HDL Synthesis Design with Synplify: CPLD Flow Tutorial

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

October 2005

Copyright

Trademarks

Disclaimers

Contents

HDL Synthesis Design with Synplify Tutorial: CPLD Flow 1
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: Create a New Project **3**
Task 2: Target a Device **5**
Task 3: Create a Synplify Project **6**
Task 4: Add the VHDL Source Files **9**
Task 5: Set Implementation Options **11**
Task 6: Import the EDIF File into Your Project **14**
Task 7: Fit the Design and View the Report **15**
Task 8: Perform Static Timing Analysis **17**
Summary **20**
Glossary **21**
Recommended Reference Materials **21**

1

HDL Synthesis Design with Synplify Tutorial: CPLD Flow

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

Note

If you want to learn how to use Synplify in standalone mode or understand more about its advanced features, see the *Synplify for Lattice User Guide* and the *Synplify for Lattice Reference Manual* in the online help

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 system, fit the design, generate a JEDEC file, and view the Fitter report.
- 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 this tutorial:

- ispLEVER Starter
- ispLEVER

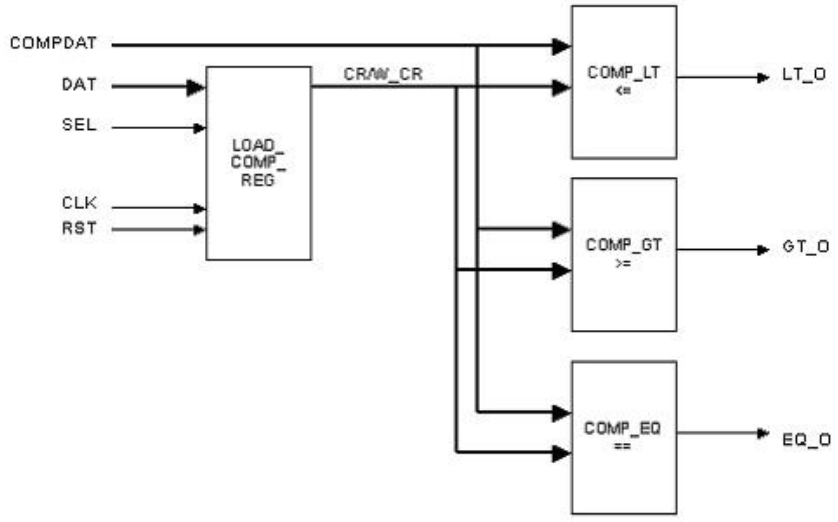
Accessing Online Help

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 consists of a simple set of equal-to, greater-than, and less-than data comparators, as shown in Figure 1:

Figure 1: Tutorial Design

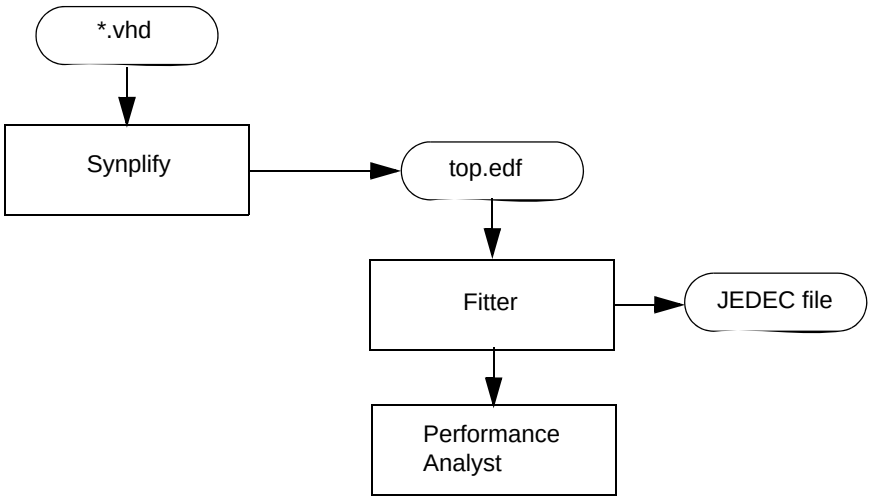


This tutorial first directs you to create an EDIF project in the Project Navigator, then select the target device in which the design will be implemented. It assumes that functional simulation has already been performed. Next, you start Synplify and open a new Synplify project. After you import the VHDL 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 fit the design. Finally, you perform a static timing analysis and examine the results.

