_clk_c" 200.000000 MHz ; 30 items scored, 2 timing errors detected.         Error: The following path exceeds requirements by 0.149ns         Logical Details: Cell type Fin type       Cell/ASIC name         Source:       FF       Q       11_encoder/tsx_data         Delav:       5.056ns       (32.3% logic. 67.7% route).       III	S       II_encoder/data_reg[2]       ul_encoder/tx_data       0.040       4.334         4       ul_encoder/busy_cnt[4]       ul_encoder/tx_data       0.056       4.944         5       ul_encoder/data_reg[6]       ul_encoder/tx_data       0.062       4.938         4       III       III       III         sis Type         Report: setup         Preference: FREQUENCY NET "enc_pll_olk_c" 200.000000 MHz ; 30 items scored, 2 timing errors detected.         Error: The following path exceeds requirements by 0.149ns         Logical Details: Cell type Pin type       Cell/ASIC name (clock net +/ Source: FF Q ul_encoder/busy_cnt[0] (from ul_encoder/tx_data (to enc r Destination: FF Data in ul_encoder/tx_data (to enc r III)         Delay: 5.056ns (32.3% logic, 67.7% route). 4 logic level	S       In_encoder/data_reg[2]       un_encoder/tx_data       0.040       4.934       5         4       un_encoder/tx_data       0.056       4.944       5         5       un_encoder/tx_data       0.062       4.938       5         4       Intercoder/tx_data       Intercoder/tx_data       Intercoder/tx_data       5         5       Intercoder/tx_data       Intercoder/tx_data       10       10       10         6       Intercoder/tx_data       Intercoder/tx_data       10       10       10       10         6       Intercoder/tx_data       Intercoder/tx_data       10       10       10       10         10       Inte

The sections in Timing Analysis View display the trace settings from your active strategy and the preferences from the active .tpf or .lpf file. All timing preferences are listed in the Preference Reports section of the Preferences pane. If you have selected the "Check Unconstrained Paths" option in the active strategy, the Other Reports section will contain a list of suggested preferences for unconstrained paths. You can then examine the path table for each suggested preference, including the classification, Start Point, and End Point of each unconstrained path.

Timing Analysis View also provides views of the path table, detailed path tables, schematic path and report.

Settings	Values	Path Table - "FREQUENCY NET "end	_pll_clk_c" 200.000000 N	1Hz " (setup)				
Device Family Device Package	LatticeXP2 LFXP2-17E FTBGA256	Source Filter:	De	stination filter:			Case sensitiv	ve
Setup Performance Grade Hold Performance Grade	Default Default	Source	Destination	Weighted Slack	Arrival	Required	Data Delay	,
Check Unconstrained Connections	No	1 u1_encoder/busy_cnt[0]	u1_encoder/tx_data	-0.149	5.149	5	5.056	
Check Unconstrained Paths Report Asynchronous Timing Loops	Yes No	2 u1_encoder/busy_cnt[4]	u1_encoder/tx_data	-0.135	5.135	5	5.042	
Report Style	Verbose Timing	3 u1_encoder/data_reg[2]	u1_encoder/tx_data	0.046	4.954	5	4.861	
Full Name Worst-Case Paths	No 10	4 u1 encoder/busy cnt[4]	u1 encoder/tx data	0.056	4.944	5	4.851	
Number of Unconstrained Paths	0	•					•	
		Schematic Path view Setup Time: 0.093						
Preference Name	Analysis Type	Secup Time: 0.095						
	c setup c hold setup hold c setup c hold setup setup	0.383 1.277 00 01_encoder, R20C458 Fanin R19C458.D1 R19C458.B1	0.494 0.94 0.94 0FX0_0 0FX	A1 OFX0	0.20 0 oder/N_120 R21	F0 -	D10 r/bc_m CLK R21C450	C

Figure 108: Timing Analysis Schematic Path View

The Timing Analysis vertical toolbar on the left contains the following controls:

- Export exports the timing paths to a cvs file.
- Settings allows you to change settings for a timing analysis run. To change the settings permanently, you must edit the trace settings of the active strategy.
- Change Timing Preferences displays the Timing Preferences tab of the Spreadsheet view. This is a graphical view of the .tpf file.
- Fit All Columns sizes each column of the Detailed Path Table to fit the information displayed.
- View All Columns sizes all columns of the Detailed Path Table to fit inside the window.
- Path Table displays or hides the path table.
- Report: setup displays or hides the Report: Setup tab.
- Schematic Path view displays or hides the Schematic Path View tab.
- Detailed Path Tables displays or hides the Detailed Path Tables tab.
- Update when this button is visible and rotating, indicates a preference has been changed. When clicked, it recalculates timing.

Timing Analysis View enables you to cross-probe a selected timing path to view it in Floorplan View or Physical View, or to jump to the specific section of the timing report. Right-click a path, a schematic block, or highlighted text in the report, and then choose the desired command from the "Show In" pop-up menu.

# Updating and Saving Timing Preference File Settings

When you change a timing preference, you must run **Update** to recalculate the timing. Use the following steps to change timing preferences, recalculate timing and save your preferences:

- 1. In Timing Analysis View, click the Change Timing Preferences button is on the vertical toolbar. This opens Spreadsheet TPF View.
- 2. Modify a preference in Spreadsheet TPF View. An asterisk indicating a data change appears in both the Spreadsheet and Timing Analysis tabs.
- 3. Select Timing Analysis View again.

The Update button in the vertical toolbar is now visible and rotates, indicating that a timing preference has changed and that the timing has not yet been recalculated.

- 4. Click the rotating Update button to recalculate the timing.
- 5. To save the .tpf file, make sure that Timing Analysis View is active, and then choose **File > Save**.

Timing Analysis view must be active in order to use the File > Save command. You cannot save timing preferences from Spreadsheet TPF View.

### **Exporting TPF Settings**

To save your TPF settings to the logical preference file, first select the Timing Preferences tab of Spreadsheet TPF View. Right-click the preference you want to export and choose **Export to LPF**. This will export the selected setting to the active .lpf file.

Figure	109:	Export	TPF

	Start Page 🗵 🛛	🔃 Reports 🗵 🛛 🧐	Timing	j Analysis View  🗵	📽 Spreadsheet View - TPF 🛛	Б×
ø	Preference Name	Preference	e Value	Preference Unit		
<b>ૻ</b> ૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢૢ	Preference Name BLOCK FREQUENCY Frequency Hold Marg PAR_ADJ PERIOD INPUT_SETUP CLOCK_TO_OU MULTICYCLE MAXDELAY CLKSKEWDIFF	Modify Delete Export to LPF	)	MHz		
		III	Crown			4
	Global Preferences	Timing Preferences	Group			

### Importing In-Memory Timing Preferences

If you edit timing preferences in the regular Spreadsheet View after you have launched Timing Analysis View, you must import these in-memory timing preferences in order to run timing analysis on them.

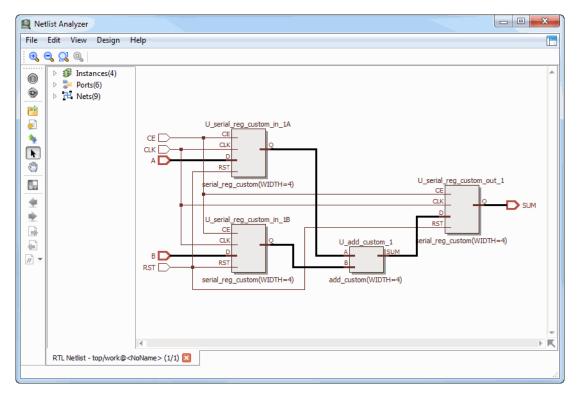
- In Timing Analysis View, click the Change Timing Preferences button on the vertical toolbar.
- 2. In the TPF Spreadsheet View, choose File > Import > Copy LPF to TPF.
- Return to the Timing Analysis View main window and click the Update button.

### **Netlist Analyzer**

Netlist Analyzer works with Lattice Synthesis Engine (LSE) to produce schematic views of your design while it is being implemented. (Synplify Pro also provides schematic views.) Use the schematic views to better understand the hierarchy of the design and how the design is being implemented..

The Netlist Analyzer window has four parts:

#### Figure 110: Netlist Analyzer



- Tool bar provides buttons for various functions.
- Browser provides nested lists of module instances, ports, and nets.
- Schematic view shows a schematic of the design. Depending on the size of the design, the schematic may be made of multiple sheets.
- Mini-map, which is a miniture view of the sheet, helps you pan and zoom in the schematic view.

Netlist Analyzer can have multiple schematics open. The open schematics are shown on tabs along the bottom of the window.

There are several ways to adjust the view of a schematic and to navigate through the hierarchy.

For detailed information about Netlist Analyzer, see *About Netlist Analyzer* in the Lattice Diamond online Help.

### **LDC Editor**

The Lattice Design Constraints (LDC) Editor is a synthesis constraint tool for use with the Lattice Synthesis Engine (LSE). Currently, the Lattice Synthesis Engine and LDC Editor support the MachXO, MachXO2, MachXO3L, and Platform Manager device families.

LSE is a synthesis tool custom-built for Lattice products and fully integrated with Diamond. Depending on the design, LSE may lead to a more compact or

faster placement of the design than another synthesis tool would do. LSE can be run from the Diamond main window or through the command line, similar to the other integrated synthesis tools.

The LDC Editor uses a spreadsheet style user interface that enables you to quickly create and edit Synopsys Design Constraints and save them to a Lattice Design Constraint file (.ldc). You can create several .ldc files and select one of them to serve as the active synthesis constraint file for the current implementation.

#### Figure 111: LDC Editor

ſ	) S	tart Page	🗙 🔃 Reports 🗵	ể attributes_ldc.ldc * 🛛	Ē
		Туре	Port	Clock	Value(ns)
	1	Input	[all_inputs]	clock	1.200000
	2	Output	[all_outputs]	clock	1.200000
	3				

After you have selected the Lattice Synthesis Engine (LSE) as the synthesis tool, the LDC Editor will open automatically each time you create or open an .ldc file. You also have the option of viewing and editing .ldc files in the Source Editor. The LDC Editor includes individual tabs for Clocks, Inputs/Outputs, and Delay Paths. Each sheet enables you to define synthesis constraints by double-clicking a cell and selecting or typing a value.

