HDL Synthesis Design with Synplify: ispXPGA Flow

Table of Contents

HDL Synthesis Design with Synplify: ispXPGA Flow 2
Task 1: Create a New Project 5
Task 2: Target a Device 7
Task 3: Create a Synplify Project 9
Task 4: Add the VHDL Source Files 12
Task 5: Set Implementation Options 14
Task 6: Import the EDIF File into Your Project 17
Task 7: Pack and Place the Design 20
Task 8: Route the Design 22
Task 9: Perform Static Timing Analysis 25
Summary 28
Glossary 29
Recommended Reference Materials 30

HDL Synthesis Design with Synplify: ispXPGA Flow

This tutorial shows you how to use Synplify from within ispLEVER® to synthesize a VHDL design and generate an EDIF netlist file for a Lattice ispXPGA device.

***Note**: If you want to learn how to use Synplify in standalone mode, or understand more about its advanced features, please see the third-party manuals online by choosing **Help > ispLEVER Documentation Library** from the ispLEVER Project Navigator.*

Learning Objectives

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

- Create a new EDIF project in ispLEVER and target a device.
- Start Synplify from within ispLEVER, synthesize your VHDL design, and generate an EDIF netlist file.
- Import the EDIF file into the ispLEVER, and implement the design using the pack, place, and route processes.
- Set report viewing options and view the reports.
- Run static timing analysis using the 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:

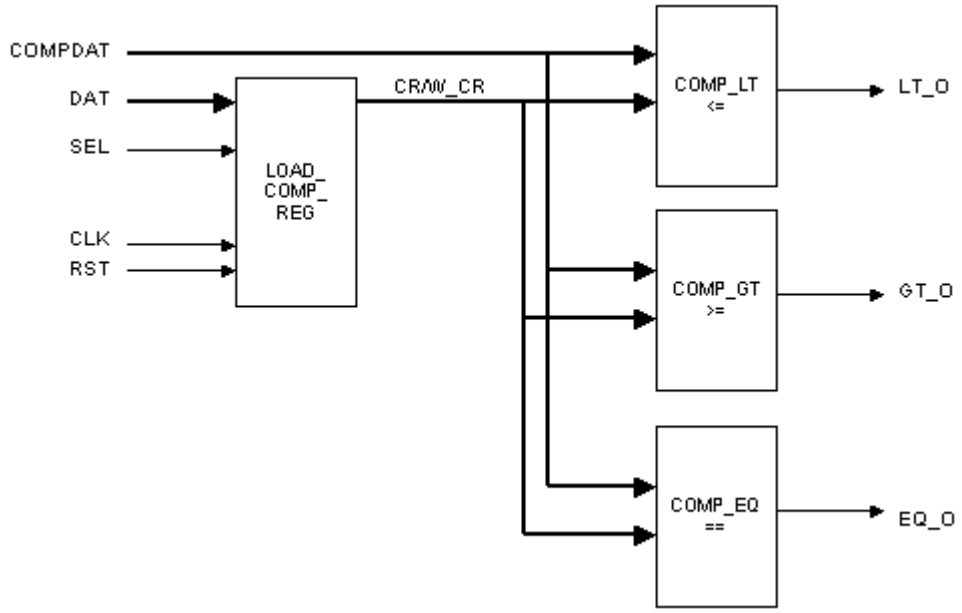
- ispLEVER Starter
- ispLEVER Base
- ispLEVER Advanced
- ispLEVER Advanced System with active Synplicity Synplify license

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

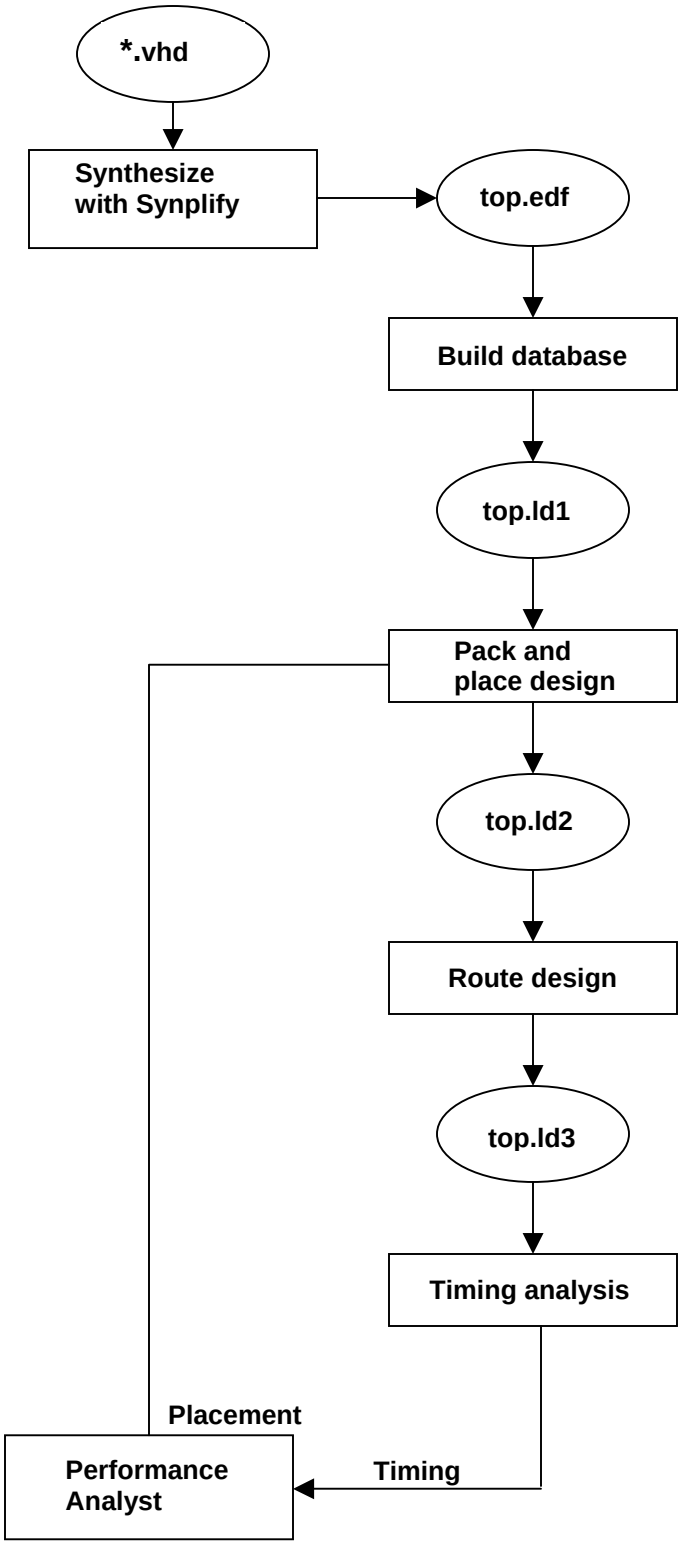
The tutorial design consists of a simple set of equal-to, greater-than, and less-than data comparators, as shown in the following figure:



This tutorial first directs you to create an EDIF project in the Project Navigator, then select the target device in which the design will be implemented. It assumes that functional simulation has already been performed. Next, you start Synplify and open a new Synplify project. After you import the VHDL source files and set the implementation options, the tool synthesizes the design into the target device and generates an EDIF netlist.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

The following figure 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.



Task 1: Create a New Project

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

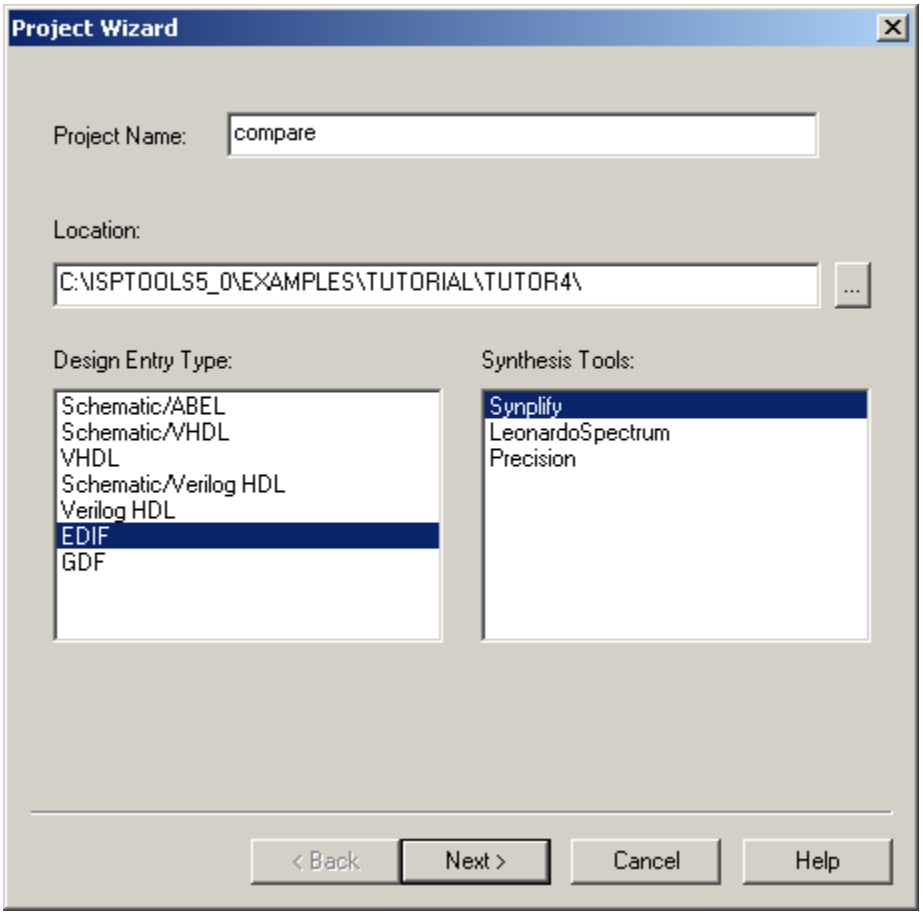
The ispLEVER software saves an initial design file with the `.syn` file extension in the directory that you specify. All project files are copied to or created in this directory. The project type specifies that all design sources will be of this type.