About the Tutorial Data Flow

Figure 2 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 2: Tutorial Design Flow



Task 1: Create a New Project

To begin a new project in the Project Navigator, you must create a project directory. Then 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 Project Wizard dialog box.

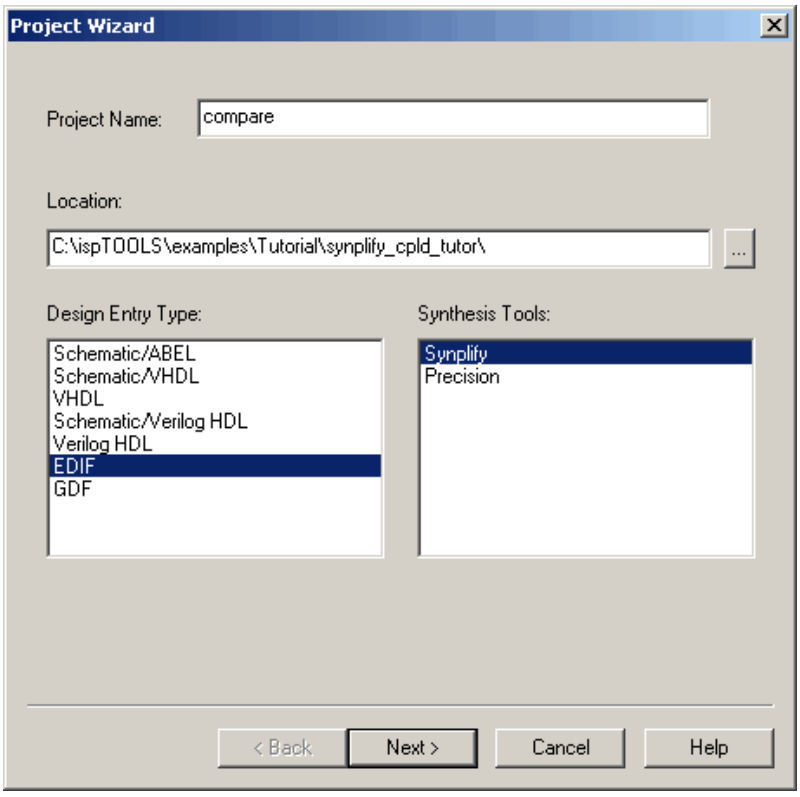
3. In the dialog box, do the following:
   a. In the Project Name box, type **compare**.
   b. In the Location box, change to the following directory:
      `<install_path>\examples\tutorial\`
      `synplify_cpld_tutor`

Note

If you want to preserve the original tutorial design files, save the `synplify_cpld_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. Click **Next** to activate the Project Wizard – Select Device dialog box, shown in Figure 3.