**Clocks** The Clocks tab allows you to define an alias to be associated with an existing clock port or net from the source file. Double-click the Source cell to select an existing clock port or net, and then enter an alias for the clock in the Clock Name cell. Enter a clock period in nanoseconds.

**Inputs/Outputs** The Inputs/Outputs tab enables you to specify an input or output delay relative to a clock. Double-click the Type cell to select the type of delay (input or output), and then select from the input or output ports in the Port cell. In the Clock cell, select an existing clock or an alias name that has been defined for a clock. Enter a delay value in nanoseconds.

**Delay Paths** The Delay Paths tab allows you to define a Multicycle path, specify a Max\_Delay for a timing path, and identify a False path that is to be excluded from timing analysis. Double-click the Delay Type cell to select the type of delay, and then specify the path information, delay, and cycles as appropriate.

For detailed information about setting SDC constraints, see *Applying Lattice Synthesis Engine Constraints* and the *Constraints Reference Guide* in the Lattice Diamond online Help.

### **Schematic Editor**

Schematic Editor works in conjunction with Symbol Editor and Symbol Library Manager. You design by laying out symbols for basic functions and other modules and drawing wires between their ports. You can also create your own library of custom schematic symbols.

To begin a new schematic design, choose File > New > File.

For more information, see the "Entering the Design" section of the Lattice Diamond online Help.

# **Power Calculator**

Power Calculator estimates the power dissipation for your design. It uses parameters such as voltage, temperature, process variations, air flow, heat sink, resource utilization, activity and frequency to calculate the device power consumption. It reports both static and dynamic power consumption.

Power Calculator files (.pcf) are managed in the Analysis Files folder of the File List.

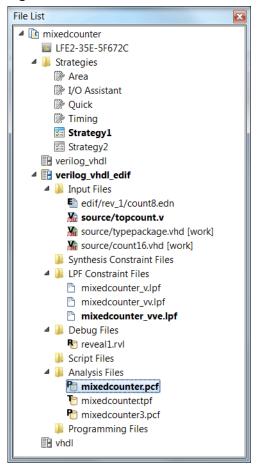


Figure 112: Power Calculator File

To launch Power Calculator from Diamond, choose **Tools > Power Calculator** or click the Power Calculator button in the toolbar.

When Power Calculator is launched, the .pcf file it uses will depend on the following conditions:

- If an active .pcf file exists, it will be used. An inactive .pcf file will be used if it is the only one available.
- If an active or inactive .pcf file in the File List Analysis Files folder is double-clicked, it will be used.
- If no .pcf file exists, Power Calculator will perform power calculations based on the current open design.

**Power Calculation Modes** Power Calculator opens in estimation mode or calculation mode, depending on the status of the selected .pcf file. If it opens in calculation mode, it will import any Bank VCCIO settings that are in the preference file and display these settings on the I/O page. When you make certain data changes in calculation mode, Power Calculator reverts to estimation mode. In Diamond 2.1 or higher, if you are using a named .pcf file and have not yet saved the changes, Power Calculator will enable you to revert to calculation mode by using the Edit > Revert to Calculation Mode command.

**Power Calculator Pages** When Power Calculator opens, it displays the Power Summary page, which enables you to change the targeted device, operating conditions, voltage, and other basic parameters. Updated estimates of power consumption are then displayed based on these changes. Tabs for other pages, including Power Matrix, Logic Block, Clocks, I/O, I/O Term, Block RAM, Graph, and Report, are arranged across the top. The number and types of these pages will depend on the target device.

**Implementation Comparisons** Power Calculator enables you to compare the power consumption among multiple implementations of your design. Each implementation must target the same device and have at least one power calculator project file (.pcf). You can access the implementation comparisons from the Edit menu.

### Figure 113: Power Summary

Power Calculat	tor - Untitle	d								X
ile <u>E</u> dit <u>W</u> ind	dow <u>H</u> elp									E
🖄 🖯 🗎 🖨 🚺										
Lattice Po	wer Ca	lculator						Software Mode	e: Calculation	
Power Summary	Power	Matrix Logi	ic Block Clock	ks I/O i	/O Term	EFB Misc	Graph Report			
Device							Environment			
Family:	MachXO2	▼ Pe	erformance grade	: 5		•		Thermal Profile		
Device:	LCMXO2-25	56НС 🔻 Ор	perating condition	s: Commercia		•	Ambient Temperature(	°C): 25		
Package type:	QFN32	▼ Pa	rt Names:	LCMXO2-25	6HC-5SG	32C 🔻	Effective Theta-JA:	17.29		
Device Power F	Parameters						Junction Temperature(	°C): 25.56		
				-1	Bou	ver Control	Maximum Safe Ambier	nt(°C): 84.37		
Process Type:	Typical	▼ Power File	e Revision: Fina	1	FUN	ver control	Haximan Sale Ambler			
Process Type:			Current by Pov			Power by Pow Static (W)	er Supply	Power by Block		1
	amic Power M	1ultiplier	Current by Pov	ver Supply	ıtal (,	Power by Pow	ver Supply Dynamic (W) :al (	Power by Block	0.000662	
Voltage/Dyna	amic Power M Voltage	1ultiplier DPM	Current by Pov Static (A)	ver Supply Dynamic (A)	ıtal (. 0.0	Power by Pow Static (W)	rer Supply Dynamic (W) :al ( 0.028315 0	Power by Block Logic Block Clocks	0.000662	
Voltage/Dyna	whice Power M Voltage 3.300	1ultiplier DPM 1.00	Current by Pov Static (A) 0.001215	ver Supply Dynamic (A) 0.008580	ıtal (, 0.0 0.0	Power by Pow Static (W) 0.004010	er Supply Dynamic (W) :al ( 0.028315 0 0.000000 0	Power by Block	0.000662	
Voltage/Dyna Vcc Vccio 3.3	Voltage 3.300 3.300	1ultiplier DPM 1.00 1.00	Current by Pov Static (A) 0.001215 0.000000	ver Supply Dynamic (A) 0.008580 0.000000	ıtal (, 0.0 0.0 0.0	Power by Pow Static (W) 0.004010 0.000000	Ver Supply           Dynamic (W)         :al (           0.028315         0           0.000000         0           0.000000         0	Power by Block Logic Block Clocks I/O	0.000662 0.000073 0.000060	
Voltage/Dyna Vcc Vccio 3.3 Vccio 2.5	amic Power M Voltage 3.300 3.300 2.500	1ultiplier DPM 1.00 1.00 1.00	Current by Pov Static (A) 0.001215 0.000000 0.000001	ver Supply Dynamic (A) 0.008580 0.000000 0.000000	ıtal (, 0.0 0.0 0.0	Power by Pow Static (W) 0.004010 0.000000 0.000002	Dynamic (W)         :al (           0.028315         0           0.000000         0           0.000000         0           0.000000         0	Power by Block Logic Block Clocks I/O EFB	0.000662 0.000073 0.000060 0.000006	
Voltage/Dyna Vcc Vccio 3.3 Vccio 2.5 Vccio 1.8	amic Power M Voltage 3.300 3.300 2.500 1.800	tultiplier DPM 1.00 1.00 1.00 1.00	Current by Pov Static (A) 0.001215 0.000000 0.000001 0.000000	ver Supply Dynamic (A) 0.008580 0.000000 0.000000 0.000000	rtal (. 0.0 0.0 0.0 0.0	Power by Pow Static (W) 0.004010 0.000000 0.000002 0.000000	Dynamic (W)         :al (           0.028315         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0	Power by Block Logic Block Clocks I/O EFB Misc	0.000662 0.000073 0.000060 0.000006 0.031526	
Voltage/Dyna Vcc Vccio 3.3 Vccio 2.5 Vccio 1.8 Vccio 1.5	amic Power M Voltage 3.300 3.300 2.500 1.800 1.500	Aultiplier DPM 1.00 1.00 1.00 1.00 1.00	Current by Pov Static (A) 0.001215 0.000000 0.000000 0.000000 0.000000	ver Supply Dynamic (A) 0.008580 0.000000 0.000000 0.000000 0.000000	ıtal (. 0.0 0.0 0.0 0.0 0.0	Power by Pow Static (W) 0.004010 0.000000 0.000000 0.000000 0.000000	Dynamic (W)         :al (           0.028315         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.0000000         0           0.000000         0	Power by Block Logic Block Clocks I/O EFB Misc Total	0.000662 0.000073 0.000060 0.000006 0.031526	]
Voltage/Dyna Vcc Vccio 3.3 Vccio 2.5 Vccio 1.8 Vccio 1.5	amic Power M Voltage 3.300 3.300 2.500 1.800 1.500	Aultiplier DPM 1.00 1.00 1.00 1.00 1.00	Current by Pow Static (A) 0.001215 0.000000 0.000000 0.000000 0.000000 0.000000	ver Supply Dynamic (A) 0.008580 0.000000 0.000000 0.000000 0.000000 0.000000	ıtal (. 0.0 0.0 0.0 0.0 0.0	Power by Pow Static (W) 0.004010 0.000000 0.000000 0.000000 0.000000	Dynamic (W)         :al (           0.028315         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.0000000         0           0.000000         0	Power by Block Logic Block Clocks I/O EFB Misc Total	0.000662 0.000073 0.000060 0.000006 0.031526	
Voltage/Dyna Vcc Vccio 3.3 Vccio 2.5 Vccio 1.8 Vccio 1.5	amic Power M Voltage 3.300 3.300 2.500 1.800 1.500	Aultiplier DPM 1.00 1.00 1.00 1.00 1.00	Current by Pow Static (A) 0.001215 0.000000 0.000000 0.000000 0.000000 0.000000	ver Supply Dynamic (A) 0.008580 0.000000 0.000000 0.000000 0.000000 0.000000	ıtal (. 0.0 0.0 0.0 0.0 0.0	Power by Pow Static (W) 0.004010 0.000000 0.000000 0.000000 0.000000	Dynamic (W)         :al (           0.028315         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.000000         0           0.0000000         0           0.000000         0	Power by Block Logic Block Clocks I/O EFB Misc Total	0.000662 0.000073 0.000060 0.000006 0.031526	

### Figure 114: Implementation Comparisons

	Implementation	PCF	Logic Block	Clocks	I/O	EFB	Misc	Total
1		C:/lscc/dia	0.002830	0.000356	0.002905	0.000023	0.031764	0.037878
2	mixed1	C:/lscc/dia	0.045292	0.001594	0.045896	0.000027	0.031774	0.124583

### Added Features for Low-Power Architecture

Three power modules are available for MachXO2, MachXO3L, and Platform Manager 2 devices: Power Controller, Dynamic Bank Controller, and Power Guard. You can generate these modules from IPexpress. Refer to the *IPexpress Module Reference Guide* for more information. Power Calculator provides tools for taking advantage of these features, including the Power Option Controller, which is accessible on the Power Summary page. It enables you to conserve power by turning on and off blocks in the chip that consume power, manage the entrances and exits of signals, and use power guard (PG) to stop signals from entering the chip.