To create a new project:

1. Start the ispLEVER system, if it is not already running.
2. In the Project Navigator, choose **File > New Project** to open the Create New Project dialog box.
3. In the dialog box, do the following:
   - In the Project Name box, type **compare**.
   - In the Location box, change to the following directory: `<install_path>\examples\tutorial\tutor4`.

***Note**: If you want to preserve the original tutorial design files, save the `tutor4` directory to another location on your computer before proceeding.*

- In the Design Entry Type box, select **EDIF**.
- In the Synthesis Tools box, select **Synplify**.
- Click **Next** to activate the Project Wizard – Select Device dialog box.

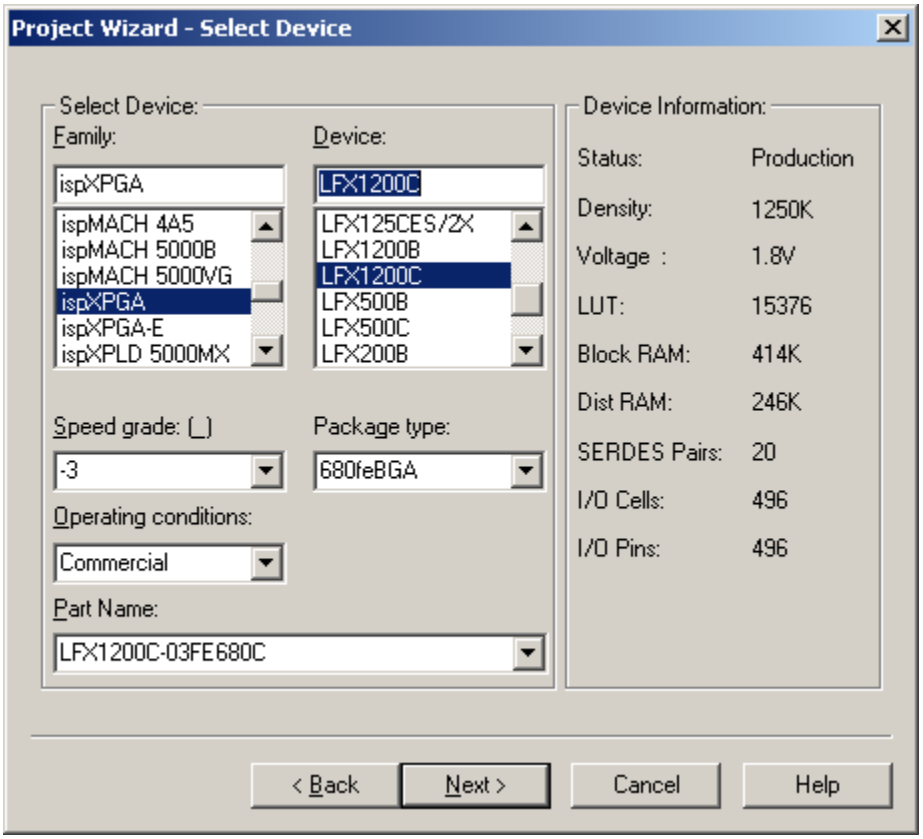


Task 2: Target a Device

In the Project Navigator Sources in Project window, the device icon appears next to the target device for the project. The Project Navigator enables you target a design to a specific Lattice 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:

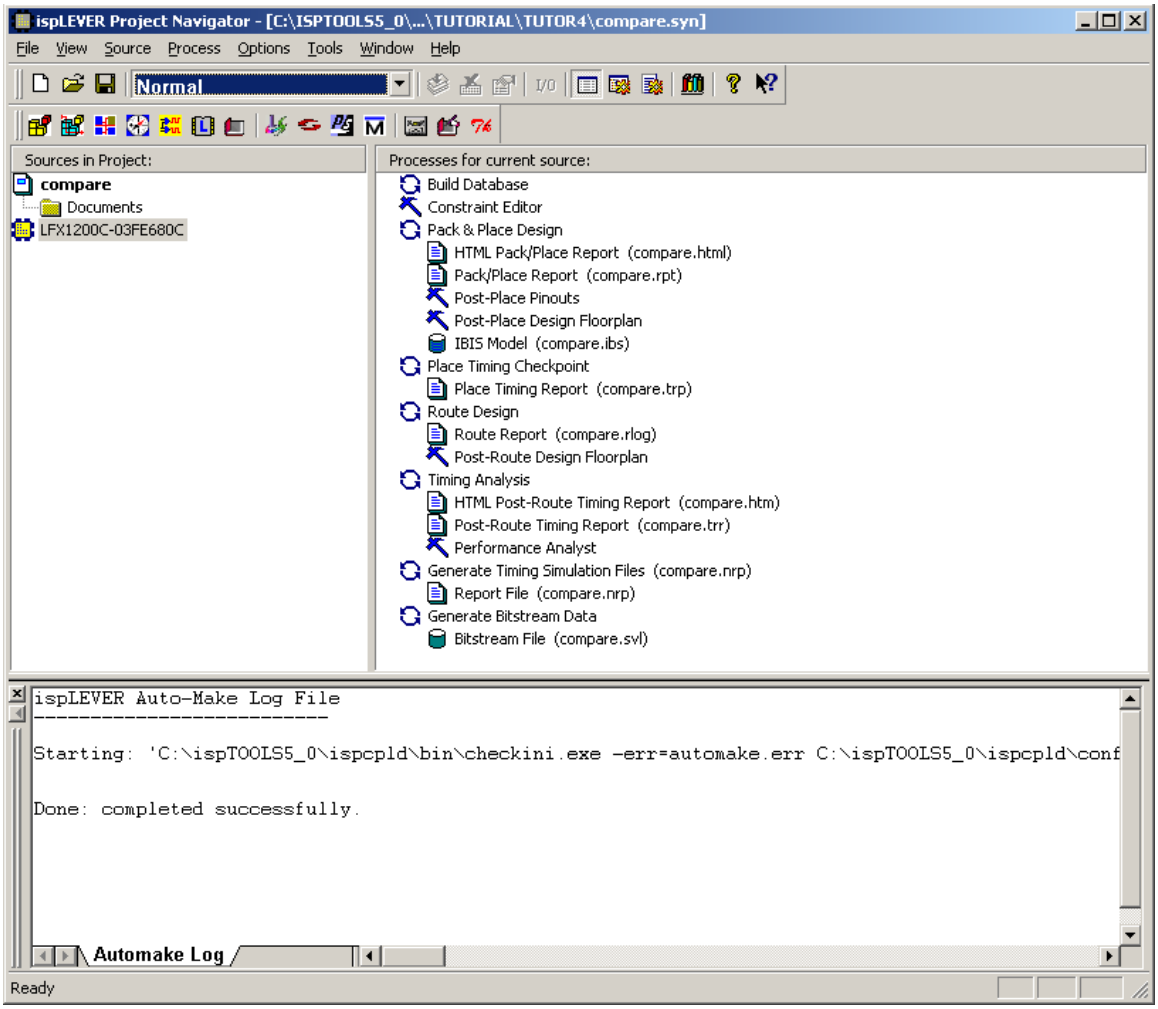
1. In the Project Wizard – Select Device dialog box, do the following:
   - In the Family box, choose **ispXPGA**.
   - In the Device box, choose **LFX1200C**.
   - In the Part Name box, choose **LFX1200C-03FE680C**.
   - Click **Next** to activate the Project Wizard - Add Source dialog box.



2. In the Project Wizard – Add Source dialog box, click **Next**, then click **Finish**.

   Your Project Navigator should look like this:

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



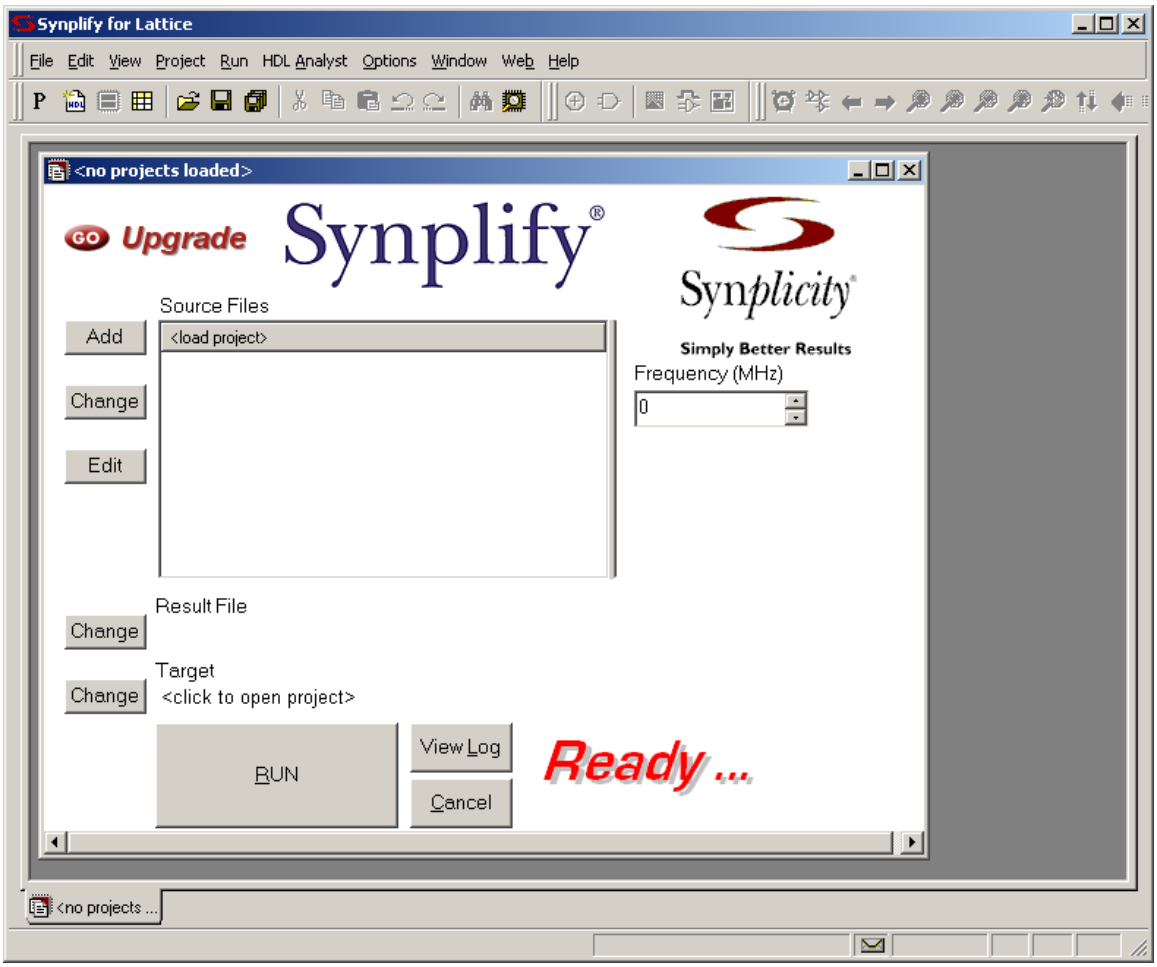
Task 3: Create a Synplify Project

For HDL designs, the ispLEVER software provides two synthesis tools that are integrated into the Project Navigator environment: Synplify and LeonardoSpectrum. You can synthesize your Verilog or VHDL design as a standalone process by choosing the synthesis tool from the Lattice Semiconductor program group in your Start menu, or you can synthesize automatically and seamlessly within the Project Navigator.