Figure 3: Project Wizard Dialog Box



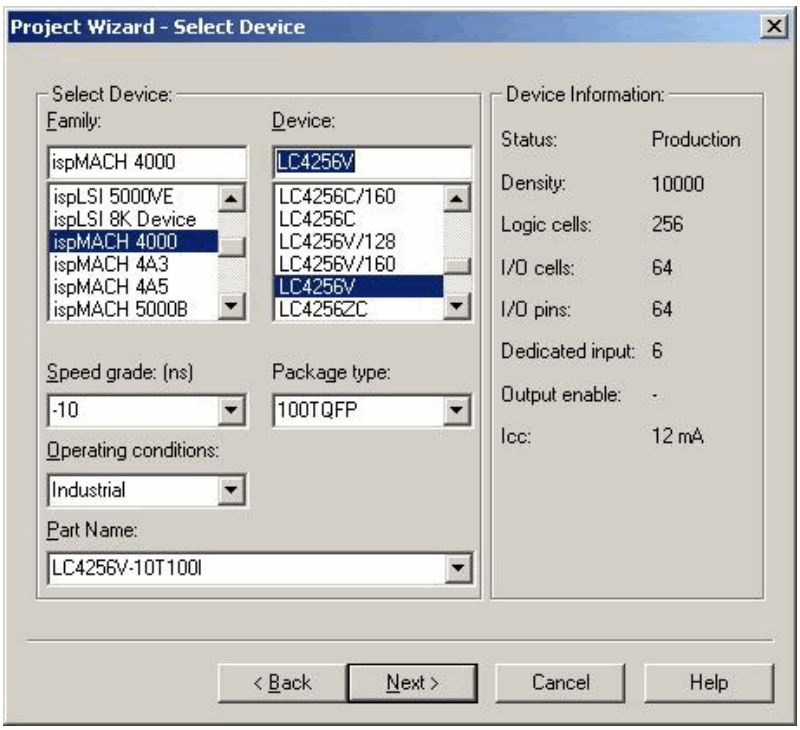
Task 2: Target a Device

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 to target a design to a specific Lattice device at any time during the design process.

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:
   a. In the Family box, choose **ispMACH 4000**.
   b. In the Device box, enter **LC4256V**.
   c. Accept the default settings, and click **Next** to activate the Project Wizard - Add Source dialog box, shown in Figure 4.

Figure 4: Project Wizard - Select Device Dialog Box



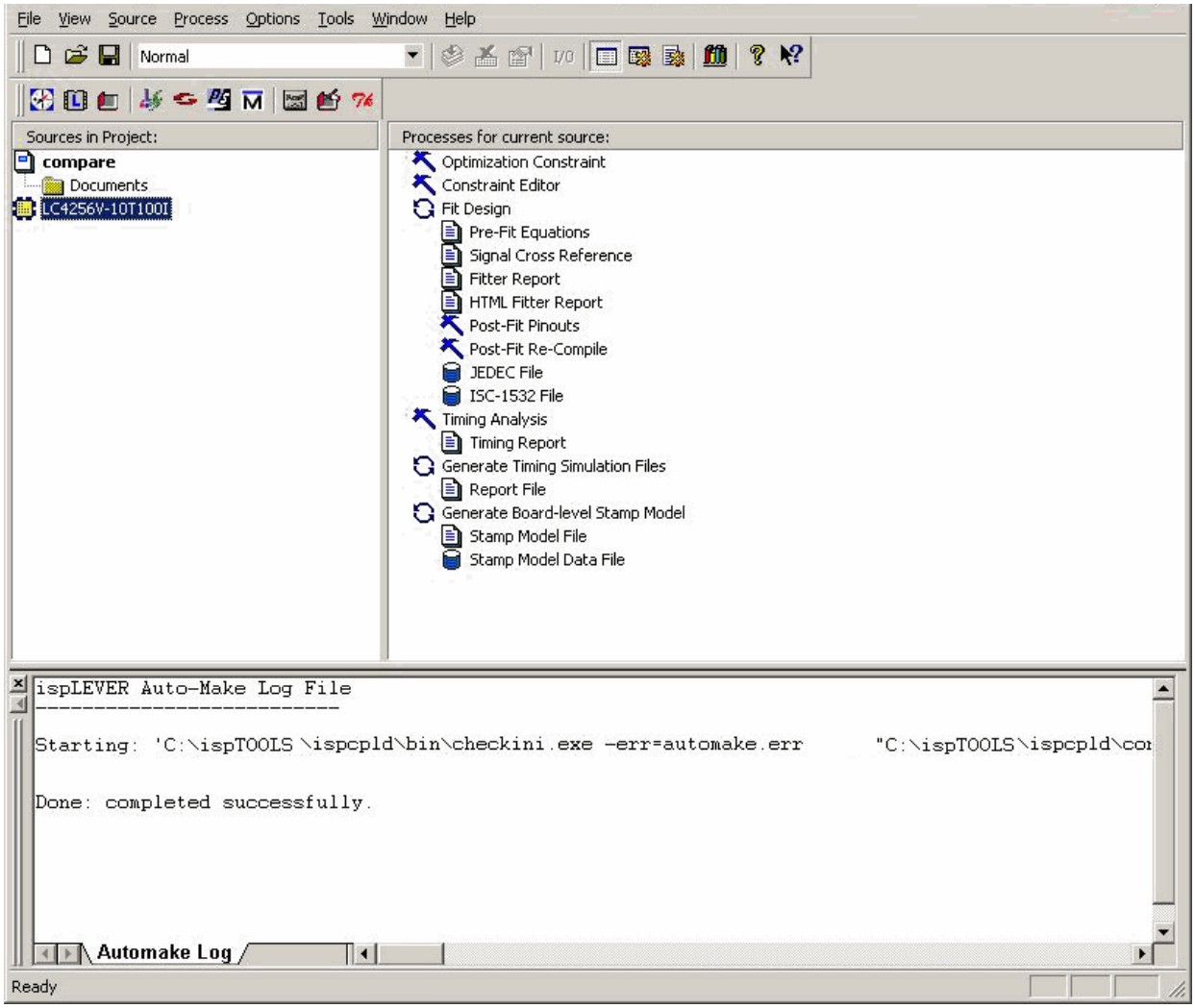
2. In the Project Wizard – Add Source dialog box, click **Next**, then **Finish**.