# Figure 115: Power Option Controller for MachXO2, MachXO3L, and Platform Manager 2

Power Option Contro	ller			? <mark>- X</mark>	3
Standby Option	Dynamic B	ank Option			
Disable BandGap	Bank0	Disable InRD	Disable LVDSO	Enable PG	
	Bank1	Disable InRD		Enable PG	
Disable POR	Bank2	Disable InRD		Enable PG	
Disable OSC	Bank3	Disable InRD		Enable PG	
Disable PLL	Bank4	Disable InRD		Enable PG	
	Bank5	Disable InRD		Enable PG	
		ОК	Cancel	Help	
		UK	Cancer	нер	

First, set the standby, shut-off, and power guard options that you want to allow, using the appropriate Power Calculator pages. Afterwards, return to the Power Summary page, click the **Power Control** button to open the Power Option Controller, and select the elements to be turned off during low-power operation.

**Power Option Controller** For MachXO2, MachXO3L, and Platform Manager 2 devices, the Power Option Controller includes the following options for turning off elements that have been placed in standby mode:

- Disable Bandgap turns off the Bandgap. When this options is selected, analog circuitry such as the PLLs, on-chip oscillator, and referenced and differential I/O buffers are also turned off.
- Disable POR turns off the power-on-reset circuit, which monitors VCC levels. When the POR circuitry is turned off, limited power detection circuitry is still active. This option is only recommended for applications in which the power supply rails are reliable.
- Disable OSC turns off the on-chip oscillator.
- Disable PLL turns off the selected phase-locked loop.

The Power Option Controller includes the following dynamic bank options for MachXO2, MachXO3L, and Platform Manager 2 devices:

Disable InRd – turns off the referenced and differential input buffers for a selected bank.

- Disable LVDSO turns off the LVDS output buffer for a selected bank.
- Enable PG enables Power Guard for the selected bank.

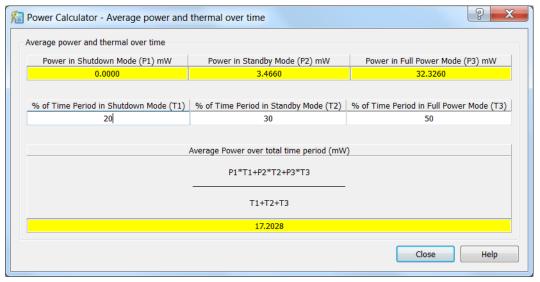
**Standby Mode Power Comparisons** For the low-power MachXO2, MachXO3L, and Platform Manager 2 devices, Power Calculator estimates the amount of power used by components that can be placed in standby mode and compares this usage with non-standby mode. The results are presented in a comparison chart that you can access from the Edit menu.

Power Calcula	tor - Compariso	on c ? X
Comparison cha	rt of power (W)	
	Standby On	Standby Off
Logic Block	0.000654	0.000662
Clocks	0.000072	0.000073
I/O	0.000058	0.000059
EFB	0.000006	0.000006
Misc	0.002676	0.031526
Total	0.003466	0.032326
	01	
	ОК	Help

#### Figure 116: Standby Mode Comparison Chart

Average Time-Based Power Usage When you target a MachXO2, MachXO3L, and Platform Manager 2 device, you will typically be using standby mode, regular mode, and shutdown mode for various periods of time. Power Calculator provides an estimate of the average power used, based on the percentage of time taken by each of these modes. As shown in Figure 117, the comparisons are presented in a table that you can access from the Edit menu.





### **Non-Integrated Power Calculator**

Power Calculator is also available as a non-integrated tool, which you can launch without opening Diamond. The non-integrated Power Calculator provides all the same functionality as the integrated version. To open the nonintegrated Power Calculator from the Windows Start menu, select **Programs** > Lattice Diamond > Accessories > Power Calculator. The Startup Wizard enables you to create a new Power Calculator project, based on a selected device or a processed design, or to open an existing Power Calculator project file (.pcf).

Lattice also provides a Power Estimator application that is available as a separate installation. Power Estimator provides the same estimation mode functionality of Power Calculator and does not require an installation of Diamond. The Power Estimator Startup Wizard allows you to estimate power based on a selected Lattice device or based on an existing Power Calculator project file.

For more information on Power Calculator see *Analyzing Power Consumption* in the Lattice Diamond online Help.

### **ECO Editor**

The Engineering Change Order (ECO) Editor enables you to safely make changes to an implemented design without having to rerun the entire process flow. Choose **Tools > ECO Editor** or click the ECO Editor button **:** on the toolbar.

	Туре	Name	Pin	Bank	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	PCICLAMP
1	Input	clk	A10	0	LVCMOS25	UP	NA	FAST	OFF
2	Input	rst	A15	1	LVCMOS25	UP	NA	FAST	OFF
3	Output	c_down[1]	E21	2	LVCMOS25	UP	12	FAST	OFF
4	Output	c_down[2]	E22	2	LVCMOS25	UP	12	FAST	OFF
5	Output	c_up_2[0]	F1	7	LVCMOS25	UP	12	FAST	OFF
6	Output	c_down[0]	F22	2	LVCMOS25	UP	12	FAST	OFF
7	Output	c_up_2[2]	K20	2	LVCMOS25	UP	12	FAST	OFF
8	Output	c_up_2[1]	L19	2	LVCMOS25	UP	12	FAST	OFF
9	Output	c_up[1]	M20	3	LVCMOS25	UP	12	FAST	OFF
10	Output	c_down_3[1]	N19	3	LVCMOS25	UP	12	FAST	OFF
11	Output	c_down_3[2]	N20	3	LVCMOS25	UP	12	FAST	OFF
12	Output	c_up[2]	N21	3	LVCMOS25	UP	12	FAST	OFF
13	Output	c_up[0]	P21	3	LVCMOS25	UP	12	FAST	OFF
14	Output	c_down_3[0]	P22	3	LVCMOS25	UP	12	FAST	OFF
15	Output	c_down_2[0]	P4	6	LVCMOS25	UP	12	FAST	OFF
16	Output	c_down_2[2]	R21	3	LVCMOS25	UP	12	FAST	OFF

#### Figure 118: ECO Editor

ECOs are requests for small changes to be made to your design after it has been placed and routed. The changes are directly written into the native circuit description database file (.ncd) without requiring that you go through the entire design implementation process.

ECOs are usually intended to correct errors found in the hardware model during debugging. They are also used to facilitate changes that had to be made to the design specification because of problems encountered when other FPGAs or components of the PC board design were integrated.

The ECO Editor includes windows for editing I/O settings, PLL settings, and memory initialization values. It also provides a Change Log window for you to track changes between the modified .ncd file and the post-PAR .ncd file.

#### Note

After you edit your post-PAR, routed .ncd file, your functional simulation and timing simulation will no longer match.

For more information, see *Applying Engineering Change Orders* in the Lattice Diamond online documentation.

### Programmer

After you have placed and routed the design and generated the JEDEC or bitstream file, you can use Diamond's integrated Programmer to program the target device. Choose **Tools > Programmer**. Programmer detects the cable type, scans the device chain, creates the XCF file, and downloads the data file to the device. If you wish to program, for example, a dual boot or a merged hex file into an SPI Flash device, you must first use the Deployment Tool to generate the hex file. See "Deploying the Design with the Deployment Tool" in the Lattice Diamond online Help for information about converting file types with Deployment Tool.

#### Figure 119: Programmer

🤮 Dia	mond Prog	rammer - LEDtest.xcf					
File	Edit Viev	v Design Help					
1	📬 🗐	8 8 8 8 8 8	103				
En	able Status	Device Family	Device	Operation	File Name		Cable Settings
1 🔽		LatticeECP3	LFE3-35EA	Fast Program	amond_tutorial/LEDtest/LEDtest_LEDtest.bit		Detect Cable
							Cable: HW-USBN-2B (FTDI)
							Port: FTUSB-0
							Custom port (HEX):
							I/O Settings  O Use default I/O settings
						2	Use custom I/O settings
						Cable and I/O Settings	
						10 S	INITN pin connected
						and	DONE pin connected
						Cable	
							<ul> <li>Set TRST high</li> </ul>
							Set TRST low
							PROGRAMN pin connected ispEN pin connected
							Ø Set ispEN high
							<ul> <li>Set ispEN low</li> </ul>
- ا			m			- F	

Programmer supports serial, concurrent (turbo), and microprocessor programming of Lattice devices in PC and Linux environments. Device chains can be scanned automatically using the Programmer graphical user interface.

Programmer is integrated into the Diamond software environment, and is also available in a standalone version.

Features include:

- Scan chain and display chain contents (.xcf file)
- Download data files to devices
- Create/modify/display .xcf file
- Generate files based on the .xcf file (Tcl/command line only)

Programmer uses a Single Document Interface (SDI) where a single .xcf project is displayed per Programmer instance. Opening additional .xcf files in Diamond-integrated mode will close the current .xcf and open the specified .xcf.

The main view displays the devices in the current Programmer project resulting from the Scan action, or from manual creation in a table.

Double-clicking on an uneditable cell or right-clicking and selecting Device Properties opens a dialog that displays more information about the selected device. Additionally, some entries may be edited directly on the table by clicking.

Columns can be displayed or hidden by choosing View > Columns. The default columns that are displayed are Process, Status, Device Family, Device, Operation, File Name, File Date/Time, Checksum, USERCODE, and Verbose Logging.

