



Synthesis Data Flow Tutorial

Lattice Semiconductor Corporation
5555 NE Moore Court
Hillsboro, OR 97124
(503) 268-8000

November 2008

Copyright

Copyright © 2008 Lattice Semiconductor Corporation.

This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Lattice Semiconductor Corporation.

Trademarks

Lattice Semiconductor Corporation, L Lattice Semiconductor Corporation (logo), L (stylized), L (design), Lattice (design), LSC, E²CMOS, Extreme Performance, FlashBAK, flexiFlash, flexiMAC, flexiPCS, FreedomChip, GAL, GDX, Generic Array Logic, HDL Explorer, IPexpress, ISP, ispATE, ispClock, ispDOWNLOAD, ispGAL, ispGDS, ispGDX, ispGDXV, ispGDX2, ispGENERATOR, ispJTAG, ispLEVER, ispLeverCORE, ispLSI, ispMACH, ispPAC, ispTRACY, ispTURBO, ispVIRTUAL MACHINE, ispVM, ispXP, ispXPGA, ispXPLD, LatticeEC, LatticeECP, LatticeECP-DSP, LatticeECP2, LatticeECP2M, LatticeMico8, LatticeMico32, LatticeSC, LatticeSCM, LatticeXP, LatticeXP2, MACH, MachXO, MACO, ORCA, PAC, PAC-Designer, PAL, Performance Analyst, PURESPEED, Reveal, Silicon Forest, Speedlocked, Speed Locking, SuperBIG, SuperCOOL, SuperFAST, SuperWIDE, sysCLOCK, sysCONFIG, sysDSP, sysHSI, sysI/O, sysMEM, The Simple Machine for Complex Design, TransFR, UltraMOS, and specific product designations are either registered trademarks or trademarks of Lattice Semiconductor Corporation or its subsidiaries in the United States and/or other countries. ISP, Bringing the Best Together, and More of the Best are service marks of Lattice Semiconductor Corporation.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium in the U.S. and other jurisdictions.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

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 SEMICONDUCTOR CORPORATION (LSC) 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 LSC 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.

LSC may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. LSC makes no commitment to update this documentation. LSC 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. LSC recommends its customers obtain the latest version of the relevant information to establish, before ordering, that the information being relied upon is current.

Type Conventions Used in This Document

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

Contents

Learning Objectives	1
Time to Complete This Tutorial	2
System Requirements	2
Accessing Online Help	2
About the Tutorial Design	2
About the Tutorial Data Flow	3
Task 1: Synthesize the Design	4
Create a Synthesis Project	4
Add the Verilog HDL Source File	5
Set the Target Device	6
Compile and Set Timing	7
Synthesize the Design	9
Task 2: Create a New Project Navigator Project	10
Task 3: Target a Device	11
Task 4: Import Predefined Constraints	14
Task 5: Map the Design	14
Task 6: Place and Route the Design	16
Task 7: Perform Static Timing Analysis	17
Summary	19
Glossary	19
Recommended References	20

Synthesis Data Flow Tutorial

This tutorial shows you how to use Synopsys® Synplify Pro® for Lattice with ispLEVER® to synthesize a Verilog HDL design and to generate an EDIF file for a Lattice FPGA device. It is intended for the user who is using Synplify Pro for the first time and who is looking for the basic data and process flow from HDL source code to a fully implemented FPGA. For more information on FPGA design strategy and techniques with ispLEVER, see the *FPGA Design Guide*. For a more in-depth tutorial on ispLEVER tools, timing closure techniques, DSP design, or on-chip debugging, see the ispLEVER “Tutorials” section of the online help.

Note

If you want to learn how to use Synplify Pro in stand-alone mode or understand more about its advanced features, see its manuals.

Learning Objectives

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

- ◆ Start Synplify Pro, synthesize your Verilog HDL design, and generate an EDIF netlist file.
- ◆ Create a new EDIF project in ispLEVER and target a device.
- ◆ Import the EDIF file and a predefined constraints file into ispLEVER and implement the design using the map, place, and route processes.
- ◆ Set report viewing options and view the reports.
- ◆ Run static timing analysis using Performance Analyst and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 20 minutes.

System Requirements

One of the following software configurations is required to complete the tutorial:

- ◆ ispLEVER Starter
- ◆ ispLEVER

Accessing Online Help

You can find online help information on any tool included in the tutorial at any time by pressing the F1 key.

About the Tutorial Design

The tutorial design demonstrates a telecommunications application that consists of a data frame aligner that reads an unaligned 33-bit data set from a 66-bit data stream to produce a 64-bit aligned data stream. Frame lock status and bit error rate (BER) outputs are available. The system operates at 156 MHz.

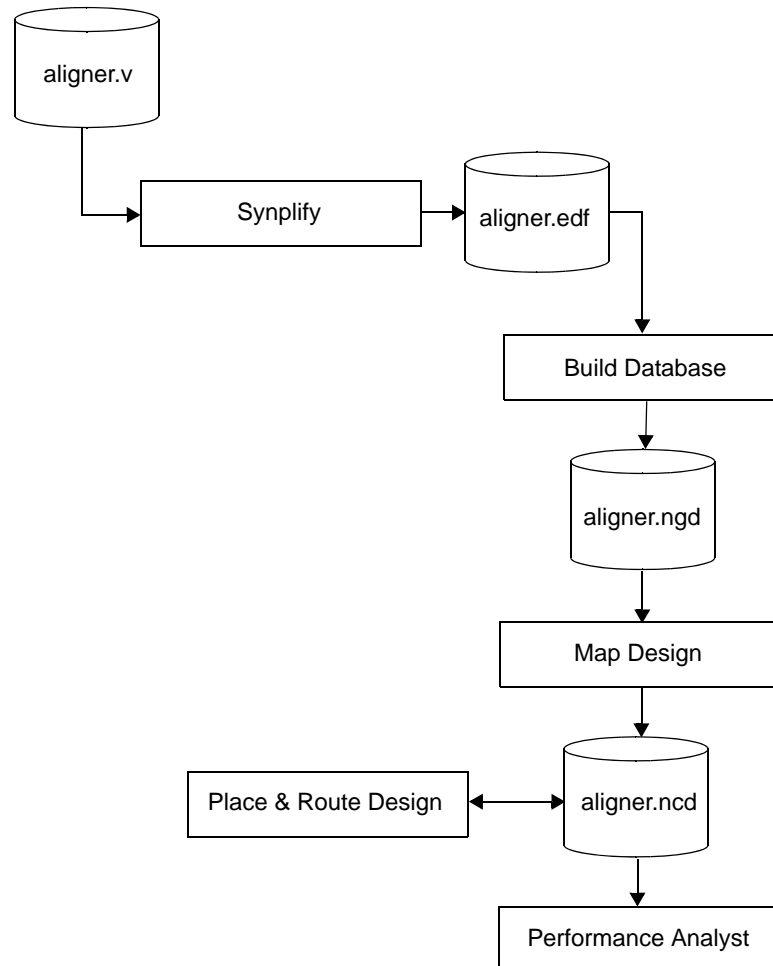
Given the relative design size and performance requirements, the tutorial design is targeted to a Lattice MachXO 1200 FTBGA256 device (LCMXO1200), which provides 1200 LUTs and 159 I/Os.