Synplify is a logic synthesis tool that starts with a high-level design written in Verilog or VHDL hardware description languages (HDLs). Then Synplify converts the HDL description into small, high-performance design netlists that are optimized for Lattice devices.

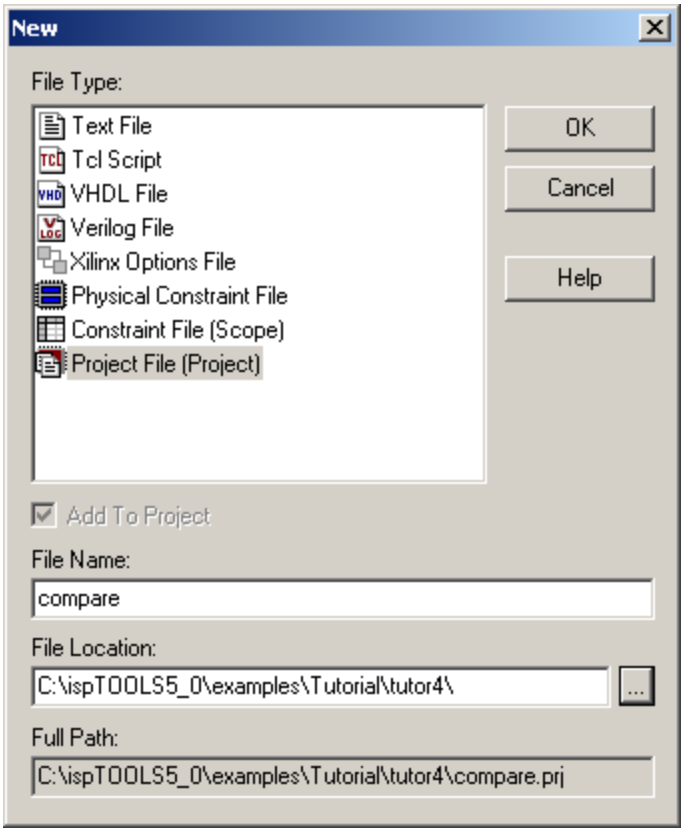
To start Synplify:

1. In the Project Navigator, choose **Tools > Synplify Synthesis** to open the Synplify synthesis tool in the Project view.
2. Click **OK** to close the Tip of the Day.

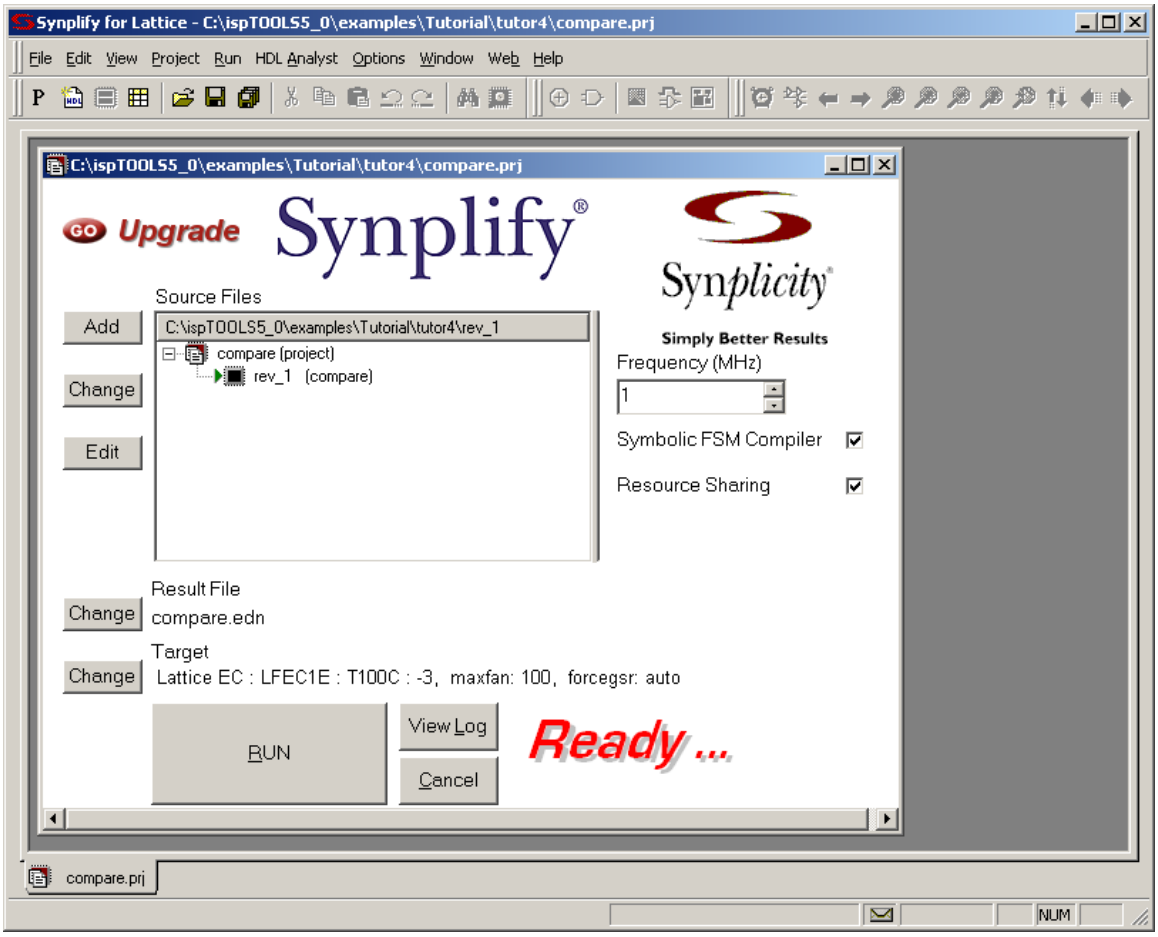


3. Choose **File > New** to open the New dialog box.
4. In the dialog box, do the following:

- In the File Type box, select **Project File**.
- In the File Name box, type **compare**.
- In the File Location box, make sure you are in the **tutor4** folder.
- Click **OK**.



The Synplify screen should look like this:


Synplify for Lattice - C:\ispTOOLS5_0\examples\Tutorial\tutor4\compare.prj
File Edit View Project Run HDL Analyst Options Window Web Help
C:\ispTOOLS5_0\examples\Tutorial\tutor4\compare.prj
Upgrade
Synplify®
Synplicity
Simply Better Results
Source Files
Add
Change
Edit
C:\ispTOOLS5_0\examples\Tutorial\tutor4\rev_1
compare (project)
rev_1 (compare)
Frequency (MHz)
1
Symbolic FSM Compiler
Resource Sharing
Result File
Change
compare.edn
Target
Change
Lattice EC : LFEC1E : T100C : -3, maxfan: 100, forcegsr: auto
RUN
View Log
Cancel
Ready ...
compare.prj
NUM

Task 4: Add the VHDL Source Files

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

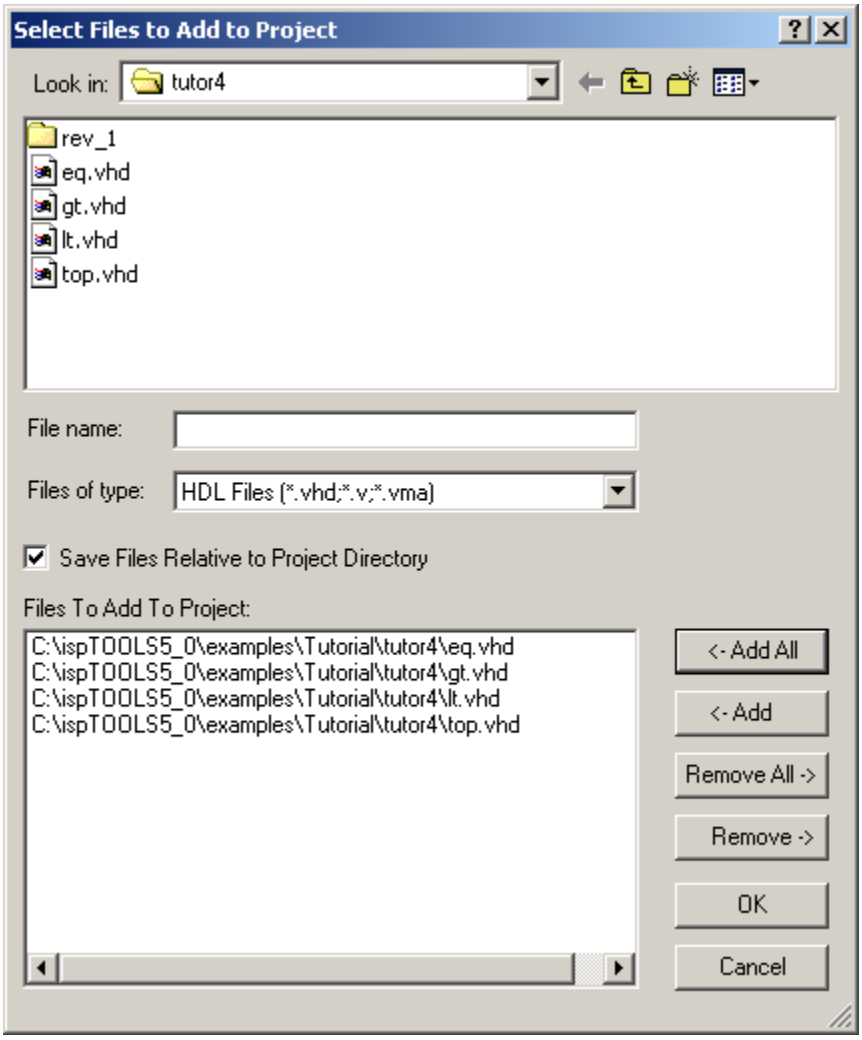
To add source files to the project:

1. On the Synplify main window, click **Add** to open the Select Files to Add to Project dialog box.

   You should see four VHDL (`.vhd`) files.