Additional columns are available, but are hidden by default. The columns are Device Vendor, Device Full Name, and Device Description.

The following columns are directly editable from the table:

- Enable
- Device Family
- Device
- File Name
- Verbose Logging

Some of these columns become un-editable (grayed out) if the selected operation or other option does not support it. For example, File Name will be grayed out for a 'Bypass' operation.

Each row has a column enabling the device. Devices where the enable option is not selected will not be programmed. They will effectively be treated as a bypass operation. This allows one .xcf file to be used whether programming all devices in a scan chain or just a single device.

Each row has a column for the device status. The status indicates whether the operation performed was successful or not. This field is also used for read back operations to display what is read back if the data to display is short. For larger data sets that are read back, a dialog box is displayed.

Programmer can also be used as a stand-alone tool. From the Windows Start menu, choose **Programs > Lattice Diamond > Accessories > Diamond Programmer**.

For more information about programming a device with Programmer, see the "Programming the FPGA" section of the Lattice Diamond online Help.

### **Soft Error Injection**

Ambient radiation occasionally collides with a circuit component in a way that alters its logical state. This effect is strongest at high altitudes, where the Earth's protective electromagnetic field is weaker. SEI (Soft Error Injection) Editor allows you to generate single-bit errors, insert them into a bitstream, and detect them for analysis, simulating the effect of radiation damage on the device's configuration memory. SEI Editor currently supports the following devices:

- ECP5U
- ECP5UM

SEI Editor is available after you have placed and routed your design and generated a bitstream.

### **Running SEI Editor**

You can run SEI Editor after you have placed and routed your design and generated a bitstream.

#### To run SEI Editor:

Choose Tools > SEI Editor

SEI Editor appears.

😽 S	EI Ed	itor	19	£.					
*		~		1	1	1	1		
*		Enable	ID	SEI File Name	Method	Block	Site	Frame	Bit In Frame
-	1		0	simple_adder_sei_0	Unused	PFU			

SEI can manage multiple errors with a variety of options. The SEI Editor toolbar has three buttons:

- Run 4 Run all selected error insertions.
- Add i Add a new error for insertion.
- Remove i Remove the selected error insertion.

The SEI Editor has the following columns:

- Enable Select this option to insert the error described on the current row.
- **ID** The error's ID number. This value is read-only.
- SEI File name The name of the file containing the inserted error. You can modify this filename by double-clicking it and entering a new name.
- Method This option determines whether the error is inserted into an unused or random block. Selecting Unused inserts the error into an unused block, which will not affect the design logic. Selecting Random inserts the error into a random block, which may or may not affect the

design logic. You can change an error's Method by double-clicking it and selecting a new value from the drop-down menu.

- Block The block where the error was inserted, in one of the following categories:
  - PFU
  - EBR
  - DSP
  - ANY
  - Routing
  - Unclassified

You can direct errors into unused PFU, EBR, or DSP blocks, or into a randomly-chosen unused block. You can change the destination Block by double-clicking it and selecting a new value from the drop-down menu.

- Site The exact Site of the inserted error. Errors inserted into a Routing block or Unclassified block do not have a Site. This field is read-only.
- **Frame** The data frame of the inserted error. This field is read-only.
- Bit In Frame The bit of the inserted error within the displayed data frame.

For more information about using SEI Editor, see the "Analyzing your Design using Soft Error Injection" section in the Lattice Diamond online Help.

### **Deployment Tool**

Deployment Tool is a stand-alone tool available from the Diamond Accessories. The Deployment Tool enables you to convert data files to other formats and use the data files to generate other data file formats. A four-step wizard helps you create a new deployment and select the deployment type, input file type, and output file type.

For more information about using the Deployment Tool, see the "Deploying the Design with the Deployment Tool" in the Lattice Diamond online Help.

### Model 300 Programmer

The Model 300 Programmer is a simple engineering device programmer that allows you to perform single-device programming directly from a PC or Linux environment. The Model 300 Programmer hardware and software support all JTAG devices produced by Lattice, with device Vcc of 1.8, 2.5, 3.3, and 5.0V. The Model 300 Programmer software controls the programming process. It is available as a stand-alone tool from the Lattice Diamond Accessories menu.

For more information about using the Using the Model 300 Programmer software, see the "Using the Model 300 Programmer" section in the Lattice Diamond online Help.

le	Edit Help			
2	📔 🗔 🔤 🔤 🔤 🛃			
de	nal Memory: Hex Conversion			
Ste	p 1 of 4: Select Input File(s)			
	File Name	Device Family	Device	
1	C:/my_diamond_tutorial/LEDtest/LEDtest_LEDtest.bit	LatticeECP3	LFE3-35EA	

### Figure 120: Deployment Tool

### Figure 121: Model 300 Programmer

S Mod	So Model 300 Programmer - new model300.job										
File E	3										
1		🔥 🖸 🛄									
Statu	Device Family	Device	Device Package	Model 300 Adapter	Device Full Name	Operation	File Name	Hardware Settings		5	×
1	LatticeXP2	LFXP2-17E	484-ball fpBGA	PN-F484/LFXP2	P2-17E-XXF484	FLASH Erase, Program, Verify	p/xp2_top_xp2_to	Cable Settings			51
								Auto-Detect	Statu	s 🍳	
								Cable:	HW-USBN-2A	•	
								Port:	EzUSB-0		
									2030-0		
								Custom port (HEX):			
								Voltage Selected			
								Pwr(RDY) Vcc(ON			
								<b>\$</b>	6 6	<b>6</b> 6	
•					1		*				
Output										8	×
Lattice V Program	M Drivers detecter mer device databa	d (HW-DLN-3C ase loaded	(Parallel), HW-USBN-	2A)							
	etected HW-USBN to Detection Activ		ort EzUSB-0								=
	No HW-DUN-3C (parallel) cable detected.										
	e HW-USBN-2B (F										
INFO - M	odel300 file opene	ed successfully.									-

## **Programming File Utility**

The Programming File Utility is a stand-alone tool that allows you to view, compare, and edit data files. When comparing two data files, the software generates an output (.out) file with the differences highlighted in red. The Programming File Utility is a stand-alone tool that is available from the Lattice Diamond Accessories menu.

#### Figure 122: Programming File Utility

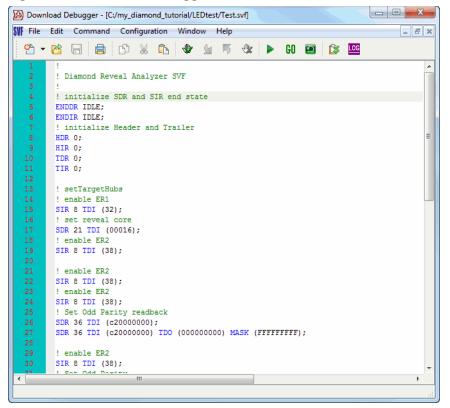
Of File Edit Command Tools Configuration Window Help	_ <i>E</i> >
NOTE DEVICE NAME: LFXP2-17E-7FPBGA484*	
NOTE PIN ASSIGNMENTS*	
NOTE PINS rfi : N21 : out*	1
NOTE PINS rstn : AB20 : in*	
NOTE PINS dout[37] : U15 : out*	
NOTE PINS dout[36] : Y14 : out*	
NOTE PINS dout[35] : AA13 : out*	
NOTE PINS dout[34] : N16 : out*	
NOTE PINS dout[33] : AA14 : out*	
NOTE PINS dout[32] : M22 : out*	
NOTE PINS dout[31] : P22 : out*	
NOTE PINS dout[30] : R21 : out*	
NOTE PINS dout[29] : N18 : out*	
NOTE PINS dout[28] : P21 : out*	
NOTE PINS dout[27] : T14 : out*	
NOTE PINS dout[26] : T20 : out*	
NOTE PINS dout[25] : T15 : out*	
NOTE PINS dout[24] : U12 : out*	
NOTE PINS dout[23] : AA16 : out*	
NOTE PINS dout[22] : R17 : out*	
NOTE PINS dout[21] : AB19 : out*	
NOTE DIVID J 1001 . AAAE	

For more information about using the Using the Programming File Utility, see the "Using Programming File Utility" section in the Lattice Diamond online Help.

### **Download Debugger**

Download Debugger is a stand-alone software tool for debugging Serial Vector Format (SVF) files, Standard Test And Programming Language (STAPL) files, and Lattice Embedded (VME) files. Download Debugger allows you to program a device, and edit, debug, and trace the process of SVF, STAPL, and VME files.

Figure 123: Download Debugger



For more information about using Download Debugger, see the "Debugging SVF, STAPL, and VME Files" in the Lattice Diamond online Help.

### **Partition Manager**

Partition Manager is the Diamond graphical user interface (GUI) used to perform such tasks such as preservation data control, re-implementation effort control, region assignment, and acts as the central interface between the user and partition database. Partition Manager supports partitioned designs for both LatticeECP2M and LatticeECP3 devices

Partition Manager is only available if Incremental Flow is enabled for an implementation in the project and can be run when the flow is at the pre-Map, post-Map, or post-PAR stage.

Partition Manager can be used after the post-Map stage to create new partitions, edit partition information or remove existing partitions. New partitions can be created for hierarchical modules. You can edit existing partition information such as partition's preservation data level, reimplementation effort, anchor and bounding box. Existing partitions can be deleted and the flow rerun without the deleted partitions.

When a partition's information has been updated in either Partition Manager or Floorplan View, the Diamond process is reset to the Translate process.

The GUI for the Partition Manager is in the form of a table. Each row corresponds to a partition in the project. The columns for each row are for the partition name, partition source, the preservation data, the reimplementation effort, the partition anchor and bounding box. All attributes except Partition Name and Partition Source may be edited.

Default values are shown in blue. Anchor and bounding box cells are blank if they are not set for the partition. The top partition module in the project is always in the first row. A color selection box allows you to change the color of the partition in as it appears in Diamond Floorplan View.