This tutorial first directs you to create a new project in Synplify Pro. After you import the Verilog HDL source file and set the implementation options and timing objectives, the tool synthesizes the design into the target device and generates an EDIF netlist. Next, you create an EDIF project in Project Navigator, then select the target device in which the design will be implemented. The tutorial assumes that functional simulation has already been performed. You then import the EDIF netlist into the Project Navigator project and map, place, and route the design. Finally, you perform a static timing analysis and examine the results.

About the Tutorial Data Flow

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

Figure 1: Tutorial Design Flow



A logic synthesis tool starts with a high-level design written in Verilog or VHDL hardware description languages (HDLs). Then it converts the HDL description into a netlist, usually in the EDIF 2 0 0 format, optimized for Lattice devices.

For HDL designs, the Windows version of ispLEVER provides two synthesis tools that are integrated into the Project Navigator environment: Synplify and Synplify Pro. Precision RTL Synthesis can also be added as an integrated tool. The UNIX and Linux versions of ispLEVER provide Synplify Pro as a stand-alone tool. Other synthesis tools can also be used stand-alone.

You can synthesize your Verilog or VHDL design as a stand-alone process by opening the synthesis tool on its own, interactively with ispLEVER by opening the tool in Project Navigator, or you can synthesize automatically and seamlessly from within Project Navigator. The primary advantage of using the

logic synthesis front-end user interface outside Project Navigator is the flexibility it gives you in defining timing and optimization constraints for design-specific elements. Lattice Applications Engineering typically recommends this “EDIF Flow” into the ispLEVER Project Navigator for all except the smallest logic designs.

Task 1: Synthesize the Design

The first step in the tutorial is to synthesize the design before you open Project Navigator.

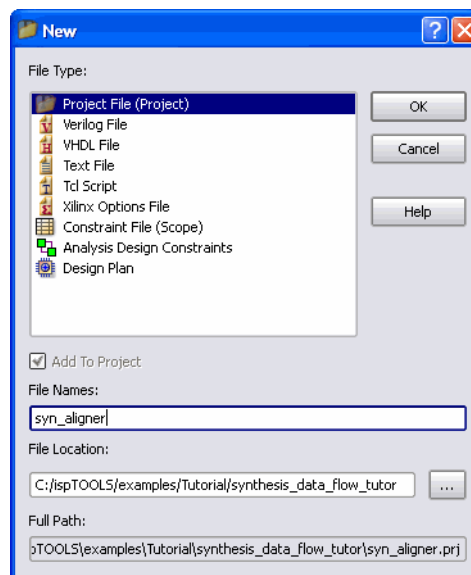
Create a Synthesis Project

First, you create a Synplify Pro project.

To start Synplify Pro:

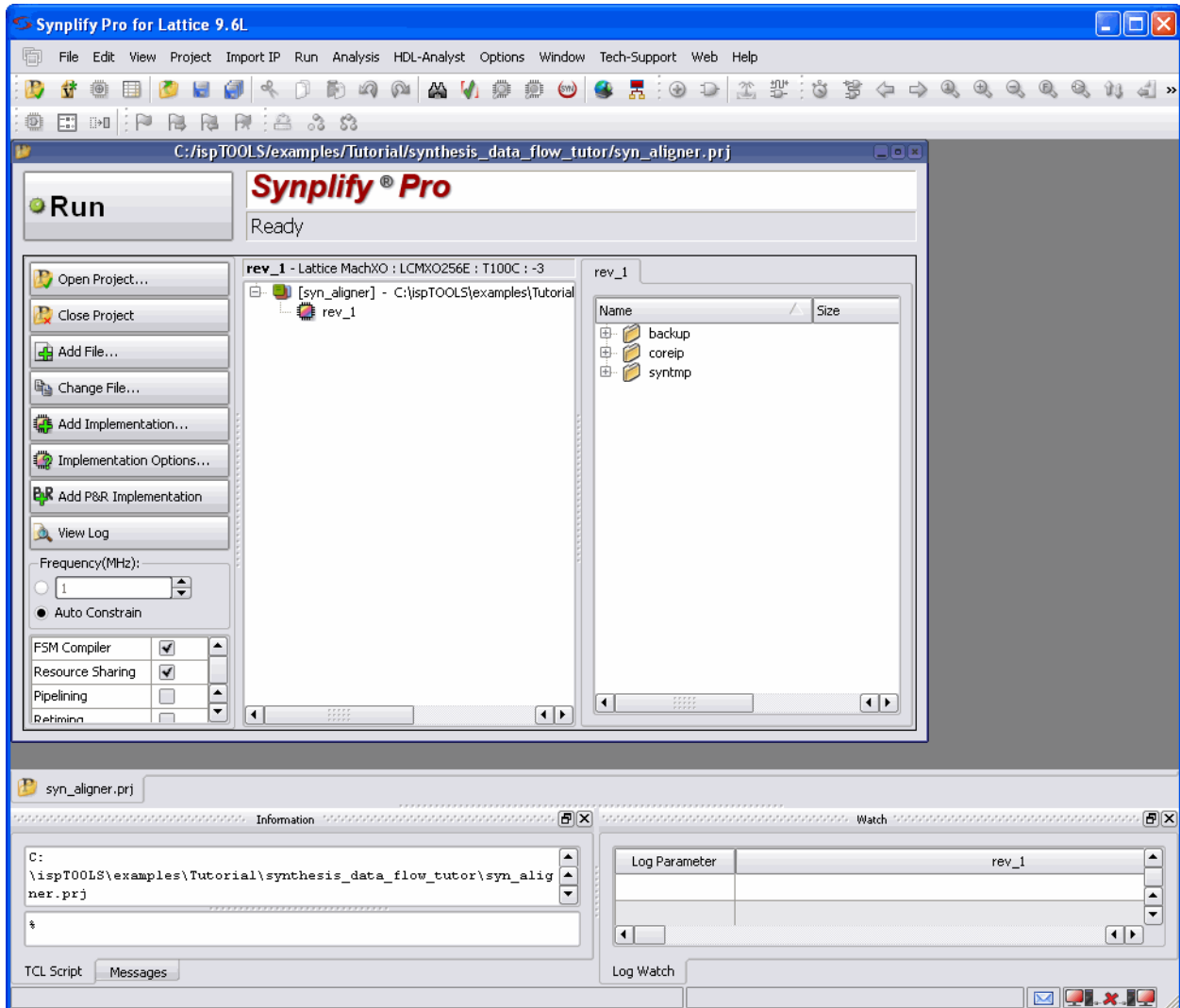
1. Choose **Start > All Programs > Lattice Semiconductor > Accessories > Synplify Pro for Lattice**.
2. Click **OK** to close the Tip of the Day.
3. Choose **File > New**.
4. In the New dialog box, shown in Figure 2, do the following:
 - a. In the File Type box, select **Project File (Project)**.
 - b. In the File Names box, type **syn_aligner**.
 - c. Under File Location, make sure you are in the following directory:
`<install_path>\examples\Tutorial\synthesis_data_flow_tutor`
 - d. Click **OK**.

Figure 2: New Dialog Box



The Synplify Pro window looks like the example shown in Figure 3.

Figure 3: Synplify Pro Window Showing syn_aligner Project



Add the Verilog HDL Source File

In this task, you will add Verilog source files to the project.

To add source files to the project:

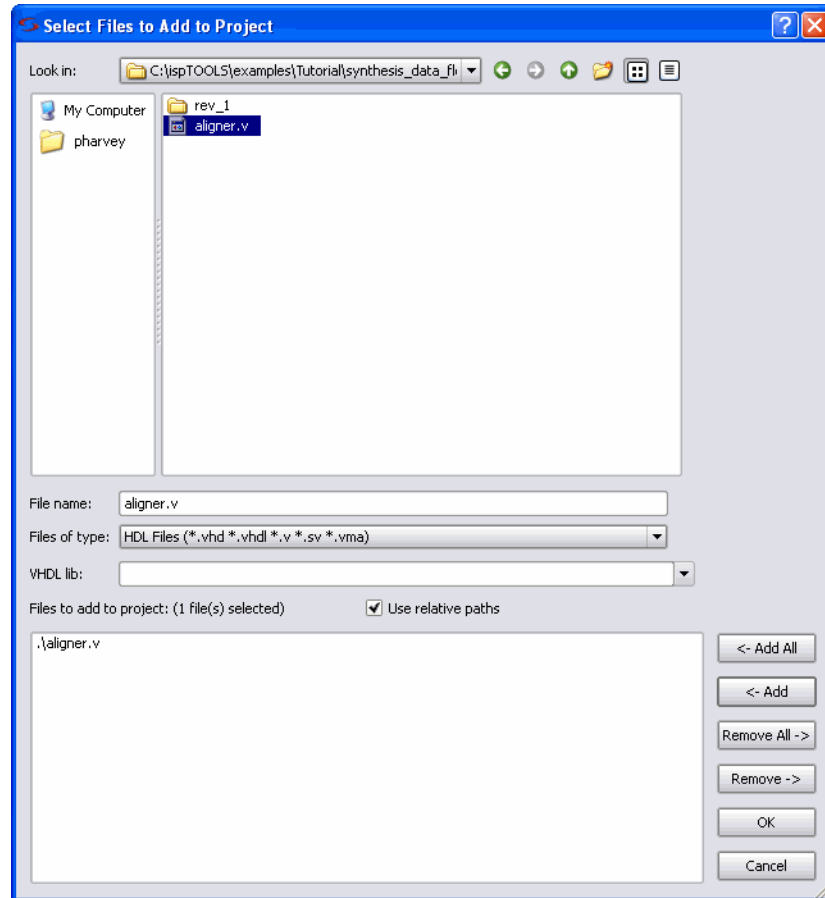
1. On the Synplify Pro main window, click **Add File** to open the Select Files to Add to Project dialog box, shown in Figure 4.

You should see one Verilog file named aligner.v.

2. Select the **aligner.v** file and click **<- Add** to add the Verilog file to the project (shown in the Files To Add To Project field).