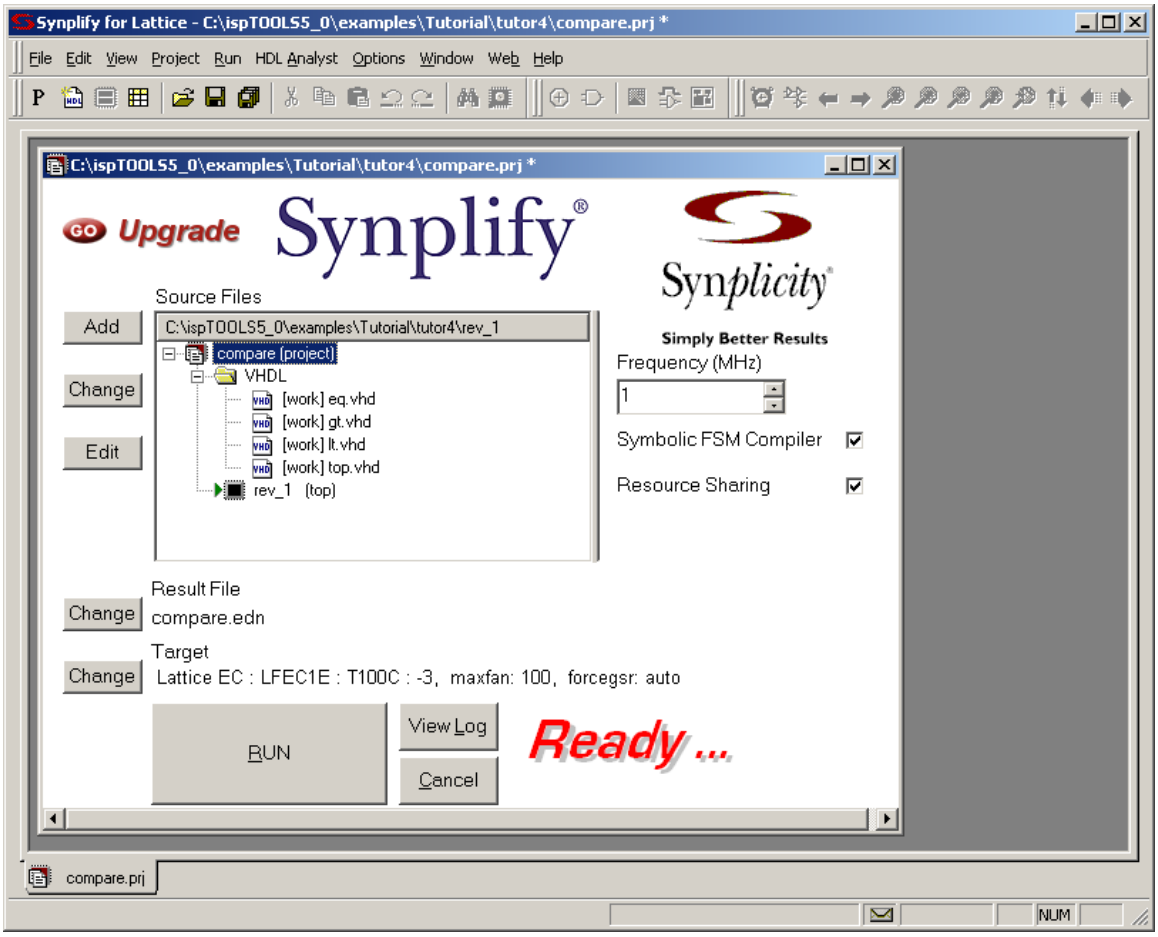
2. Click **Add All** to add all of the VHDL files to the project (shown in the Files To Add To Project field).

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



4. In the Synplify window, click the plus sign (+) in front of the **vhdl** folder to expand the view.

   The Synplify screen should look like this:



***Note**: `top.vhd` must be at the bottom of the list of VHDL files for the project in the Synplicity GUI because Synplicity only compiles the module at the bottom of the list. For example, if `eq.vhd` were at the bottom of the list, the EDIF file would only contain the logic for `eq.vhd` because it does not call any other modules. If `top.vhd` is not at the bottom of the list, click and drag the file to the bottom.*

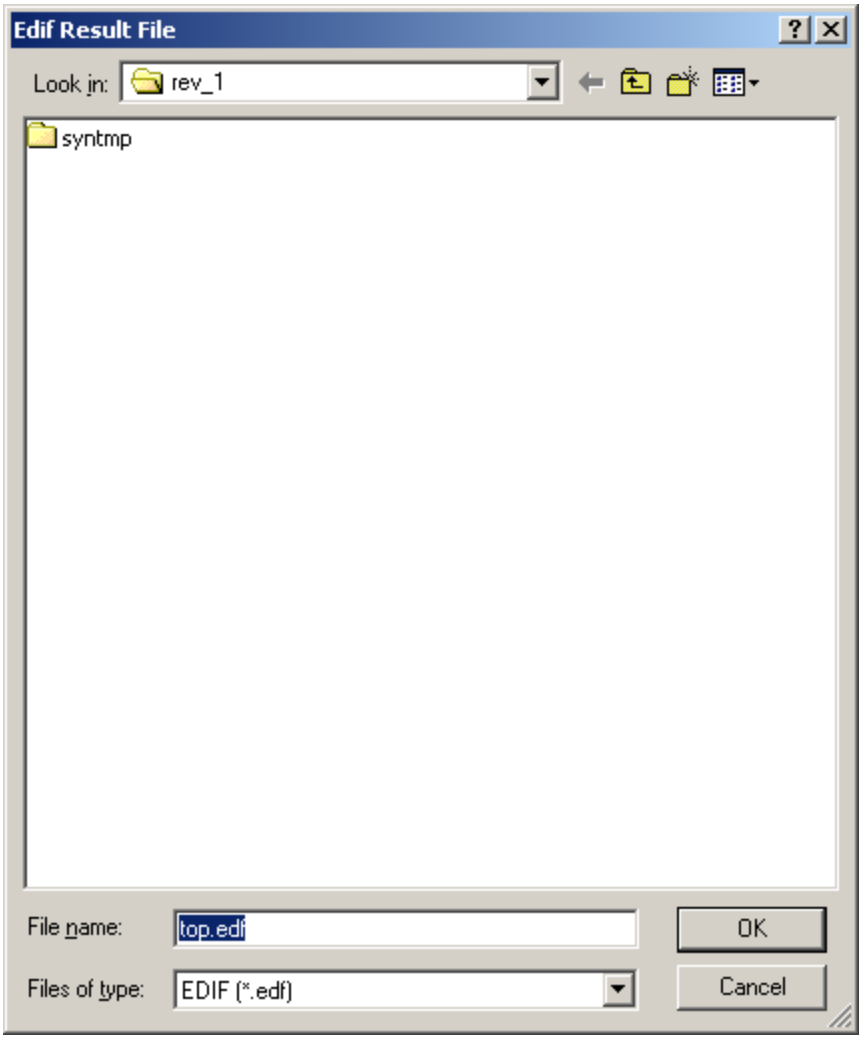
Task 5: Set Implementation Options

Synplify has probably set your implementation options correctly. However, it is a good practice to check them, especially if you had to change the location of `top.vhd` in the previous step.

To set the implementation options:

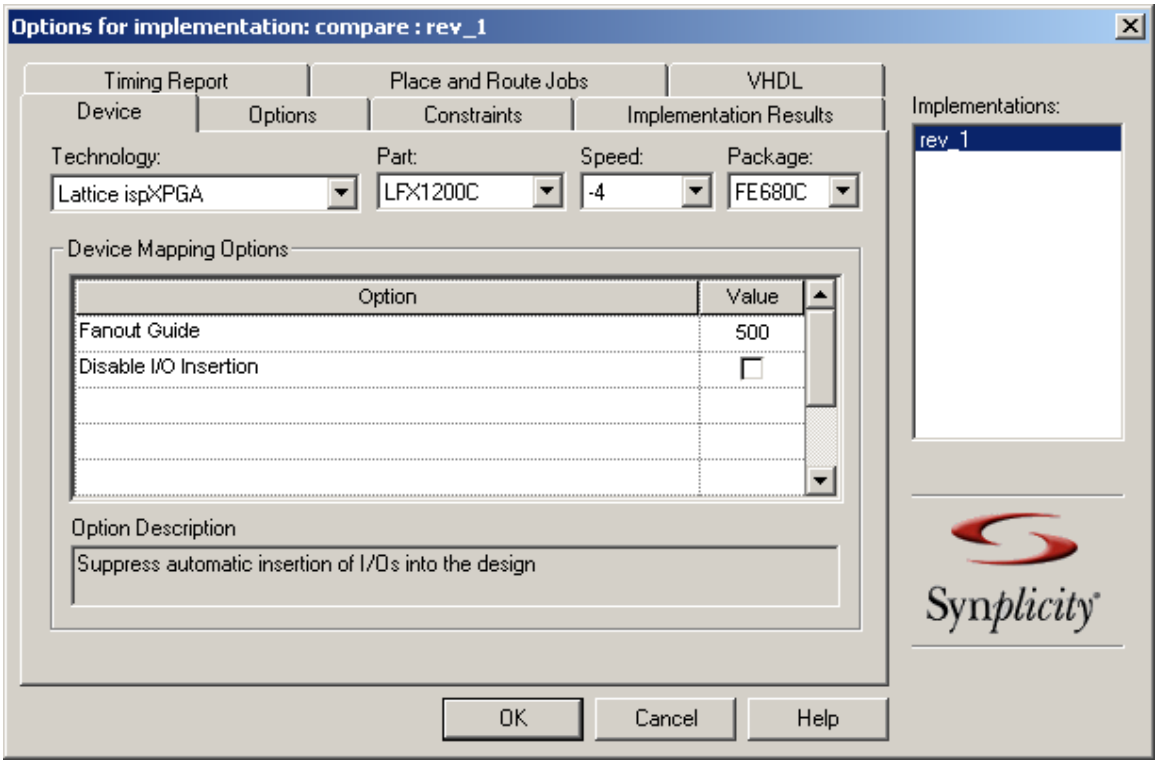
1. Click **Change** (**Result File)** to open the EDIF Result File dialog box.