Your Project Navigator should look like Figure 5.

Note

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

Figure 5: Project Navigator Window Showing New Project



Task 3: Create a Synplify Project

For HDL designs, the ispLEVER software provides two synthesis tools that are integrated into the Project Navigator environment: Synplicity's Synplify and Mentor Graphics' Precision RTL Synthesis. 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.

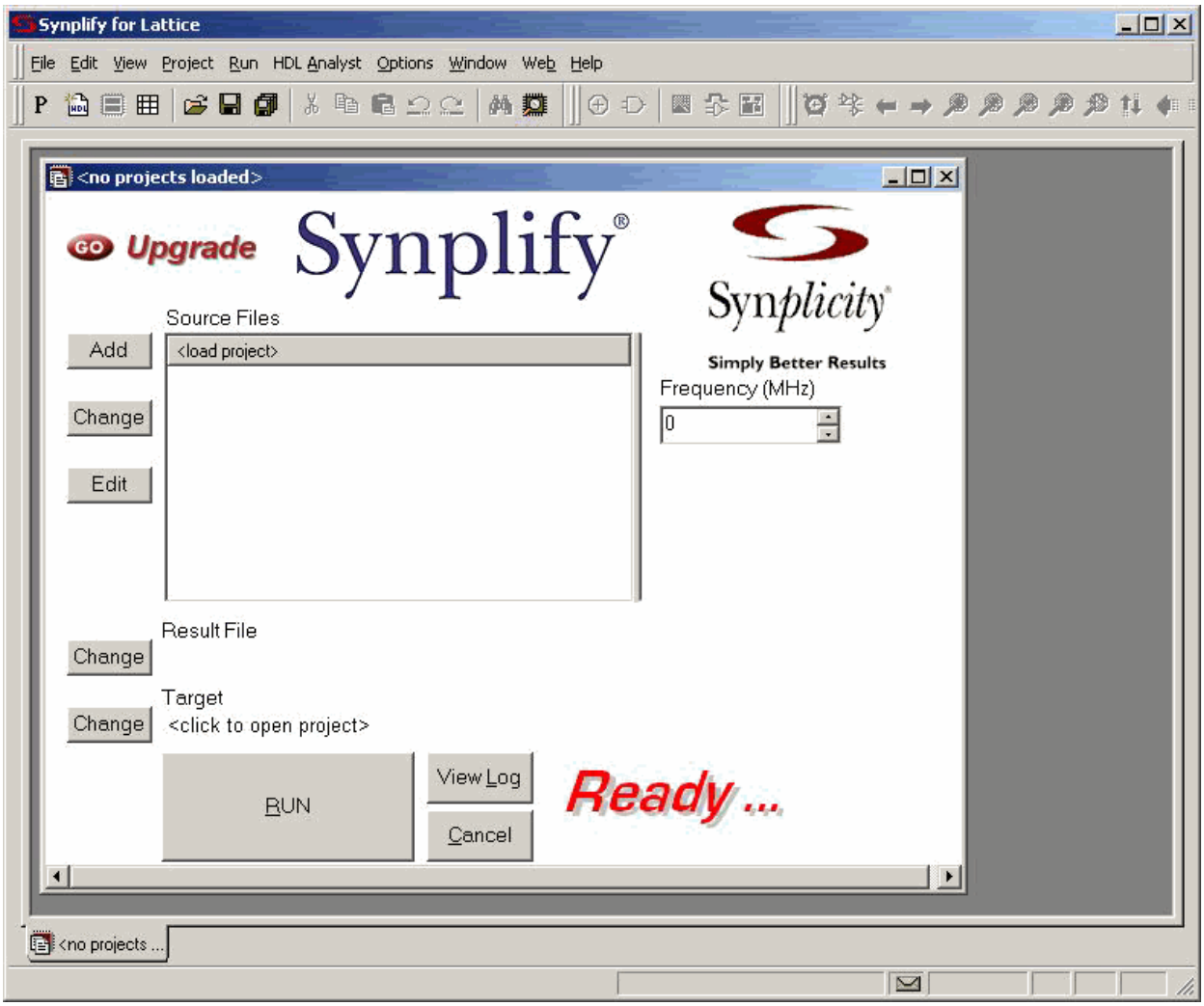
Synplify is a logic synthesis tool that starts with a high-level design written in Verilog or VHDL hardware description languages (HDLs). Then Synplify converts the HDL description into small, high-performance design netlists that are optimized for Lattice devices.