3. Click **OK** to close the dialog box.

Figure 4: Select Files to Add to Project Dialog Box



4. In Synplify Pro, expand the Verilog folder by clicking on the plus sign (+).

Set the Target Device

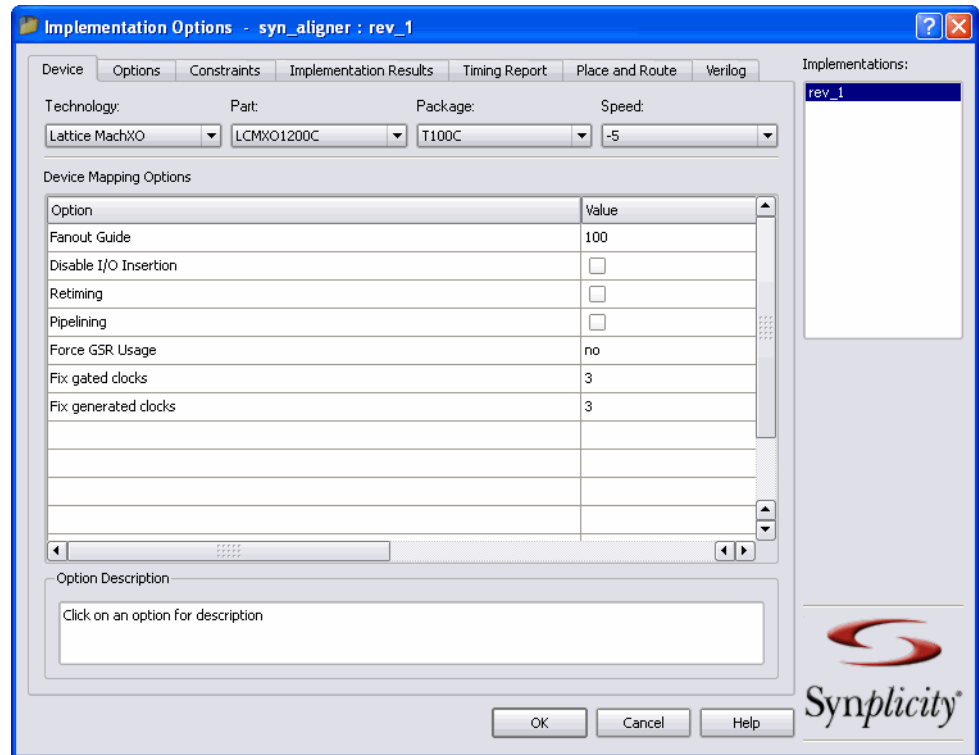
In this task, you set the target device for logic synthesis.

To set the implementation options:

1. Click **Implementation Options** to open the Implementation Options dialog box, shown in Figure 5.
 - a. Under Technology, select **Lattice MachXO**.
 - b. Under Part, select **LCMXO1200C**.
 - c. Under Speed, select **-5**.
 - d. Under Package, select **FT256C**.
2. Click the **Implementation Results** tab.

This dialog box enables you to change the file name or destination of the output EDIF file. In this tutorial, you will not change anything, so all the project files remain in the current project directory. Synplify Pro

Figure 5: Options for Implementation Dialog Box



automatically creates revision folders inside your project folder. You should be in the rev_1 folder inside your project folder.

3. Click **OK** to close the dialog box.

Compile and Set Timing

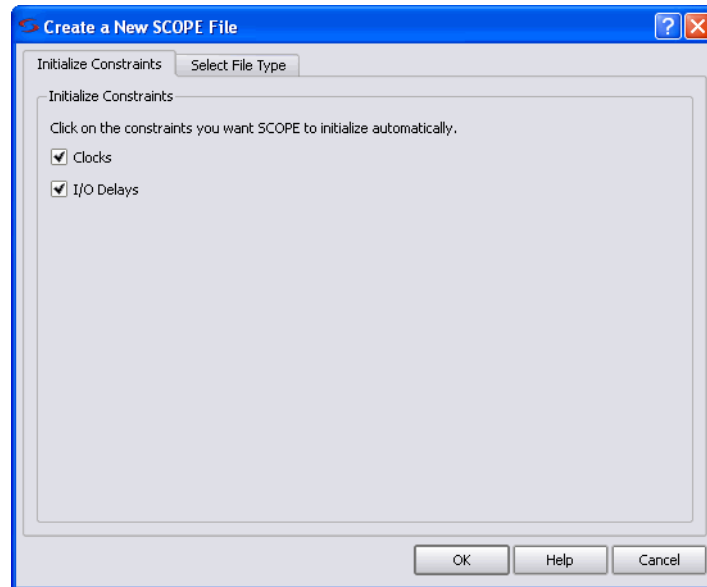
In this task, you compile and then specify your timing objectives for logic synthesis.

To compile the design:

1. Select **[syn_aligner]**, click the right mouse button, and choose **Resynthesize All**.
2. Choose **File > New**.
The New dialog box appears.
3. Select **Constraint File (Scope)**.
4. Specify **df_aligner.sdc** in the File Names box.
5. In the File Location box, browse to the following directory:
`<install_path>\examples\Tutorial\synthesis_data_flow_tutor`
6. Click **OK**.

The Create a New SCOPE File dialog box now appears, as shown in Figure 6.

Figure 6: Create a New SCOPE File Dialog Box



7. Make sure that the Clocks and I/O Delays options are turned on, then click **OK**.

The df_aligner_1.sdc constraint file appears in the SCOPE editor.

8. Select the **Clocks** tab, if it is not already selected, and enable the **clk_156mhz** port.
9. Specify **182** in the **Frequency (MHz)** cell and press **Enter**.
10. Select the **Inputs/Outputs** tab and enable **<input default>** and **<output default>**.
11. In both rows, double-click in the Clock Edge cells and choose **clk_156mhz:r** from the drop-down menu.
12. Specify **6** in the **Value (ns)** cells for both rows, and press **Enter**.

Figure 7 shows the completed Inputs/Outputs tab.

Figure 7: SCOPE Editor Dialog Box

	Enabled	Port	Type	Clock Edge	Value (ns)	Route (ns)	Comment
1	<input checked="" type="checkbox"/>	<input default>	input_delay	clk_156m...	6		
2	<input checked="" type="checkbox"/>	<output default>	output_delay	clk_156m...	6		
3	<input type="checkbox"/>	unalign_data_66b[...	input_delay				
4	<input type="checkbox"/>	align_data_64b[63...	output_delay				
5	<input type="checkbox"/>	sync_bits[1:0]	output_delay				
6	<input type="checkbox"/>	slip	output_delay				
7	<input type="checkbox"/>	frm_lock	output_delay				
8	<input type="checkbox"/>	hi_ber	output_delay				
9	<input type="checkbox"/>	channel_rdy	input_delay				
10	<input type="checkbox"/>	sys_reset_z	input_delay				
11							
12							
13							

13. Choose **File > Save** and click **Save** in the Save As dialog box.

14. In the box displaying the following prompt, click **Yes**.

```
Do you wish to add the file C:\ispTOOLS\examples\Tutorial
\synthesis_data_flow_tutor\df_aligner_1.sdc to the project?
```

15. Close the SCOPE editor.

Note

Only rarely will you be able to estimate the device capacity and speed achievable for a new design on the first pass through logic synthesis. You may need to iterate a few times to find the right device and package. For your convenience, it has already been determined that an LCMXO1200 device and a fpBGA256 package is a good match for the data frame aligner design.