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



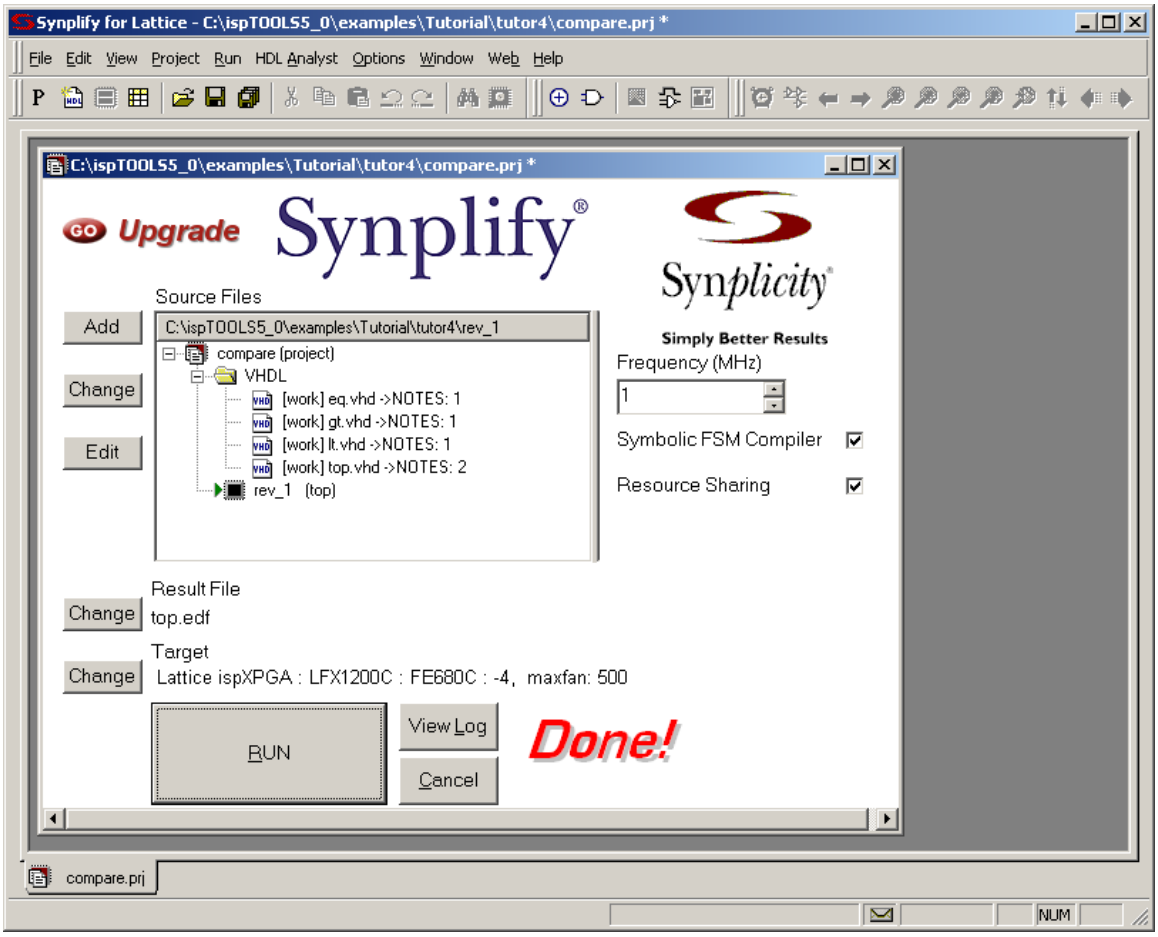
2. Click **Cancel** to close the dialog box.
3. Click **Change** (**Target)** to open the Options for Implementation dialog box.
4. In the dialog box, do the following:
   - In the Technology box, select **Lattice ispXPGA**.
   - In the Part box, select **LFX1200C**.

- Under Speed, select -4.
- In the Package box, accept the default (FE680C).
- Click **OK**.



5. Click **Run** to start the synthesis process.

   Synplify synthesizes the VHDL design and creates an EDIF file, as well as several other files, and displays them in the window. If you like, you can double-click the different files and view them. When you are finished, close the files.



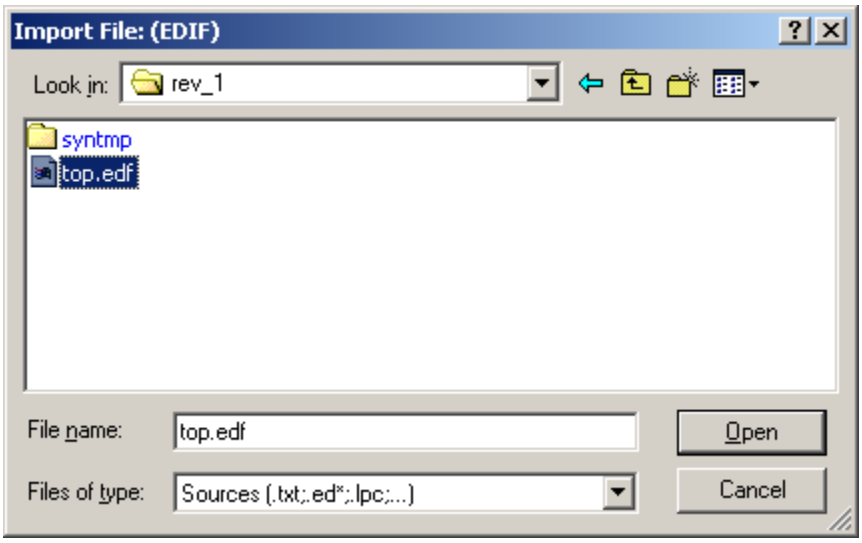
6. Choose **File > Save** to save the Synplify project.
7. Choose **File > Exit** to close Synplify.

Task 6: Import the EDIF File into Your Project

Now you are ready to import the EDIF file into ispLEVER project. You can import EDIF 2 0 0 netlists from third-party synthesis tools, such as Synplicity Synplify, into ispLEVER.

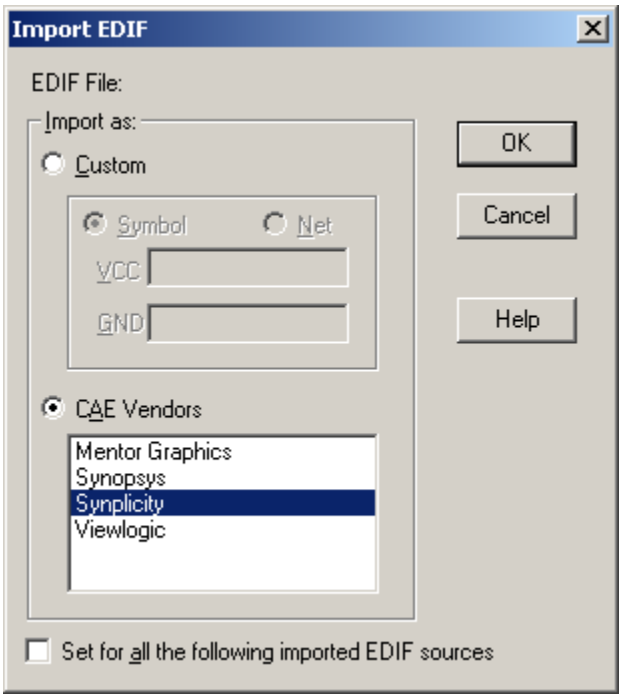
To import an EDIF netlist into your project:

1. In the ispLEVER Project Navigator, choose **Source** > **Import** to open the Import File dialog box.
2. Go to the **rev_1** folder within your project, select **`top.edf`**, and click **Open** to open the Import EDIF dialog box.