Figure 124: Partition Manager

			Partition Name	Partition Source	Preservation Data	Paimplementatio	Partition Anchor & BBox	Color	
4 🕂 iobasic			Partition Name	Partition Source	Preservation Data	Reimpiementatio	Partition Anchor & BBOX	Color	
🔺 🗾 Instances: 117	=	1	CGROUP_TOP_MODULE_iobasic	SYNTHESIS	ROUTED	UNGUIDED			
GSR_INST		2	CGROUP_compa_inst	TRANSLATE	PLACED	GUIDED			
I vcc		2	CGROOP_compa_mst	TRANSLATE	FLAGED	GOIDED			
💷 clki_pad									
🕃 clkr_pad									
clkt_pad									
Image: Compa_inst(CGROUP_compa_inst)									
GND_0									
VCC_0									
compare2									
💷 equali									
equali_3									
r2data[0]									
<pre>r2data[1]</pre>									
<pre>r2data[2]</pre>									
r2data[3]									
III r2data[4]									
r2data[5]									
120000[0]									
I r2data[6]						Remo	ove Partition Discard all Cha	nges Apply	

For more information about Partition Manager and incremental design, refer to "Using Incremental Design Flow" in the Diamond online Help.

### **Run Manager**

Run Manager runs the processes for the different implementation/strategy combinations. Choose **Tools > Run Manager** or click the Run Manager button in the toolbar.

Run Manager takes the design through the entire process flow for each selected implementation. If you are running on a multi-core system, Run Manager will distribute the iterations so that they are executed in parallel. The options "Maximum number of implementation processes in run manager" and "Maximum number of multi-par processes in run manager" are available in the Environment > General section of the Tool Options dialog box. Choose

lle	Window												l
3	0 implementations are hidde	n. <u>Click b</u>	o configure visible	implementations									
3	Implementation <st< th=""><th>rategy&gt;</th><th>Current Step</th><th>Status</th><th>Next Step</th><th>Worst Slack</th><th>Timing Score</th><th>Worst Slack (Hold)</th><th>Timing Score (Hold)</th><th>Slice</th><th>Unrouted Net</th><th>Number Of Signals</th><th>Number Of Connection</th></st<>	rategy>	Current Step	Status	Next Step	Worst Slack	Timing Score	Worst Slack (Hold)	Timing Score (Hold)	Slice	Unrouted Net	Number Of Signals	Number Of Connection
	Image: mixed < Strategy1		N/A	Completed	N/A	5.366	0	0.169	0	14/7584	0	N/A	N/A
3	N/A		N/A	N/A	N/A	5.366	0	0.169	0	14/7584	0	N/A	N/A
8	N/A		N/A	N/A	N/A	5.552	0	0.141	0	14/7584	0	N/A	N/A
•	N/A		N/A	N/A	N/A	5.486	0	0.169	0	14/7584	0	N/A	N/A
	N/A		N/A	N/A	N/A	5.392	0	0.154	0	14/7584	0	N/A	N/A
	N/A		N/A	N/A	N/A	5.381	0	0.154	0	14/7584	0		N/A
	Image: A strategy and a strategy astrategy	2>	N/A	Completed	N/A	2.659	0	0.181	0	14/7584	0		N/A
	N/A	Se	lect All		VA	2.659	0	0.181	0	14/7584	0		N/A
	N/A	Un	select All		/A	2.849	0	0.154	0	14/7584			N/A
	N/A	Ru			VA	2.697	0	0.167	0	14/7584			N/A
	N/A	_			VA.	2.819	0	0.159	0	14/7584	0		N/A
	N/A	🙆 Re			VA	2.726	0	0.153	0	14/7584	0	N/A	N/A
		00 Pai	ise										
-		Sto	p										
		🔄 Sh	ow/Hide Imple	mentation									
		_	port										
		_	t As Active		_								

#### Figure 125: Run Manager

**Tools > Options** to access it. These options enable you to set the maximum number of processes to run in parallel. Generally, the maximum number of processes should be the same as the number of cores in your processor; but if the strategy is using the "Multi-Tasking Node List" option for Place & Route Design, this number should be set to one.

You can use the Run Manager list to set an implementation as active. Rightclick the implementation/strategy pair and choose **Set as Active**.

For an implementation that uses multiple iterations of place-and-route, you can select the iteration that you want to use as the active netlist for further processes. Expand the implementation list, right-click the desired iteration, and choose **Set as Active**. The active iteration is displayed in italics.

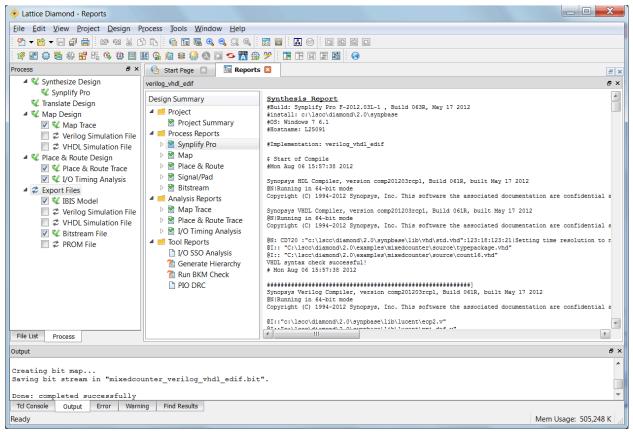
To examine the reports from each process, set an implementation as active, and then select the Reports View.

See the "Managing Projects" section of the Lattice Diamond online Help for more information about using implementations, strategies, and Run Manager.

## **Synplify Pro for Lattice**

Synplify Pro for Lattice is an OEM synthesis tool used in the Lattice Diamond design flow. Synplify Pro runs in batch mode when you run the Synthesize Design step in Process View. To examine the output report, select **Synplify Pro** in the Process Reports folder of Reports View.

#### Figure 126: Synthesis Report



You can also run Synplify Pro in interactive mode. Choose **Tools > Synplify Pro for Lattice** or click the Synplify Pro button **C** on the toolbar.

For more information, see the *Synplify Pro User Guide*, which is available from the Lattice Diamond Start Page or the Synplify Pro Help menu.

### **Active-HDL Lattice Edition**

The Active-HDL Lattice Edition tool is an OEM simulation tool that is closely linked to the Lattice Diamond environment. It is not run as part of the Process implementation flow. To run Active-HDL, choose **Tools > Active-HDL Lattice Edition** or click the Active-HDL button **A** on the toolbar.

See "Simulation Flow" on page 67 for more information about simulating your design. See "Simulation Wizard" on page 123 for information about creating a simulation project to run in Active-HDL.

For complete information about Active-HDL, see the *Active-HDL Online Documentation*, which is available from the Lattice Diamond Start Page or the Active-HDL Help menu.

### **Simulation Wizard**

The Simulation Wizard enables you to create a simulation project for your design. To open Simulation Wizard, choose **Tools > Simulation Wizard** or click the Simulation Wizard button in on the toolbar. The wizard leads you through a series of steps that include selecting a simulation project name and location, specifying the simulator to use (if you have more than one installed), selecting the process stage to use (from RTL to Post-Route Gate-Level + Timing), specifying the language (VHDL or Verilog), and selecting the source files. You can optionally run the simulation directly from the wizard.

#### Figure 127: Simulation Wizard

Simulator Project Name Enter a name for your simulator project and specify a directory where the project file will be stored.				
Project name:	mixed_sim			
Project location:	C:/Projects/Design examples/mixed_mode			
Simulator				
Active-HDL				
O ModelSim				

### **Common Tasks**

Lattice Diamond gathers the many FPGA implementation tools into one central design environment. This gives you common controls for active tools, and it provides shared data between views.

### **Controlling Tool Views**

Tool views are highly configurable in the Lattice Diamond environment. You detach a tool view to work with it as a separate window, and you can create tab groups to display two views side-by-side.

### **Detaching and Attaching a Tool View**

Each Diamond integrated tool view contains a Detach button in the upperright corner that allows you to work with the tool view as a separate window.

After a tool view is detached, the Detach button changes to an Attach button —, which reintegrates the view into the Lattice Diamond main window.

You can detach as many tool views as desired. The Window menu keeps track of all open tool views and allows you to reintegrate one or all of them with the main window or detach one of them. Those that are already integrated are displayed with a check mark.

Figure 128: Window Integrate Tools Menu

Wir	ndow Help			
	Split Tab Group			
	Merge Tab Group			
	Move to Another Tab Group			
<b>*</b>	Switch Tab Group position			
	Load Default Layout			
	Load Layout	×		
	Save Layout			
	Manage Layout			
¥*	Integrate All Tools			
	Integrate Tools	۲	✓	Reports
$\checkmark$	Reports		✓	Start Page
	Start Page		<	Spreadsheet View
	Spreadsheet View			

You can also use the Integrate All Tools button 📷 on the toolbar to dock all detached views back into the main window.

### **Tab Grouping**

Lattice Diamond allows you to split one or more active tools into a separate tab group. You can use the Window menu or the toolbar buttons to create the tab group and control the display.

#### Figure 129: Tool Tab Controls

	E			鹊
--	---	--	--	---

The Split Tab Group button separates the currently active tool into a separate tab group. Having two separate tab groups enables you to work with two tool views side-by-side. This is especially useful for dragging and dropping to make preference assignments; for example, dragging a port from Netlist View to Package View to assign it to a pin or dragging nets to the Route Priority preference sheet to prioritize them.

Having two separate tab groups is also useful for examining the same data element in two different views, such as the Floorplan and Physical View layouts.

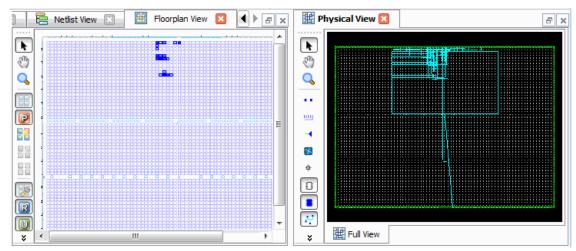
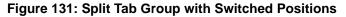
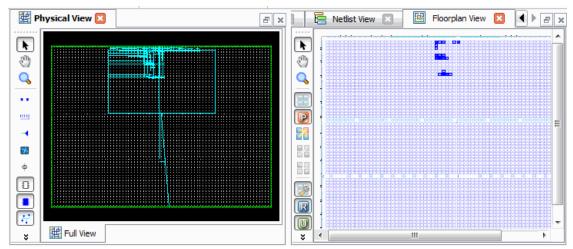


Figure 130: Split Tab Group with Side-by-Side Layout Views

You can move an active tool view from one tab group to another by dragging and dropping it, or you can use the Move to Another Tab Group button in the toolbar.