Synthesize the Design

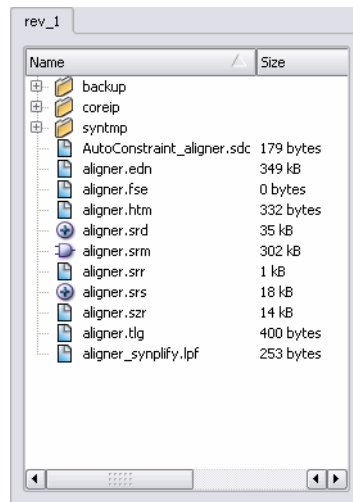
Now you are ready to synthesize your design.

To synthesize the design:

1. Click the large **Run** button to start the synthesis process.

Synplify Pro synthesizes the Verilog design and creates an EDIF file, as well as several other files, and displays them in the rev_1 tab, as shown in Figure 8.

If you like, you can double-click the different file names and view them. When you are finished, close the files.

Figure 8: Synplify Pro Window After Synthesis

2. Choose **File > Save** to save the Synplify Pro project.
3. Choose **File > Exit** to exit Synplify Pro.

You will activate the Project Navigator Main window again in “Task 2: Create a New Project Navigator Project” on page 10.

Task 2: Create a New Project Navigator Project

To begin a new project in Project Navigator, you must create a project directory. Then you give the project file a name (.syn) and declare the project type (EDIF).

The ispLEVER software saves an initial design file with the .syn file extension in the directory that you specify. All project files are copied to or created in this directory. The project type specifies that all design sources will be of this type.

To create a new project:

1. Start Project Navigator, if it is not already running.
2. In Project Navigator, choose **File > New Project** to open the Project Wizard dialog box, shown in Figure 9.
3. In the dialog box, do the following:
 - a. In the Project Name box, type **df_aligner**.
 - b. In the Location box, change to the following directory:

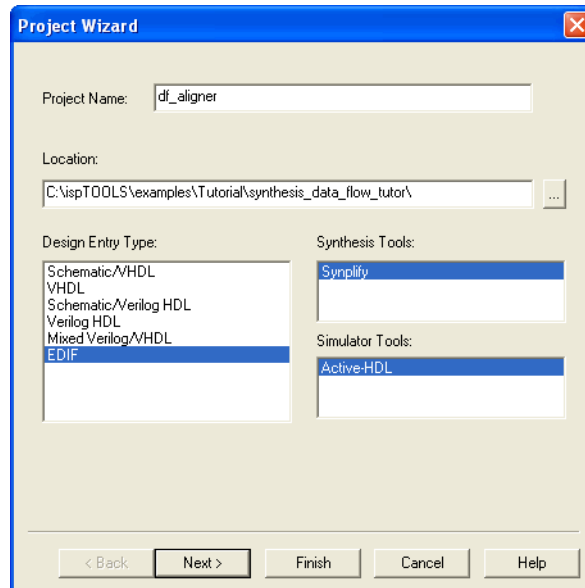
`<install_path>\examples\Tutorial\synthesis_data_flow_tutor`

Note

If you want to preserve the original tutorial design files, save the synthesis_data_flow_tutor directory to another location on your computer before proceeding.

- c. In the Design Entry Type box, select **EDIF**.
- d. In the Synthesis Tools box, select **Synplify**.
- e. In the Simulator Tools box, select your preferred tool.
- f. Click **Next** to activate the Project Wizard – Select Device dialog box.

Figure 9: Project Wizard Dialog Box

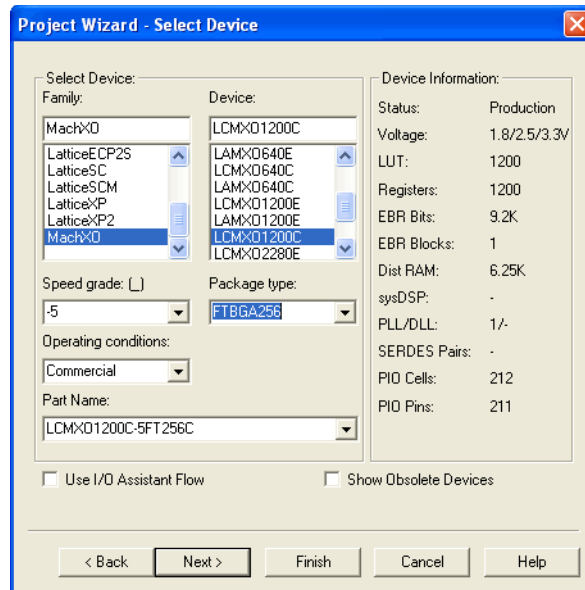


Task 3: Target a Device

In the Project Navigator Sources in Project window, the device icon appears next to the target device for the project. Project Navigator enables you to target a design to a specific Lattice Semiconductor device at any time during the design process. The default device is ispLSI5256VE-165LF256. For this project, you will target a different device.