To start Synplify:

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

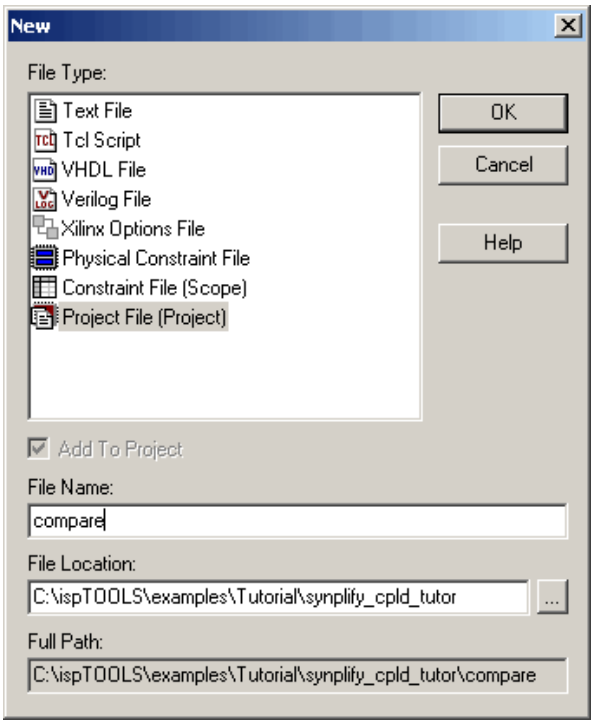
   The Synplify window is shown in Figure 6.