To switch the positions of the two tab groups, click the Switch Tab Group Position button an on the toolbar.





To merge the split tab group back into the main group, click the Merge Tab Group button an the toolbar.

# **Using Zoom Controls**

Lattice Diamond includes display zoom controls in the View toolbar. There are controls for increasing or reducing the scale of the view, fitting the display contents to the window view area, and fitting a selected area or object to the window view area.

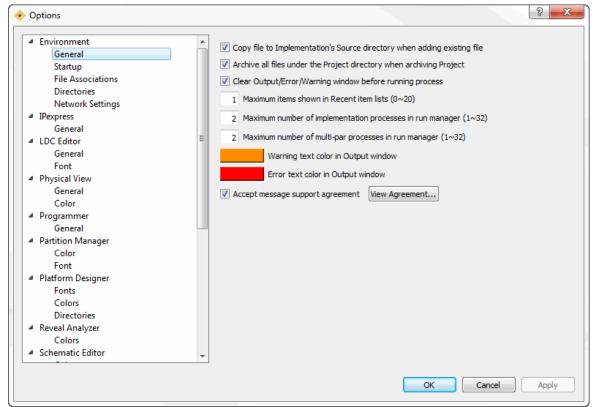
# **Displaying Tool Tips**

When you place the cursor over a graphical element in a tool view, a tool tip appears with information on the element. The same information displayed in the tool tip will also be displayed temporarily in the status bar on the lower left of the main window.

## **Setting Display Options**

The Options dialog box, which is available from the Tools menu, enables you to specify general environment options as well as customize the display for the different tools. Tool options include selections for color, font and other graphic elements.

#### Figure 132: Tool Options



# Chapter 8



# **Tcl Scripting**

This chapter describes the Tcl scripting capabilities in Lattice Diamond. The internal TCL Console and external TCL Console are described as well as some of the extended Tcl commands for Lattice Diamond control.

### Overview

Tool Command Language (Tcl) is a scripting language used for controlling software tools and automating tool control and testing. It is very useful for controlling batch operation of processes. The Lattice Diamond design environment includes an interactive Tcl Console and extended Lattice Diamond Tcl commands.

# **Tcl Console**

You can view the integrated Tcl Console by selecting its tab at the bottom of the Lattice Diamond main window. You can enter "help" at the Tcl prompt to see the major groups of Tcl commands for Lattice Diamond.

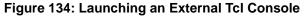
#### Figure 133: Tcl Console and Command Groups

Tcl Console	X
<pre>&gt; prj_project open "C:/lscc/diamond/examples/mixedcounter/mixedcounter.ldf" C:/lscc/diamond/examples/mixedcounter/mixedcounter.ldf &gt; help For more information on a specific command, type "help <command/>":     dtc Lattice Diamond Tcl Console extended Tcl commands.     prj Project Manager extended Tcl commands.     ncd NCD extended Tcl commands.     hle HDL Explorer extended Tcl commands.     rvl Reveal Inserter extended Tcl commands.     rva Reveal Inalyzer extended Tcl commands.</pre>	
pwc Power Calculator extended Tcl commands. pgr Programmer extended Tcl commands.	Ŧ

The integrated Tcl Console can be opened, closed, detached and attached (using the double-click method).

### **External Tcl Console**

All of the Lattice Diamond Tcl commands that are available in the integrated Tcl Console can also be used in the external console that is included with the Lattice Diamond installation. In the folder where you launched Lattice Diamond is an **Accessories** folder that contains the external Tcl Console.





Select this Tcl Console to run the external Tcl shell. You can enter "help" at the prompt to see the major groups of Tcl commands for Lattice Diamond. Help on individual commands is available both in the online Help and from the "help" command within the console. The console's help command can be used to get the top level help, help for a particular dictionary, or help for an individual command.



TCL Console	
% help	
For more information on a specific command, type "help <command/> ":	
prj Project Manager extended Tcl commands.	
ncd NCD extended Tcl commands.	=
ngd NGD extended Tcl commands.	
hle HDL Explorer extended Tcl commands.	
rul Reveal Inserter extended Tcl commands.	
sbp System builder planner extended Tcl commands.	
rua Reveal Analyzer extended Tcl commands.	
pwc Power Calculator extended Tcl commands.	
pgr Programmer extended Tcl commands.	
icf Incremental Design Flow Database extended Tcl commands.	
cmp Compile Lattice FPGA simulation libraries.	
%	
	-

To use an existing Tcl script file use "source <script\_file.tcl>". To save commands in the Tcl Console to a script file use "save\_script <script\_file.tcl>".

See "Tcl Scripts" on page 139 for more information on writing Tcl scripts.

**Running a Tcl Script When Launching Diamond** You can use a command line shell to launch the Diamond software and automatically run a Tcl script. Your Tcl script can be standard Tcl commands as well as Diamond-specific Tcl commands.

#### To launch Diamond and run a Tcl script:

In the Tcl console or other command line shell, enter the following command:

On Windows:
pnmain.exe -t<tcl\_path\_file>
On Linux:
diamond -t<tcl\_path\_file>

**Sample Diamond Tcl Script** The following simple Tcl script, running in Windows, opens a project and runs the design flow through the MAP process.

prj\_project open C:/test/iobasic\_diamond/io1.ldf
prj\_run Map -impl io1

The above example is saved in Windows as the file mytcl.tcl in the directory C:/test. By running the following command in a DOS shell or the Tcl console in Windows, the Diamond GUI starts, the project i01.ldf opens, and the MAP process automatically runs.

pnmain.exe -tc:/test/mytcl.tcl

### Commands

Every function available in the user interface is available to you as part of the extended Tcl commands for Lattice Diamond. You can create scripts to run design flow processes and manage project data as well as perform all standard Tcl operations.

The help command (help) is very useful for getting listings of available commands and their syntax.

> help

For more information on a specific command, type "help <command>":

dtc Lattice Diamond Tcl Console extended Tcl commands
prj Project Manager extended Tcl commands
ncd NCD extended Tcl commands
ngd NGD extended Tcl commands
hle HDL Explorer extended Tcl commands
rvl Reveal Inserter extended Tcl commands
rva Reveal Analyzer extended Tcl commands

pwc Power Calculator extended Tcl commands
pgr Programmer extended Tcl commands
icf Incremental Design Flow Database extended Tcl commands
cmp Compile Lattice FPGA simulation libraries

The following sections show the help command and output for each major grouping of the Lattice Diamond extended Tcl commands. You can use the help command (*help*) for more information on each specific group or command. See "Tcl Scripts" on page 139 for more information on writing Tcl scripts.

### Lattice Diamond Tcl Console

```
> help dtc
```

#### Lattice Diamond Tcl Console extended Tcl commands

history:	Shows the commands history
reset:	Reset History and clear up console
clear:	Clear up console
save_script:	Saves a script of executed commands
	Usage: save_script <script_name></script_name>
set_prompt:	Set a new prompt
	Usage: set_prompt <newprompt></newprompt>

### **Project Manager**

> help prj

Project Manager extended Tcl commands For more information on a specific command, type hlp command-name:

```
prj_project Project commands to manipulate project
 prj src
             Project source commands to manipulate project
sources
 prj_impl
             Project implementation commands to manipulate
implementation
 prj_strgy
             Project strategy commands to manipulate
strategies
             Project flow running command to run a flow
 prj_run
process
             Project synthesis tool commands to list or set
 prj_syn
synthesis tool
 prj_dev Project device commands to list or set the
device used in the project
```

### NCD

> help ncd

NCD extended Tcl commands

For more information on a specific command, type the command without any options:

ncd_port	NCD port command	
ncd_inst	NCD instance command	
ncd_net	NCD net command	
ncd attr	NCD attribute command	

### NGD

> hlp ngd

NGD extended Tcl commands

For more information on a specific command, type the command without any options:

ngd_port	NGD port command
ngd_inst	NGD instance command
ngd_net	NGD net command
ngd_attr	NGD attribute command

### **Reveal Inserter**

> help rvl

Reveal Inserter extended Tcl commands For more information on a specific command, type hlp command-name:

rvl\_project RVL project commands to manipulate reveal insert project rvl\_core RVL core commands to manipulate cores in current project rvl\_trace RVL trace commands to manipulate trace signals and optins for a debug core in current project rvl\_tu RVL trigger unit commands to manipulate trigger units for a debug core in current project rvl\_te RVL trigger expression commands to manipulate trigger expressions for a debug core in current project rvl\_tokenmgr RVL token manager commands to manipulate tokens in current project

### **Reveal Analyzer**

> help rva

Reveal Analyzer extended Tcl commands For more information on a specific command, type hlp command-name:

\_

rva_trace	Reveal	Analyzer	trace commands
rva_core	Reveal	Analyzer	core commands
rva_tu	Reveal	Analyzer	tu commands

\_

rva_te	Reveal	Analyzer	te comma	ands
rva_trigoptn	Reveal	Analyzer	trigger	options
rva_project	Reveal	Analyzer	project	commands

### **Power Calculator**

> help pwc

#### Power Calculator extended Tcl commands

For more information on a specific command, type hlp command-name:

pwc_command	Power	Calculator	command commands
pwc_device	Power	Calculator	device command
pwc_parameters	Power	Calculator	parameters command
pwc_thermal	Power	Calculator	thermal command
pwc_settings	Power	Calculator	settings command
pwc_supply			supply command
pwc_logicblocks	Power	Calculator	logicblocks command
pwc_clocks	Power	Calculator	clocks command
pwc_inout	Power	Calculator	inout command
pwc_blockram	Power	Calculator	blockram command
pwc_dspblock	Power	Calculator	dspblock command
pwc_plldll	Power	Calculator	plldll command
pwc_maco	Power	Calculator	maco command
pwc_serdes	Power	Calculator	serdes command
pwc_writereport	Power	Calculator	writereport command
pwc_efb	Power	Calculator	efb command
pwc_misc	Power	Calculator	misc command
pwc_power	Power	Calculator	power control command

### Programmer

> help pgr

Programmer extended Tcl commands

For more information on a specific command, type help command-name:

pgr_project	Programmer	project	commands
pgr_program	Programmer	program	commands

### **Incremental Design Flow**

> help icf

Incremental Design Flow Database extended Tcl commands For more information on a specific command, type hlp command-name:

icf\_data Commands to load or save the partition info for Incremental Design

icf\_part Set options on partition

# Compile Lattice FPGA Simulation Libraries

> help cmp

Compile Lattice FPGA simulation libraries. For more information, type the following command without any options:

```
> cmpl_libs
```

### **Tcl Scripting From Command-Line Shells**

Lattice Diamond Tcl Scripts can also be run directly from a command line shell such as DOS or bash. For this to be accomplished, certain environment variables need to be set up and a Tcl script created.