To view the list of available devices and to change the target device:

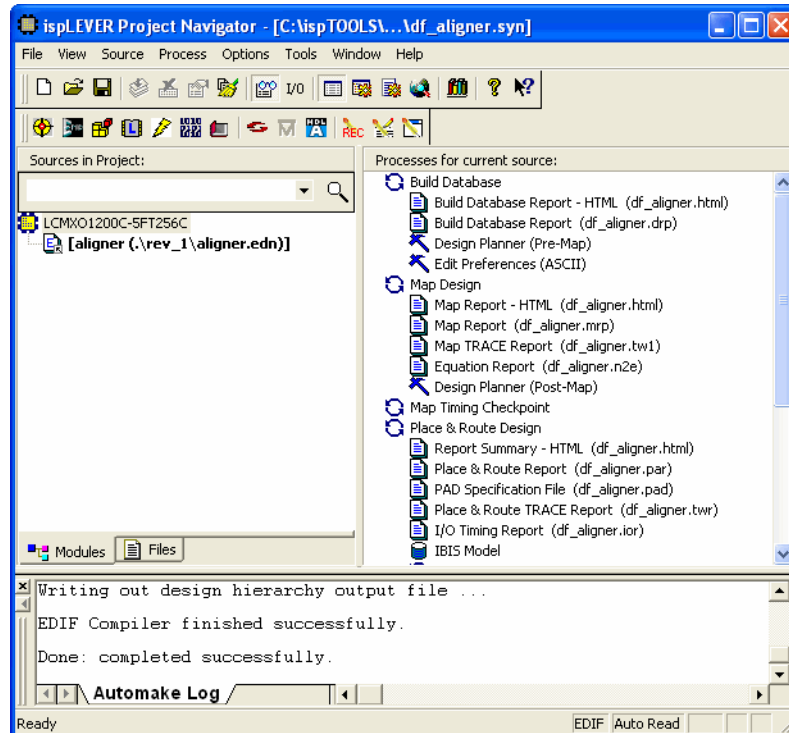
1. In the Project Wizard – Select Device dialog box, shown in Figure 10, do the following:
 - a. In the Family box, choose **MachXO**.
 - b. In the Device box, choose **LCMXO1200C**.
 - c. In the Speed Grade: box choose **-5**.
 - d. In the Package type box, choose **FTBGA256**.
 - e. Click **Next** to activate the Project Wizard - Add Source dialog box.

Figure 10: Project Wizard - Select a Device Dialog Box

2. In the Project Wizard – Add Source dialog box, click **Add Source**.
3. Select the EDIF file by browsing to the .rev_1 folder in the Import File (EDIF) dialog box and select **aligner.edn**. Click **Open**.
4. In the Project Wizard - Add Source dialog box, click **Next**, then click **Finish**.

Project Navigator displays the imported EDIF file, as shown in Figure 11.

Figure 11: Project Navigator Window Showing Imported EDIF File



Note

Click on the part name to see the contents of the Processes for Current Source window.

Note

After you import an EDIF file into the ispLEVER project, it is always linked to Project Navigator. Therefore, if you make changes and recompile your HDL file to create a new EDIF file, your project is automatically updated as well.

5. Choose **Tools > Timing Checkpoint Options** to open the Timing Checkpoint Options dialog box.
6. In the Checkpoint Options section, select "If Checkpoint fails: **Continue**".
7. Click **OK** to close the dialog box.

Task 4: Import Predefined Constraints

Now you are ready to import timing and location constraints prepared for the tutorial design into the ispLEVER project. You can import constraint files from third-party synthesis tools, such as Synplify Pro, into ispLEVER. These files are often good starting points for your own specifications and refinements to meet your system-level I/O placement requirements and timing objectives.

For more information on the ispLEVER preference language and the Design Planner, see the Process Flow online Help.

To import a constraint file into your project:

1. In Project Navigator, choose **Source > Import Constraint/Preference File** to open the Import Constraint File dialog box.
2. Go up one level to the `synthesis_data_flow_tutor` folder, and select **All Files** in the Files of Type box.
3. Select the constraint file by selecting **df_aligner_s.lpf**. Click **Open**.
4. Click **OK** in the Import Constraints information dialog box.
5. Click **Yes** to the `Do you want to reset the project status?` prompt.

Task 5: Map the Design

The ispLEVER software has a single user interface with all options preset to deliver the highest possible push-button performance for most devices. When you double-click a process, all the processes prior to that process run automatically. Therefore, all you have to do is double-click the final process. However, here you will run one process at a time and view the results as you go.

After an initial internal database is generated from the EDIF netlist, the Map Design process maps logical elements produced by logic synthesis into physical elements like slices, sysMEM EBRs, or sysDSP blocks of the device. It generates a file that you can submit to the Place & Route Design process.

To map the design and view the reports:

1. In Project Navigator, select the **LCMXO1200C-5FT256C** device in the Sources in Project window.
2. Double-click on the **Map Design** process in the Processes for current source window in Project Navigator.

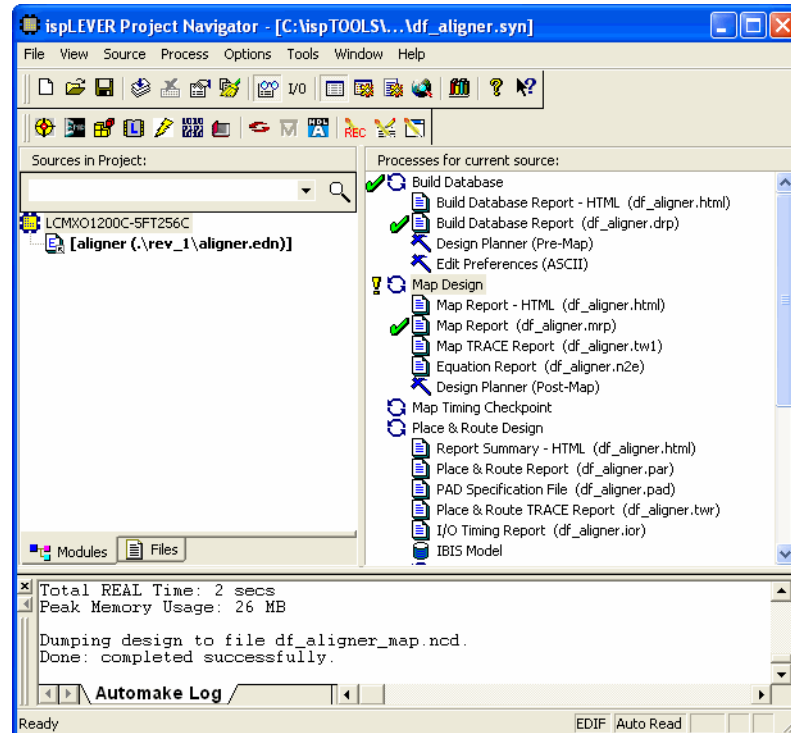
Note

You can ignore the warnings.