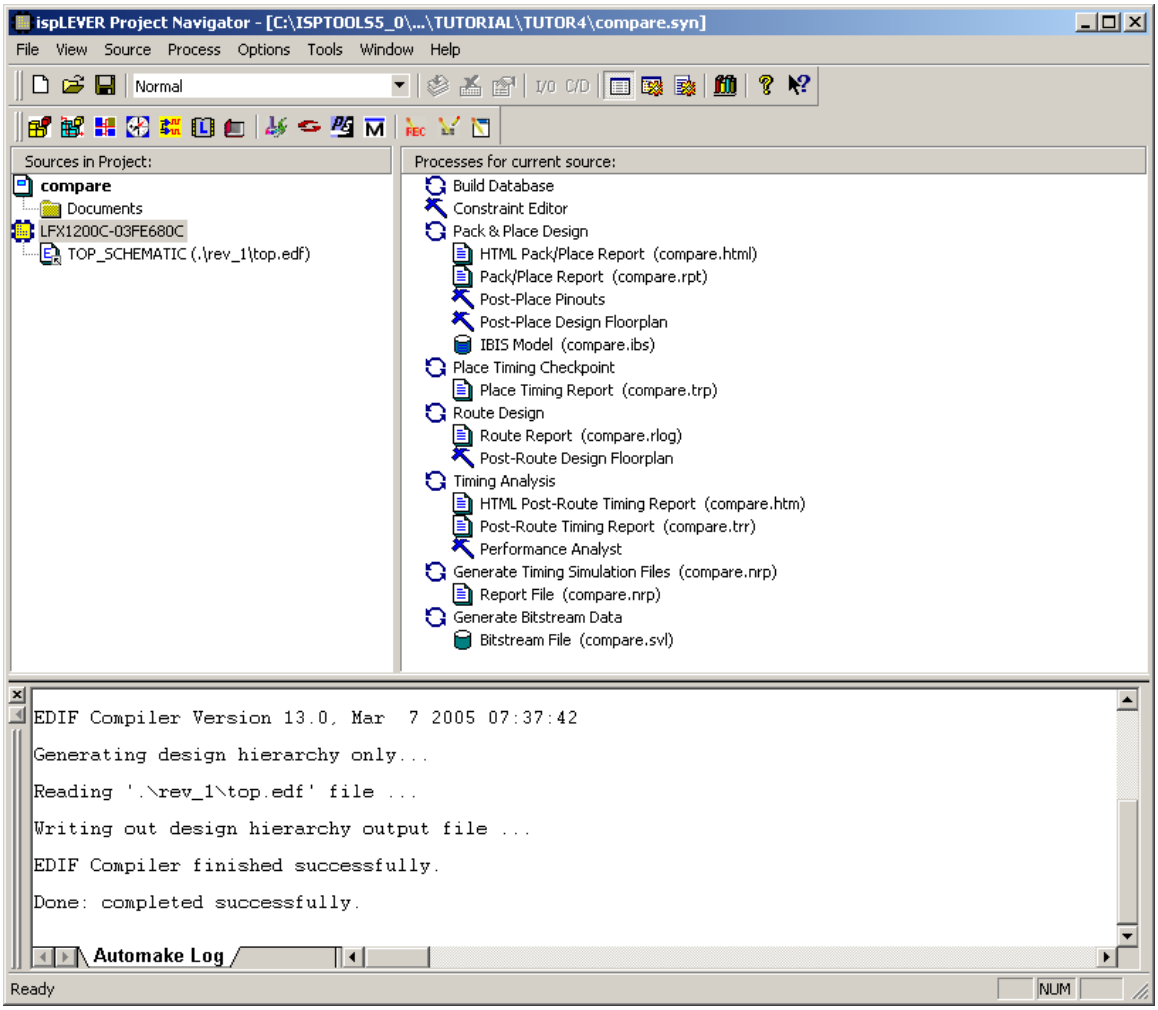


The software adds the selected EDIF file (`top.edf`) to the project sources.

3. In the Import EDIF dialog box under CAE Vendors, select **Synplicity**, and click **OK**.



The software adds the selected EDIF file (`top.edf`) to the project sources.



***Note**: After you import an EDIF file into the ispLEVER project, it is always linked to the 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.*

Task 7: Pack and Place the Design

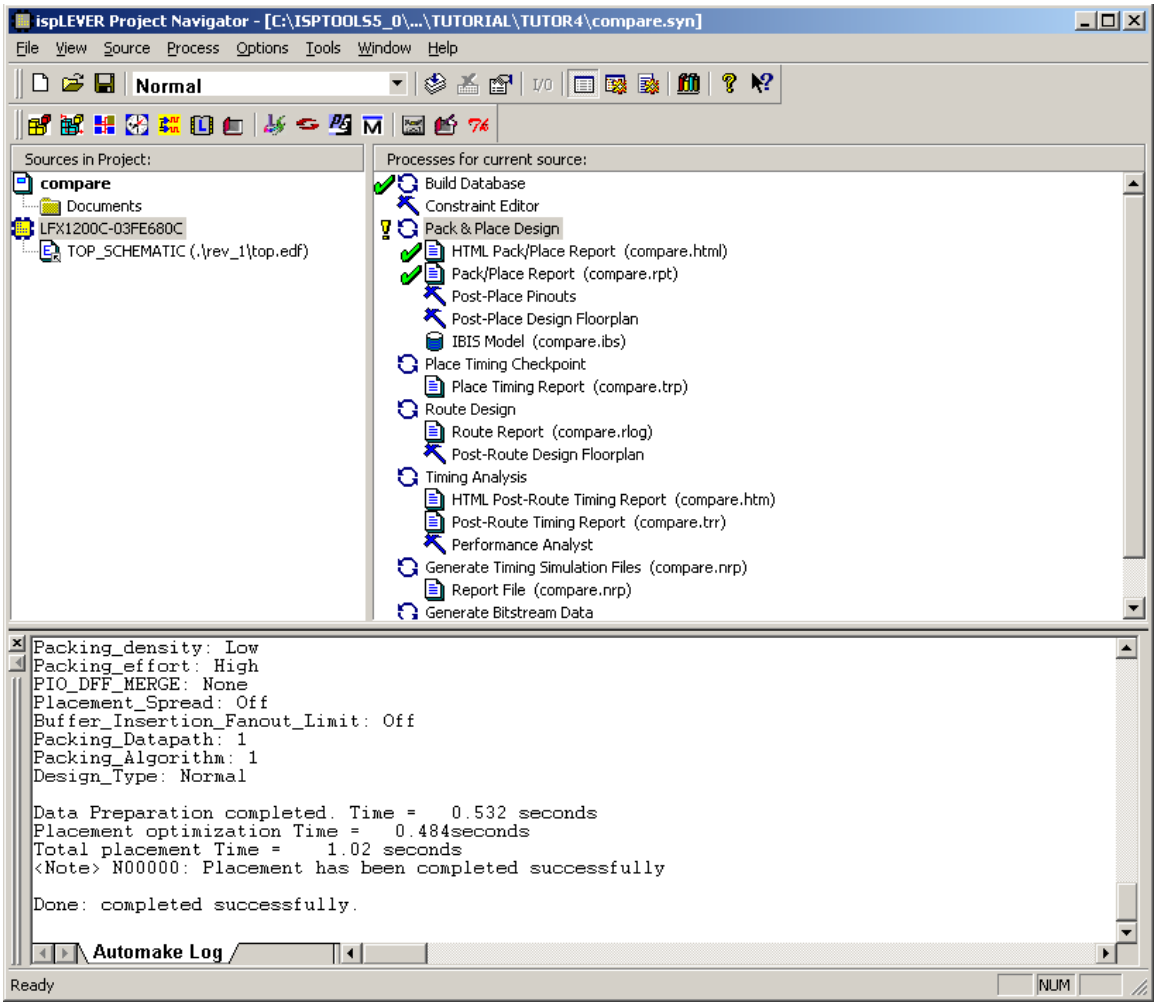
The ispLEVER software has a single user interface with all options preset to deliver the highest possible push-button performance for most devices. When you double-click a process, all the processes prior to that process run automatically. Therefore, all you have to do is double-click the final process.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

After an initial internal database is generated, the Pack & Place Design process packs the design instances into programmable functional units (PFUs) and places them on the ispXPGA device. It generates a file that you can submit to the Route Design process.

To place and route the design and view the reports:

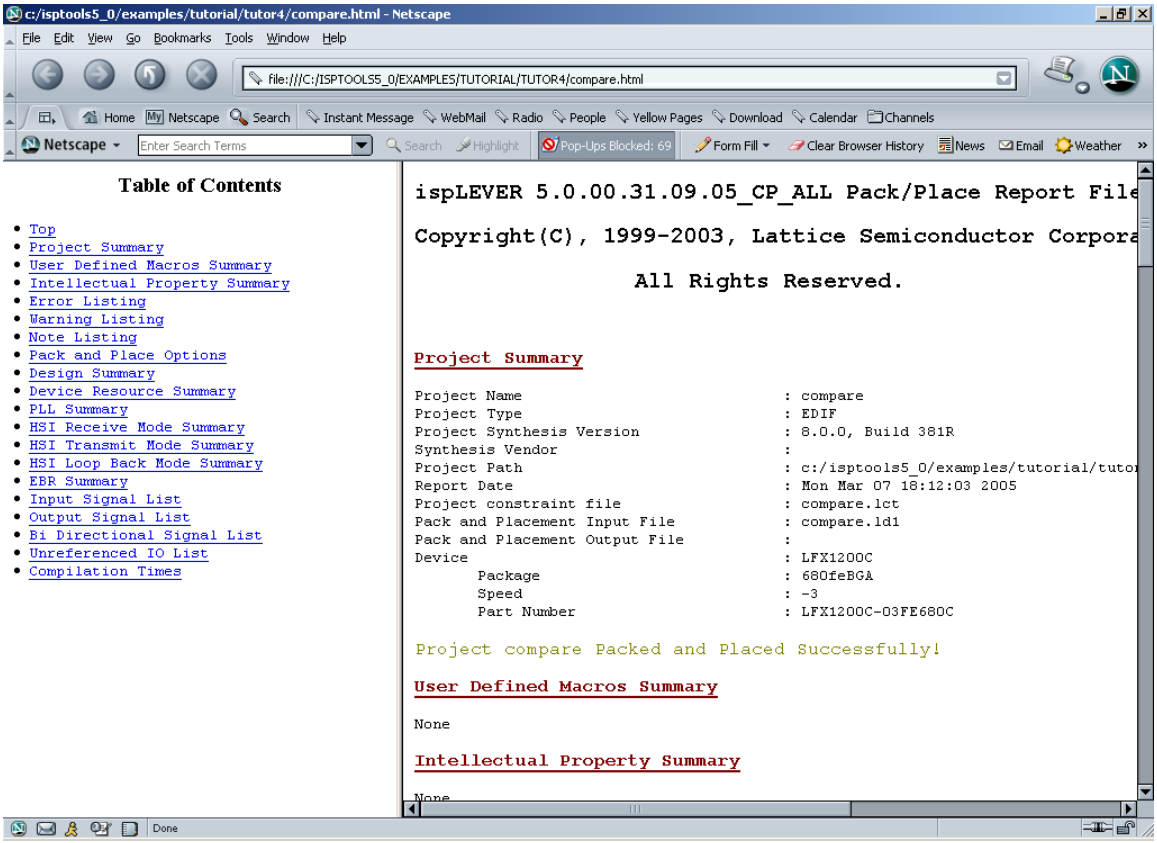
1. With the target device selected in the Sources in Project window, double-click **Pack & Place Design** in the Processes for Current Source window.

***Note**: You can ignore the warnings.*



2. Double-click **HTML Pack/Place Report** to open the report in your browser.

This report gives details about the design's packing and placement before routing and includes such sections as summary reports, signal lists, fanin, fanout, removed and added logic, and compilation time.