### Lattice Diamond Example Tcl Script

The following example shows a Tcl script created for implementing an existing project.

prj\_project open "C:/example\_path\_name/file\_name.ldf"
prj\_run Synthesis -impl cardgameFPGA
prj\_run Translate -impl cardgameFPGA
prj\_run Map -impl cardgameFPGA
prj\_run PAR -impl cardgameFPGA
prj\_run PAR -impl cardgameFPGA -task PARTrace
prj\_run Export -impl cardgameFPGA -task Bitgen
prj\_project close

### DOS Script for Running Lattice Diamond Tcl Script

To run the Lattice Diamond Example Tcl script in a Windows DOS shell, the commands from the following DOS script need to be entered.

```
set LSC_INI_PATH=
```

```
set LSC_DIAMOND=true
set TCL_LIBRARY=C:\lscc\diamond\3.1\tcltk\lib\tcl8.5
```

```
set FOUNDRY=C:\lscc\diamond\3.1\ispFPGA
```

```
set PATH=%FOUNDRY%\bin\nt;%PATH%
```

C:\lscc\diamond\3.1\bin\nt\pnmainc.exe project.tcl > output.txt

The DOS script assumes that the default directory installation is used for the Lattice Diamond software. If the software is installed into a different directory, the script will need to be modified.

### Bash Script for Running Lattice Diamond Tcl Script on Windows

Similar to the DOS script, the following example shows how to run a Lattice Diamond Tcl in a Cygwin bash shell on Windows.

```
export TEMP=/cygdrive/c/TEMP
export LSC_INI_PATH=""
export LSC_DIAMOND=true
export TCL_LIBRARY=/cygdrive/c/lscc/diamond/3.1/tcltk/lib/
tcl8.5
export FOUNDRY=/cygdrive/c/lscc/diamond/3.1/ispFPGA
export PATH=$FOUNDRY/bin/nt:$PATH
/cygdrive/c/lscc/diamond/3.1/bin/nt/pnmainc.exe project.tcl >
output1.txt
```

Like the DOS script, the bash script assumes that the default directory installation is used for the Lattice Diamond software. If the software is installed into a different directory, the script will need to be modified.

# Bash Script for Running Lattice Diamond Tcl Script on Linux

Lattice Diamond software is supported on Linux also. The following is an example for running the Lattice Diamond Tcl script example in a Linux bash shell.

```
export TEMP=/tmp
export LSC_INI_PATH=""
export LSC_DIAMOND=true
export TCL_LIBRARY=/usr/local/lscc/diamond/3.1/tcltk/lib/tcl8.5
export FOUNDRY=/usr/local/lscc/diamond/3.1/ispFPGA
export PATH=$FOUNDRY/bin/nt:$PATH
/usr/local/lscc/diamond/3.1/bin/lin/diamondc project.tcl >
output1.txt
```

As in the previous examples, the Linux example assumes that the default directory installation is used for the Lattice Diamond software. If the software is installed into a different directory, the script will need to be modified.

# Chapter 9



# **Advanced Topics**

This chapter explains advanced concepts, features and operational methods for Lattice Diamond.

### **Shared Memory Environment**

The Lattice Diamond design environment uses a shared memory architecture. Shared memory allows all internal tool views to access the same image of the design at any point in time. Understanding how shared memory is being used can give you insight into managing the environment for optimum performance, especially when your design is large.

There is one shared database that contains the device, design, and preference information in system memory.

Generating the hierarchy of your design uses an additional database separate from the main shared memory database.

External tools referenced from within Lattice Diamond, such as those for synthesis and simulation, use their own memory in addition to what is used by Lattice Diamond.

When you launch the first tool view that accesses shared memory, it will take longer than the launch of subsequent views.

## Memory Usage

The main window of the UI displays a memory usage figure in the lower right corner.

#### Figure 136: Memory Usage

Mem Usage: 235,148 K

This indicates the current memory usage for the Lattice Diamond environment, including all open tools.

# **Clear Tool Memory**

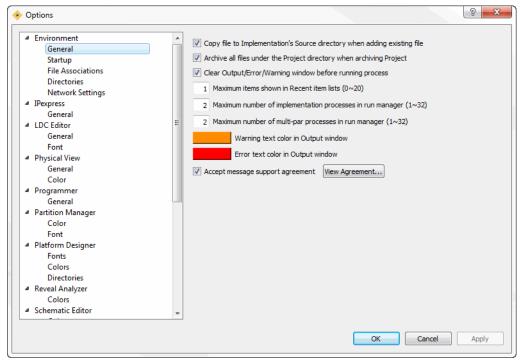
The "Clear Tool Memory command, available from the Tools menu, clears the device, design, and preference information from system memory. Clearing the tool memory can speed up memory-intensive processes such as place and route. When your design is very large, it is good practice to clear memory prior to running place and route.

If you have open tool views that will be affected by clearing the tool memory, a confirmation dialog box will open to give you the opportunity to cancel the memory clear.

### **Environment and Tool Options**

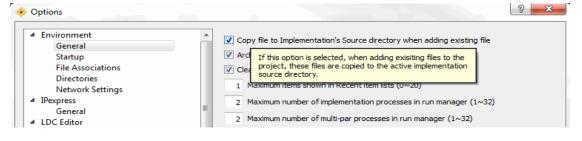
Lattice Diamond provides many environment control and tool view display options that enable you to customize settings. Choose **Tools > Options** to access these options.

Figure 137: Environment and Tool Options



The Options dialog box is organized into functional folders. You can use the context-sensitive help button in the upper right to view information about each parameter in a folder. Click the context-sensitive help button, and then click the item of interest.

Figure 138: Context-Sensitive Help Information



Commonly configured items include:

- Environment > General > Automatically Generate Hierarchy runs the Generate Hierarchy function whenever a project is opened.
- Environment > Startup enables you to configure the default action at startup and also to control the frequency of checking for software updates.
- Environment> File Associations allows you to set the programs to be associated with different file types based on the file extensions.

The tool view folders include customizable display properties such as color and font.

### **Pin Migration**

The "Incompatible Pins" dialog box helps you migrate pin assignments to a different device of the same family and package. This information enables you to save your current pinout while you explore other devices.

♦ Incompatible Pins	
Compatible Devices	
All Devices	
LFXP2-5E	
LAXP2-5E	
IFXP2-8E	
LAXP2-8E	
LAXP2-17E	
LFXP2-30E	
Check Areas	
Select All	
V Pin Type	
Configuration	
LVDS/HIGHSPEED	
Rank Number	
Disable Incompatible Pins Assignment	
OK Cancel Help	

#### Figure 139: Incompatible Pins Dialog Box

The "Incompatible Pins" dialog box is available from the View menu in Spreadsheet View or Package View after the Translate Design process. After selecting one or more devices for possible pin migration, you can view the incompatible pins in Package View and in the Pin Assignments sheet of Spreadsheet View. For more information about pin migration, refer to "Migrating Pin Assignments" in the Lattice Diamond online Help.

### **Batch Tool Operation**

The core tools in the FPGA implementation design flow can all be run in batch mode using command-line tool invocation or scripts. For detailed information, see the *Command Line Reference Guide*, available from the Lattice Diamond Start Page.

### **Tcl Scripts**

The Diamond Extended Tcl language enables you to create customized scripts for tasks that you perform often in Lattice Diamond. Automating these operations through Tcl scripts not only saves time, but also provides a uniform approach to design. This is especially useful when you try to find an optimal solution using numerous design implementations.

# Creating Tcl Scripts from Command History

A good first step in Tcl programming is to create a Tcl script by saving some command history and then modifying it as needed. This allows you to get started by using existing command information.

#### To create a Tcl command script using command history:

1. In the Tcl Console window, first perform a *reset* command so that your script won't contain any of the actions that may have already executed.

reset

- 2. Now perform the commands that you want to save as a script.
- Optionally, enter the history command in the Tcl Console window to ensure that the commands you wish to save are in the console's memory.
- 4. In the Tcl Console window type

save\_script <script\_name>

where <*script\_name*> is any identifier that has no spaces and contains no special characters except underscores. For example, *myscript* or *design\_flow\_1* are acceptable script name values, but *my*\$*script* and *my script* are invalid.

It is possible to give your *<script\_name>* value a file path and name if you want to keep it in a subfolder in your project or in a folder where you generally keep your scripts. However, you must save it to an existing folder. File paths, using forward slashes, with an identifier are valid if using an absolute file path to an existing folder.

 Navigate to your script file and use the text editing tool of your choice to make any necessary changes, such as deleting extraneous lines or invalid arguments.

In most cases, you will need to edit the script you saved and take out any invalid arguments or any commands that cannot be performed in the Lattice Diamond environment due to a conflict or exception. You will need to revisit this step later if, after running your script, you experience any run errors due to syntax errors or technology exceptions.

# **Creating Tcl Scripts from Scratch**

Tcl commands can be written directly into a script file. You can use any text editor, such as Notepad or vi, to create a file and type in the Tcl commands.