3. In the ispLEVER process message box, click **OK**.

Figure 12 shows the results in the Project Navigator window.

Figure 12: Project Navigator Window After Mapping



4. Double-click **Map Report** in Project Navigator to view the design mapping report.

After a few moments, the map report appears in the `df_aligner.mrp` tab in the output panel. This report gives details about the design's mapping and includes the following sections:

- ◆ Design Information
- ◆ Design Summary
- ◆ Design Errors/Warnings
- ◆ I/O (PIO) Attributes
- ◆ Removed Logic
- ◆ GSR Usage
- ◆ PGROUP Utilization
- ◆ Run Time and Memory Usage

Task 6: Place and Route the Design

The ispLEVER software places and routes the design in the device after it is mapped. The timing-driven placement and routing algorithm takes full advantage of the Lattice FPGA architecture to achieve maximum performance.

To place and route the design and view the reports:

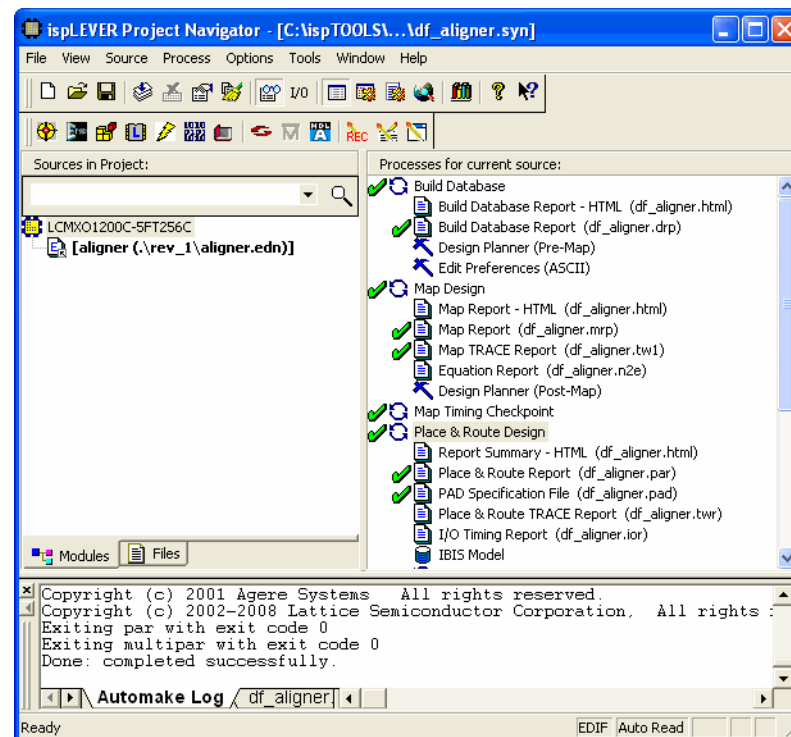
1. In Project Navigator, select the **LCMXO1200C-5FT256C** device in the Sources in Project window.
2. Double-click the **Place & Route Design** process in the Processes for current source window in Project Navigator.

The Project Navigator screen should resemble that shown in Figure 13.

Note

You can ignore the warnings.

Figure 13: Project Navigator After Placement and Routing



3. Double-click **Place & Route Report**.

The report appears in the transcript window of Project Navigator and shows how a design was routed in the target device and informs you of the success or failure of the routing.

Task 7: Perform Static Timing Analysis

Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data, such as the critical path, setup- and hold-time requirements, and the maximum frequency.

Performance Analyst traces each logical path in the design and calculates the path delays using the device's timing model and worst-case AC specifications supplied in the device data sheet.

The timing analysis results are displayed in a spreadsheet with source and destination signals. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To perform a timing analysis:

1. In the Project Navigator Sources in Project window, select the target device.
2. In the Processes for current source window, double-click the **Performance Analyst** process to run the timing analysis and open Performance Analyst.

The Performance Analyst performs three distinct analyses:

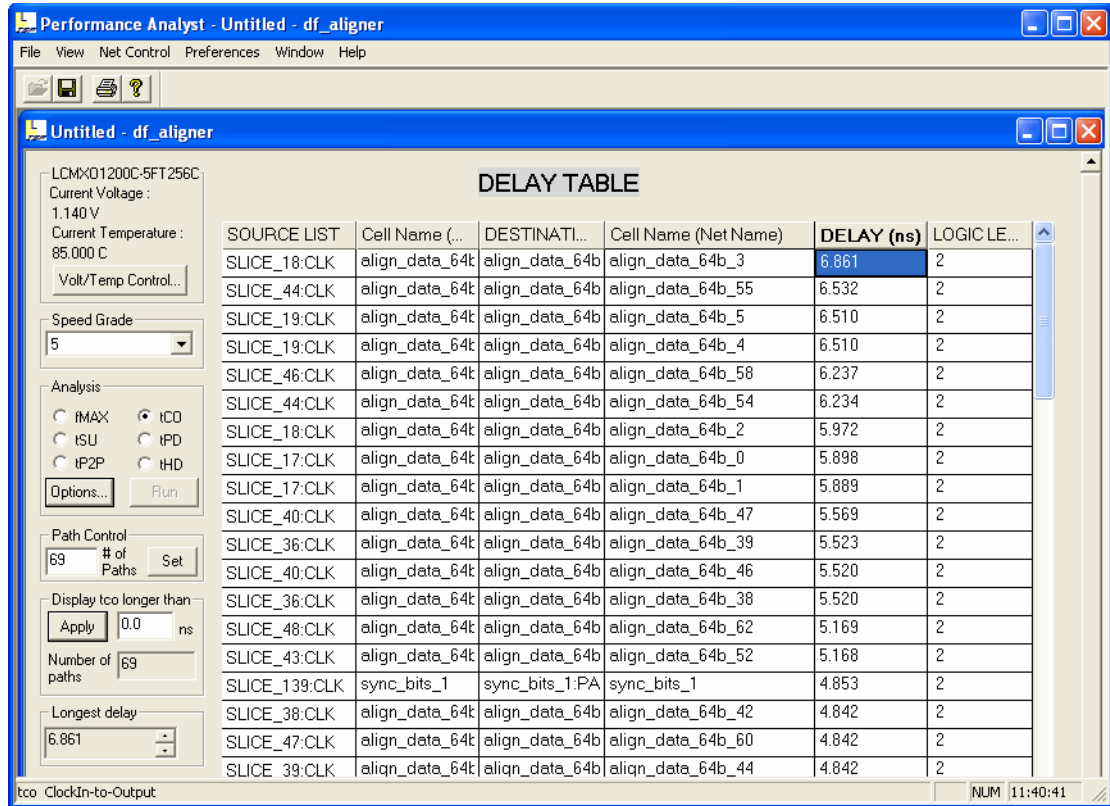
- ◆ fMAX analysis is an internal register-to-register delay analysis and measures the maximum clock operating frequency, limited by the worst-case register-to-register delay.
 - ◆ tSU analysis reports setup- and hold-time for data and clock enable signals with respect to a clock edge.
 - ◆ tCO analysis reports clock-to-out delay starting from the primary input, going through the clock of flip-flops or gate of latches, and ending at the primary output.
3. In the Analysis field, select **tCO**.