3. View the contents and close the report.

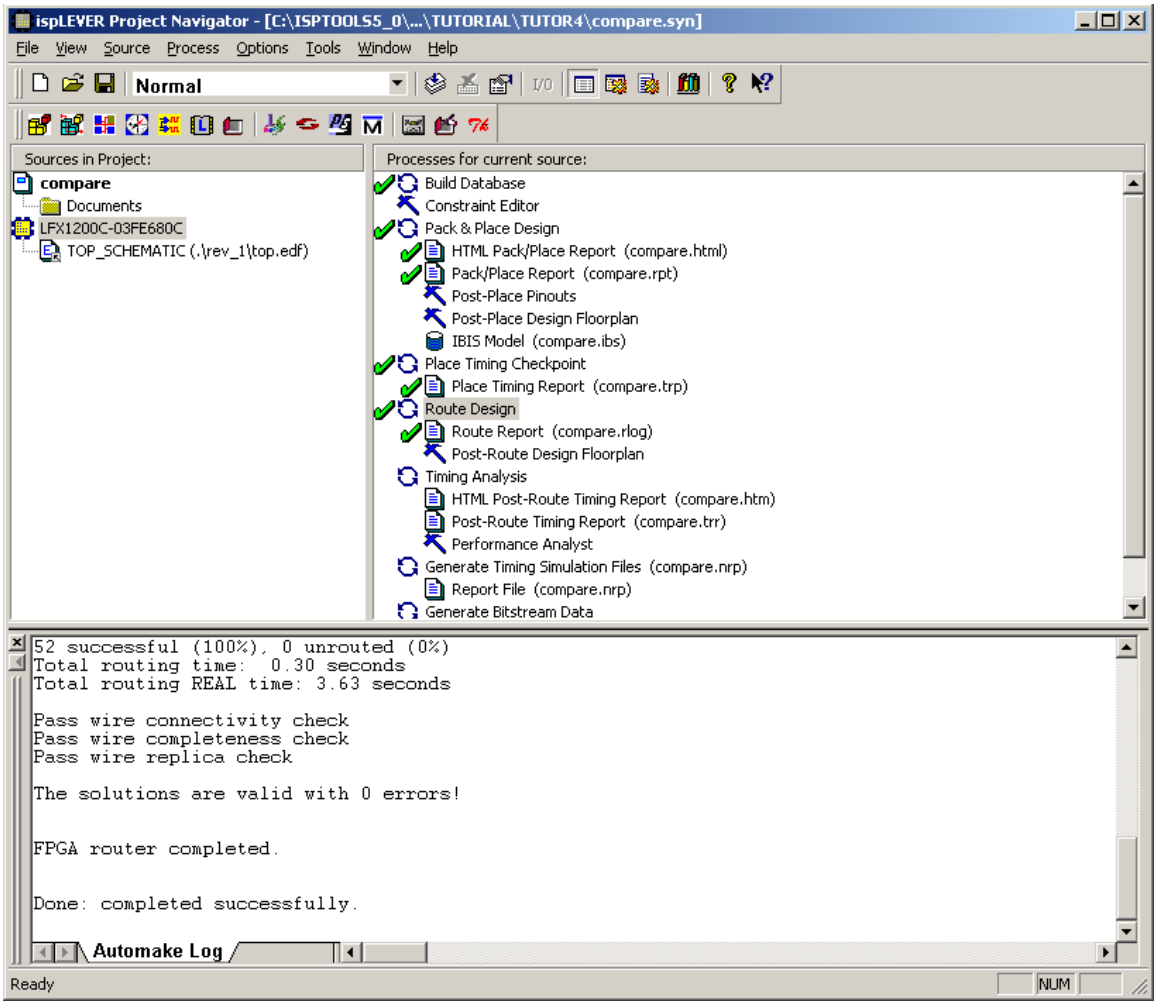
Task 8: Route the Design

The ispLEVER software routes the design in the device after placement. The routing algorithm takes full advantage of the Lattice ispXPGA architecture to achieve maximum performance. It uses congestion-driven routing to achieve a fit with minimal congestion, and it uses timing-driven routing to achieve maximum performance.

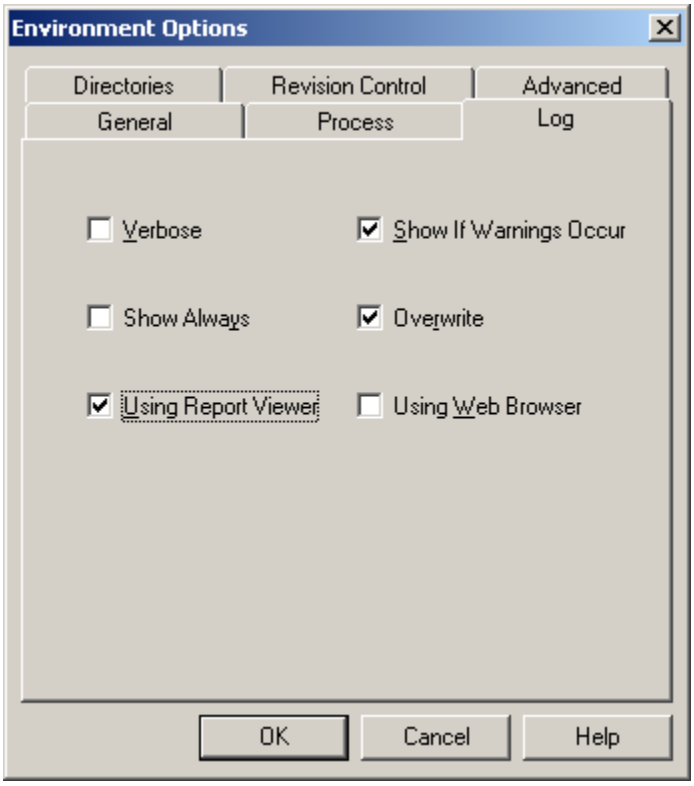
To route the design and view the route report:

1. Double-click the **Route Design** process.

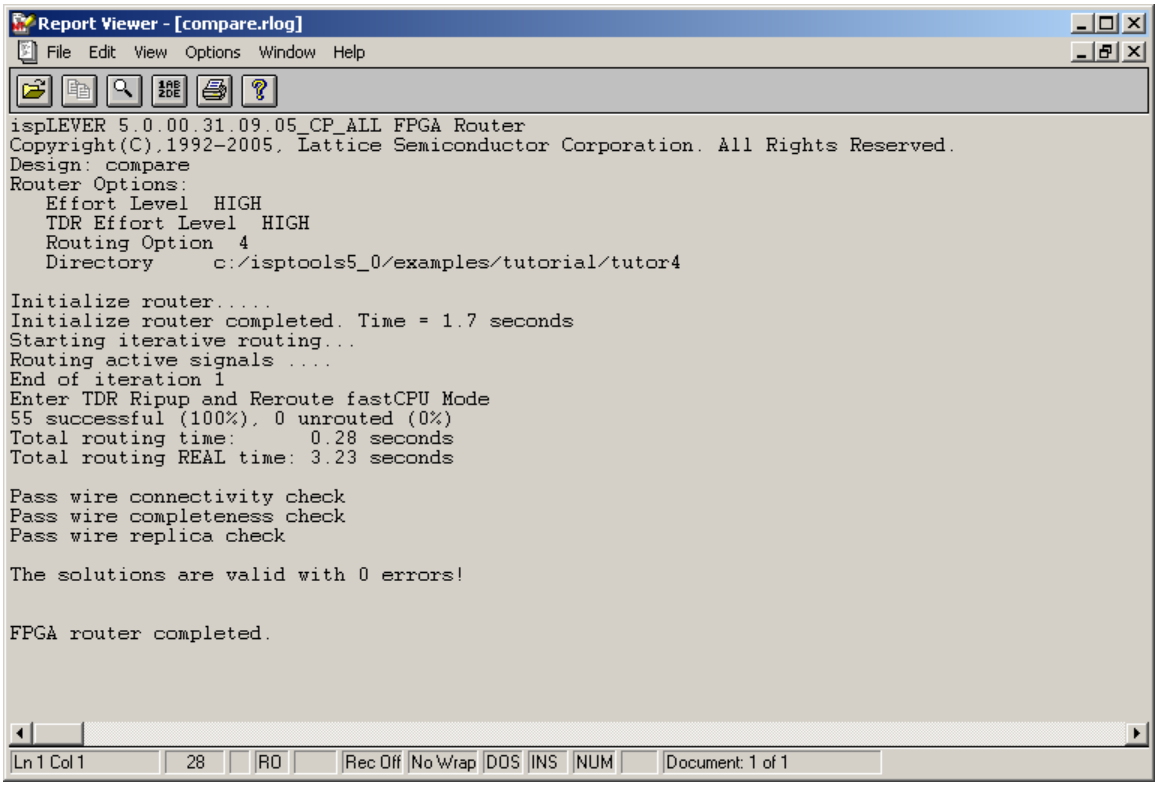
   The ispLEVER software successfully routes the design in the specified device and generates an .ld3 file. When the process is finished, a green check mark appears next to the process.



2. Choose **Options** > **Environment** to open the Environment Options dialog box.
3. Select the **Log** tab.
4. Make sure **Using Report Viewer** is selected to enable you to open report files in the Report Viewer rather than in the output panel.
5. Click **OK** to close the dialog box.



6. Double-click **Route Report** to open the route report, which shows how a design was routed in the target device and informs you of the success or failure of the routing.