## Sample Tcl Script

The following Tcl example shows a simple script that opens a project, runs the entire design flow through the Place & Route process, and then closes the project. A typical script would probably contain more steps, but you can use this example as a general guideline.

```
prj_project open "C:/lscc/diamond/edif_counter/edif.ldf"
prj_run PAR -impl edif -forceAll
prj project close
```

## **Running Tcl Scripts**

You can run scripts from the Lattice Diamond integrated Tcl Console whether your project is opened or not. You can also run scripts from the external Tcl Console prompt window. The following example procedure uses the integrated Tcl Console and the sample Tcl script from the previous section:

# To run a Tcl script that opens a project, runs processes and closes the project:

- Open Lattice Diamond but do not open your project. If your project is open, choose File > Close Project.
- 2. In the Diamond main window, click the Tcl Console tab at the bottom to open the console.
- 3. If there are previously issued commands in the console, type reset in the console command line to refresh your session and clear out all previous commands.

reset

4. Run the source command using the absolute file name and path to your script.

source C:/lscc/diamond/1.4/examples/edif\_counter/myscript2

The sample Tcl script opens the project, runs the flow through Place & Route, and closes the project.

As long as there are no syntax errors or invalid arguments, this procedure should open your project, and you should see the processes running to completion.

If there are errors in the script, you will see the errors in red in the Tcl Console after you attempt to run it. You will then need to return to your script and edit it as needed.

### **Project Archiving**

A Diamond project archive is a single compressed file (.zip) of your project. The project archive can contain all of the files in your project directory, or it can be limited to source-related files. When you use the **File > Archive Project** command, the dialog box provides the option to "Archive all files under the Project directory." When you select this option, the entire project is archived. When you clear this option, only the project's source-related files, including strategies, are archived. Many of these source-related files must be archived in order to achieve the same bitstream results for a fully implemented design.

Whichever archiving method you select, if your project contains source files stored outside the project folder, the remote files will be compressed under the remote\_files subdirectory in the archive; for example:

<project\_name>/remote\_files/sources <project\_name>remote\_files/strategies

When unarchiving, you must manually move the archived remote files to the original locations or the project will not work.

Appendix A "File Descriptions" on page 143 provides lists of the file types used in Diamond, including those generated during design implementation. The Archive column indicates the files that must be archived in order to achieve the same bitstream results.

# Appendix



# **File Descriptions**

The following tables describe many of the files that are used in Lattice Diamond, including files generated by the implementation processes. A check mark in the Archive column indicates that a file of this particular type, when used in implementing the design, must be included in the project's archive in order to create the same bitstream results.

File Type	Definition	Function	Archive
.edf	Default EDIF source file generated by Precision	Used in design translation to create the .ngd netlist.	1
.edn	Default EDIF source file generated by Synplify	Used in design translation to create the .ngd netlist.	1
.fdc	FPGA Design Constraint file	Used for specifying design-specific constraints for SynplifyPro or Precision synthesis tools. FDC replaces the legacy SDC format.	V
.ipx	Manifest file generated by IPexpress.	Enables changes to be made to a module or IP IPexpress.	1
.ldc	SDC constraints file	Used for specifying design-specific constraints for the Lattice Synthesis Engine (LSE).	1
.ldf	Diamond Project file	Used for managing and implementing all project files in Diamond.	V
.lib	Schematic symbol library	Used for adding symbols to a schematic source file.	1
.lpf	Diamond project logical preference file	Stores logical constraints for developing and implementing the design.	1
.lsdc	SDC constraints file for Lattice Synthesis Engine (LSE)	Used for specifying design-specific constraints.	1

#### **Table 1: Source Files**

File Type	Definition	Function	Archive
.mem	Memory Generator source file	Specifies the initial contents for the memory modules in your design. Required for creating ROM and optional for creating RAM modules.	1
.ngo	Native Generic Object (NGO) netlist format	Used as an alternative to HDL for creating a black-box module.	V
.pcf	Power Calculator project file	Stores power analysis results from information extracted from the design project.	V
.ptm	Platform Designer project file	Holds analog sense and control settings, logic, and port information, for the project.	V
.rva	Reveal Analyzer file	Defines the Reveal Analyzer project and contains data about the display of signals in Waveform View.	V
.rvl	Reveal Inserter debug file	Defines the Reveal project and its modules, identifies the trace and trigger signals, and stores the trace and trigger options.	V
sch	Schematic source file	Schematic representation of the circuit in terms of the components used and how they connect to each other.	V
sdc	SDC constraints file	Used for specifying design-specific constraints for SynplifyPro or Precision synthesis tools. SDC is replaced by the FDC format in but is still supported in Diamond.	V
spf	Simulation project file, a script file produced by the Simulation Wizard	Used for running the simulator for your project from Diamond.	J
sty	Strategy file	Defines the optimization control settings of implementation tools such as logic synthesis, mapping, and place and route.	V
sym	Symbol Editor source file	Used for creating symbols or primitive elements that represent an independent schematic module.	V
tpf	Timing Analysis Preference file	Used for obtaining quick analysis of timing preferences.	J
v	Verilog source file	Verilog description of the circuit structure and function	J
.vhd	VHDL source file	VHDL description of the circuit structure and function.	V
.wdl	Waveform Editor source file	Used for creating test stimulus files for simulation.	V
xcf	Configuration chain file	Used for programming devices in a JTAG daisy chain.	V

#### Table 1: Source Files (Continued)

### Table 2: IPexpress Files

File Type	Definition	Function	Archive?
Module Files			
%instName.lpc	Module parameter configuration file	Stores the user configurations for the module.	V
%instName.v	Verilog module file	Verilog netlist generated by IPexpress to match the user configuration of the module.	V
%instName.vhd	VHDL module file	VHDL netlist generated by IPexpress to match the user configuration of the module.	V
%instName_tmpl.v	Verilog template file	A template for instantiating the generated module. This file can be copied into a user Verilog file.	V
%instName_tmpl.vhd	VHDL module template file	A template for instantiating the generated module. This file can be copied into a user VHDL file.	~
tb_%instName_tmpl.v	Verilog testbench template	A simple Verilog testbench for the generated module. Can be edited to add more vectors, which will be overwritten during module regeneration.	V
tb_%instName_tmpl.vhd	VHDL module testbench file	A simple VHDL testbench for the generated module. Can be edited to add more vectors, which will be overwritten during module generation.	V
IP Files			
%instName.lpc	IP parameter configuration file	Used for re-creating or modifying the core in the IPexpress tool.	V
%instName_bb.v	Verilog IP black box file	Provides the Verilog synthesis black box for the IP core and defines the port list.	V
%instName_beh.v	Verilog IP behavioral file	Provides a behavioral simulation model for the IP core.	1
%instName_inst.v	Verilog IP instantiation file	Provides a Verilog instance template for the IP core.	1
%instName_inst.vhd	VHDL IP instantiation file	Provides a VHDL instance template for some IP cores. Archiving this file, though optional, will make it easier to write a top- level VHDL file and copy and paste the instantiation section from this file.	
%instName.ngo	IP database file	Provides the netlist in encrypted format for the IP core and is used by Map and PAR tools.	V
pmi_*.ngo	Support database files	Provides the netlist files in encrypted format for the Lattice parameterizable module instances (PMI) used within the IP core.	J

File Type	Definition	Function	Archive?
%IPName_params.v	IP parameters file	Contains the Verilog parameters used by the IP core simulation model.	V
*.txt	SERDES configuration file	Contains the SERDES configuration that is required during bitgen process.	V

### Table 2: IPexpress Files (Continued)

#### **Table 3: Implementation Files**

File Type	Definition	Function	Archive?
.bgn	Bitstream generation report file	Reports results of a bit generation (bitgen) run and displays information on options that were set.	
.bit	Bitstream file	Used for SRAM FPGA programming.	
.edi	SynplifyPro EDIF output file	Netlist file generated by Diamond. The file extension must be changed to .edf or .edn in order to import it into a project.	
ibs	Post-Route I/O buffer information specification file (IBIS)	Used for analyzing signal integrity and electromagnetic compatibility (EMC) on printed circuit boards	
ior	Post-PAR I/O Timing Analysis file	For each input data port, reports setup and hold time requirements and min/max clock-to- out delay for each output port.	
jed	JEDEC file	Used for Flash FPGA programming	
_map.ncd	Mapped native circuit description netlist file.	Includes mapping information.	
_mapvho.sdf	Post-map SDF simulation file for VHDL	Used for post-map functional simulation.	
_mapvho.vho	Post-map simulation file for VHDL	Used for post-map functional simulation.	
_mapvo.sdf	Post-map SDF simulation file for Verilog	Used for post-map functional simulation.	
_mapvo.vo	Post-map simulation file for Verilog	Used for post-map functional simulation.	
mcs	PROM file	Used for SRAM FPGA programming	
.mrp	Map Report file	Provides statistics about component usage in the mapped design.	
ncd	Post-route native circuit description netlist file	Includes placement and routing information.	
ngd	Native generic description file produced by the ngdbuild translation process.	Used by Map to map logical elements to physical elements in the native circuit description file (.ncd).	
ngo	Native generic object file generated by the edif2ngo translation process. Contains logical descriptions of the design's components and hierarchy	Used as input to ngdbuild, which converts the .ngo to an .ngd file.	

File Type	Definition	Function	Archive?
.pad	Post-Route PAD report file	Lists all programmable I/O cells used in the design and their associated primary pins.	
.par	Post-Route Place & Route report file	Summarizes information from all iterations and shows the steps taken to achieve a placement and routing solution.	
.prf	Physical preference file produced by the Map Design process.	Used as an input file to placement, routing and TRACE.	
.\$\$0	Post-PAR SSO analysis file	Reports the noise caused by simultaneously switching outputs.	
.tw1	Post-Map TRACE analysis file	Estimates pre-route timing.	
.twr	Post-PAR TRACE analysis file	Reports post-route timing.	
.vho	Post-Route VHDL simulation file	Used for post-route simulation.	
.vo	Post-Route Verilog simulation file	Used for post-route simulation.	
_vho.sdf	Post-Route SDF simulation file for VHDL	Used for timing simulation.	
_vo.sdf	Post-Route SDF simulation file for Verilog	Used for timing simulation.	

### Table 3: Implementation Files (Continued)

### Table 4: Incremental Design Flow Files

File Type	Description	Function	Archive?
.icf	Incremental Compilation Database File	Contains design partition placement and compilation strategy directives for incremental design flow.	V
\inc1\*	First backup of incremental files		
\inc2\*	Second backup of incremental files		
\golden\*	Golden backup of incremental files		