In the Delay column, the tCO path trace analysis reports clock-to-out delays, as shown in Figure 14.

Note

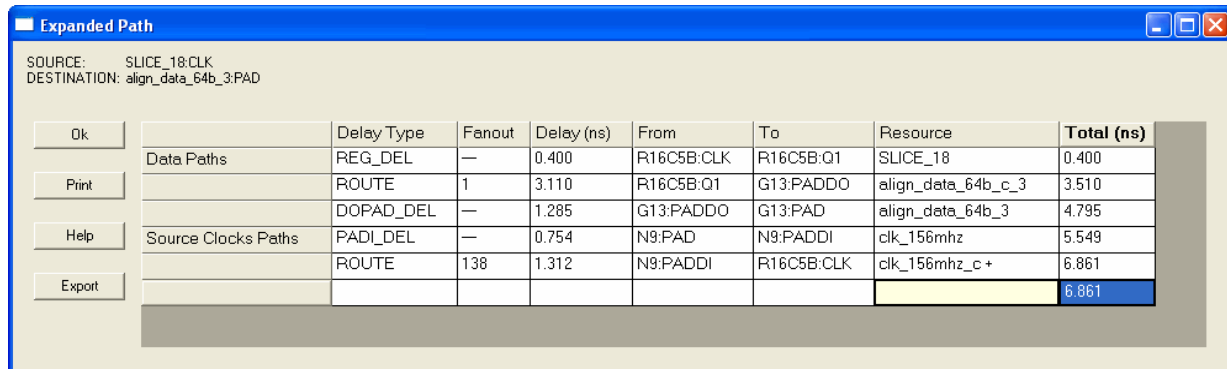
Your timing results may differ slightly.

Figure 14: Performance Analyst Window Showing tCO Analysis Results



4. Double-click the highlighted cell in the DELAY column to open the Expanded Path dialog box, shown in Figure 15.

Figure 15: Expanded Path Dialog Box



This dialog box enables you to analyze individual timing components used to calculate the timing path. It shows a source pin (From) and a destination pin (To). It also shows the delay type, the delay of that path, and the cumulative delay of all the signals.

5. Click **Ok** to close the dialog box.
6. Choose **File > Exit** to exit Performance Analyst without saving.
7. Choose **File > Exit** to exit Project Navigator without saving.

Summary

You have completed the “Synthesis Data Flow Tutorial.” In this tutorial, you have learned how to do the following:

- ◆ Start Synplify Pro from outside ispLEVER, synthesize your Verilog HDL design, and generate an EDIF netlist file.
- ◆ Use ispLEVER to create a new EDIF project and target a device.
- ◆ Import the EDIF and predefined constraint file into ispLEVER.
- ◆ Implement the design using the map, place, and route processes.
- ◆ Perform static timing analysis using Performance Analyst and view the results.

Glossary

Following are the terms and concepts that you should understand to use this tutorial effectively.

EDIF. EDIF (Electronic Design Interchange Format) is a format used to exchange design data between different electronic computer-aided design systems. It is designed to be written and read by computer programs that are constituent parts of EDA systems or tools. Its syntax has been designed for easy machine parsing and is similar to LISP. The ispLEVER software supports EDIF Version 2.0.0.

HDL. An HDL is a hardware description language, which describes the structure and function of integrated circuits.

netlist. A netlist is an ASCII file describing the logical components in a design and how they are connected. It is basically a list of connectors, a list of instances, and, for each instance, a list of the signals connected to the instance terminals. In addition, the netlist contains attribute information.

static timing analysis. Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup and hold-time requirements, and the maximum frequency.

synthesis. Synthesis is the process of translating a high-level design (RTL) description consisting of state machines, truth tables, Boolean equations, or all three into a process-specific, gate-level logic implementation.

Verilog. Verilog is a language for describing the structure and function of integrated circuits.

VHDL. VHDL (or VHSIC (Very High-Speed Integrated Circuits) Hardware Description Language) is a language for describing the structure and function of integrated circuits.

Recommended References

You can find additional information on the subjects covered by this tutorial from the following recommended sources:

- ◆ *Synplify and Synplify Pro for Lattice User Guide*
- ◆ *Synplify and Synplify Pro for Lattice Reference Manual*
- ◆ ispLEVER online Help
- ◆ Data sheets, technical notes, and other information on FPGAs on the Lattice Semiconductor Web site at <http://www.latticesemi.com/search/literature.cfm>.