7. Close the Report Viewer.

Task 9: Perform Static Timing Analysis

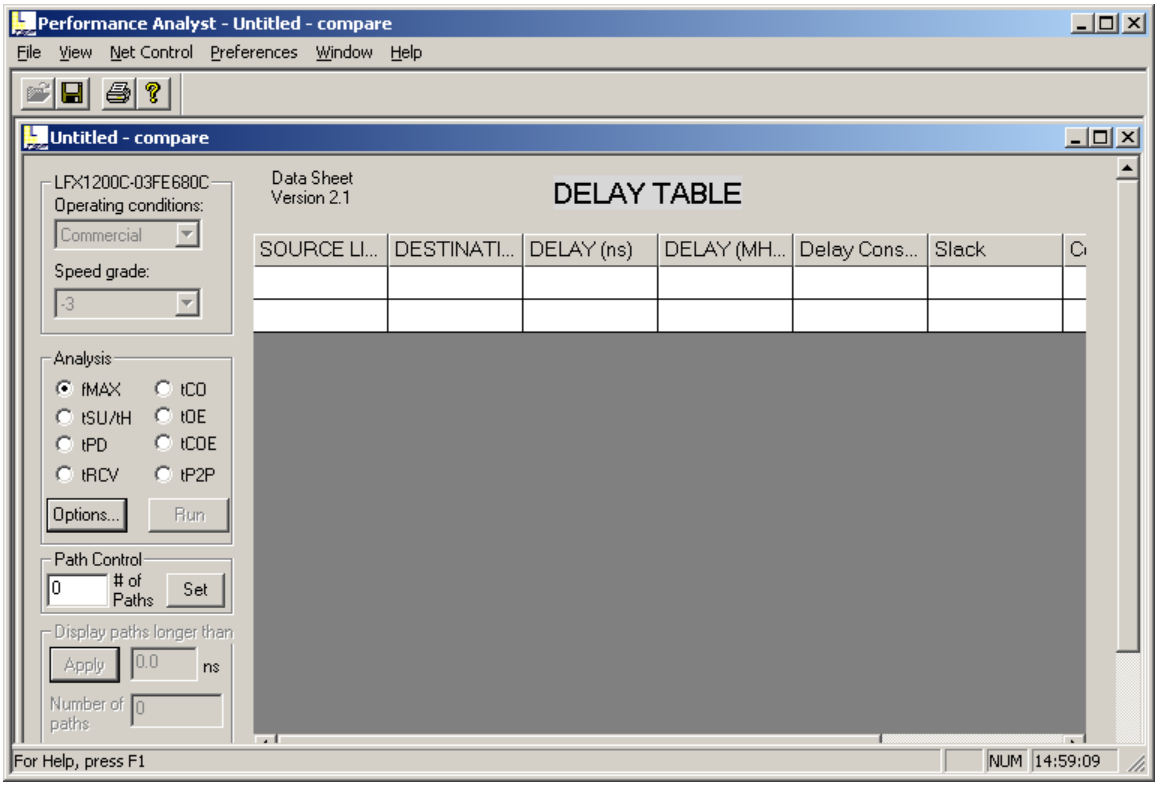
Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup and hold-time requirements, and the maximum frequency.

The 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 graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. 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 the Performance Analyst.



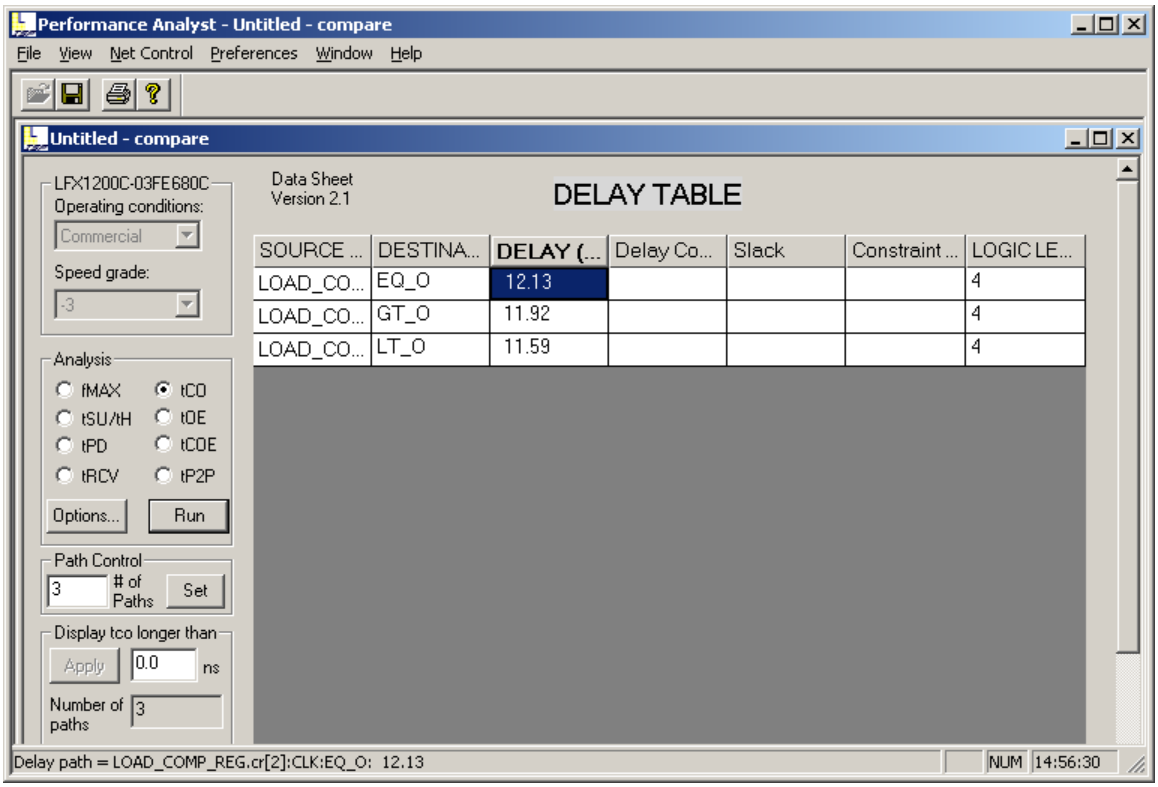
The Performance Analyst performs eight distinct analyses. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining seven types are external pin-to-pin delay analysis. Timing threshold filters, source

and destination filters, and path filters can be used to independently fine-tune each analysis.

3. In the Analysis field, select **tCO** and then click **Run**.

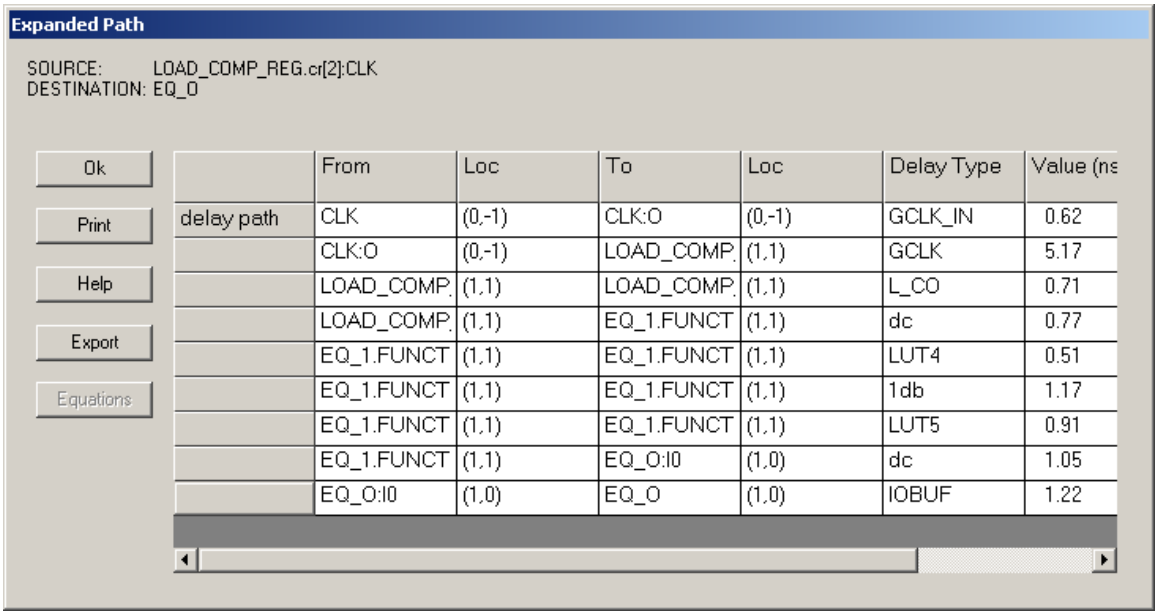
   The tCO path trace analysis reports clock-to-out delay starting from the primary input, going through the clock of flip-flops or gate of latches, and ending at the primary output. In this case, it is 12.13 ns.

***Note**: Your timing results may differ slightly.*



4. Click the highlighted cell (12.13) in the spreadsheet window to open the 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 (*value* ns), and the cumulative delay of all the signals.



	From	Loc	To	Loc	Delay Type	Value (ns
delay path	CLK	(0,-1)	CLK:O	(0,-1)	GCLK_IN	0.62
	CLK:O	(0,-1)	LOAD_COMP.	(1,1)	GCLK	5.17
	LOAD_COMP.	(1,1)	LOAD_COMP.	(1,1)	L_CO	0.71
	LOAD_COMP.	(1,1)	EQ_1.FUNCT	(1,1)	dc	0.77
	EQ_1.FUNCT	(1,1)	EQ_1.FUNCT	(1,1)	LUT4	0.51
	EQ_1.FUNCT	(1,1)	EQ_1.FUNCT	(1,1)	1db	1.17
	EQ_1.FUNCT	(1,1)	EQ_1.FUNCT	(1,1)	LUT5	0.91
	EQ_1.FUNCT	(1,1)	EQ_O:I0	(1,0)	dc	1.05
	EQ_O:I0	(1,0)	EQ_O	(1,0)	IOBUF	1.22

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

Summary

You have completed the HDL Synthesis Design with Synplify tutorial. In this tutorial, you have learned how to do the following:

- Use ispLEVER to create a new EDIF project and target a device.
- Start Synplify from within ispLEVER, synthesize your VHDL design, and generate an EDIF netlist file.
- Import the EDIF file into ispLEVER.
- Implement the design using the pack, place, and route processes.
- Set report viewing options and view the reports.
- Perform static timing analysis using the Performance Analyst and view the results.

Glossary

Following are the 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.

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 Reference Materials

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

- Synplify for Lattice User Guide
- Synplify for Lattice Reference Manual
- Lattice ispLEVER online help:
  - How To guides
  - Process Flows > ispXPGA Flows
- Data sheets, technical notes, and other information on ispXPGAs on the Lattice Web site at http://www.latticesemi.com/search/literature.cfm. Click on **FPGA > ixpXPGA**.