Figure 6: Synplify Window



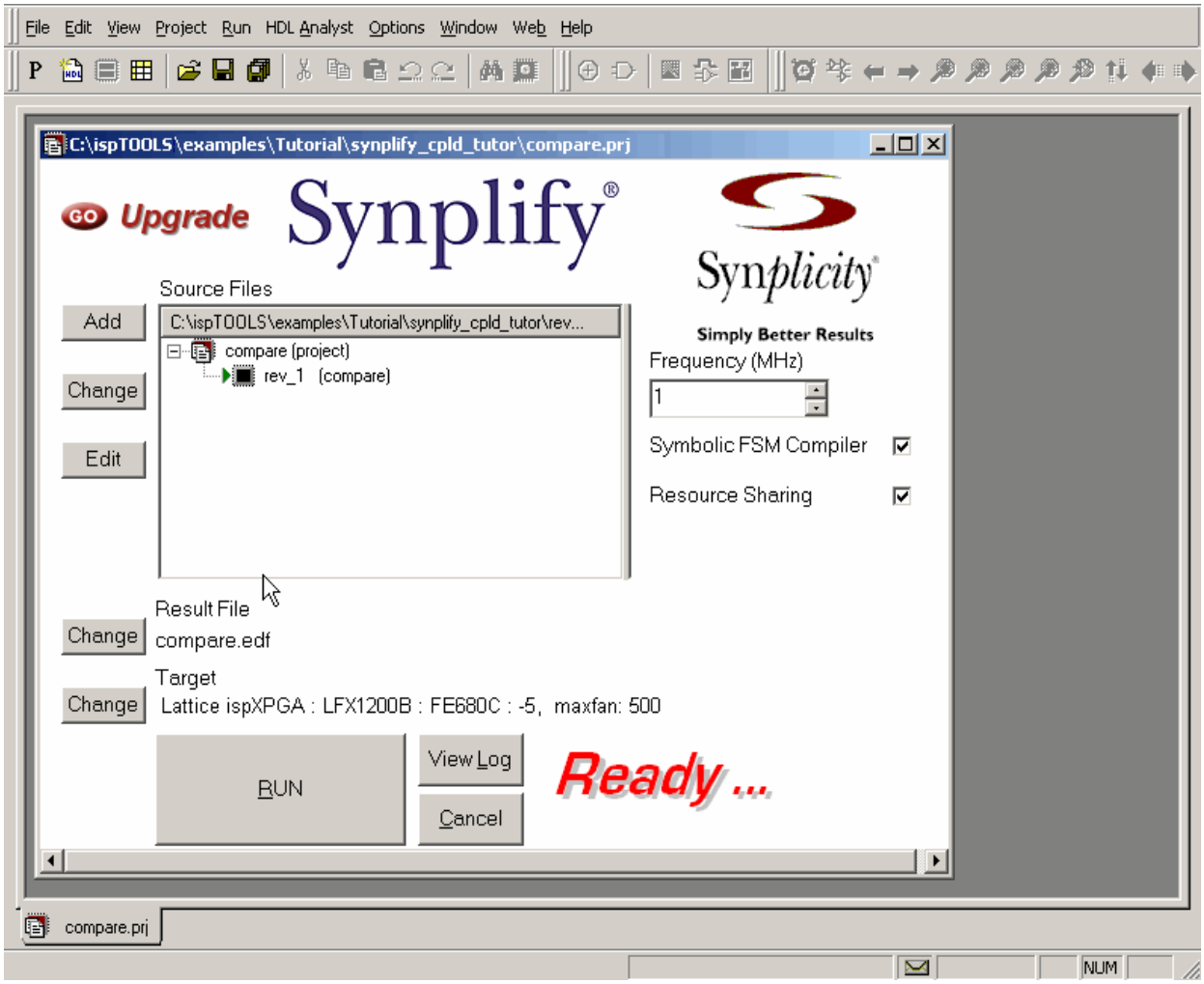
3. Choose **File > New** to open the New dialog box, shown in Figure 7.
4. In the dialog box, do the following:
   a. In the File Type box, select **Project File**.
   b. In the File Name box, type **compare**.
   c. In the File Location box, make sure you are in the **synplify_cpld_tutor** folder.
   d. Click **OK**.

Figure 7: New Dialog Box



The Synplify screen should look like the illustration in Figure 8.

Figure 8: Synplify Window Showing New Project



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:

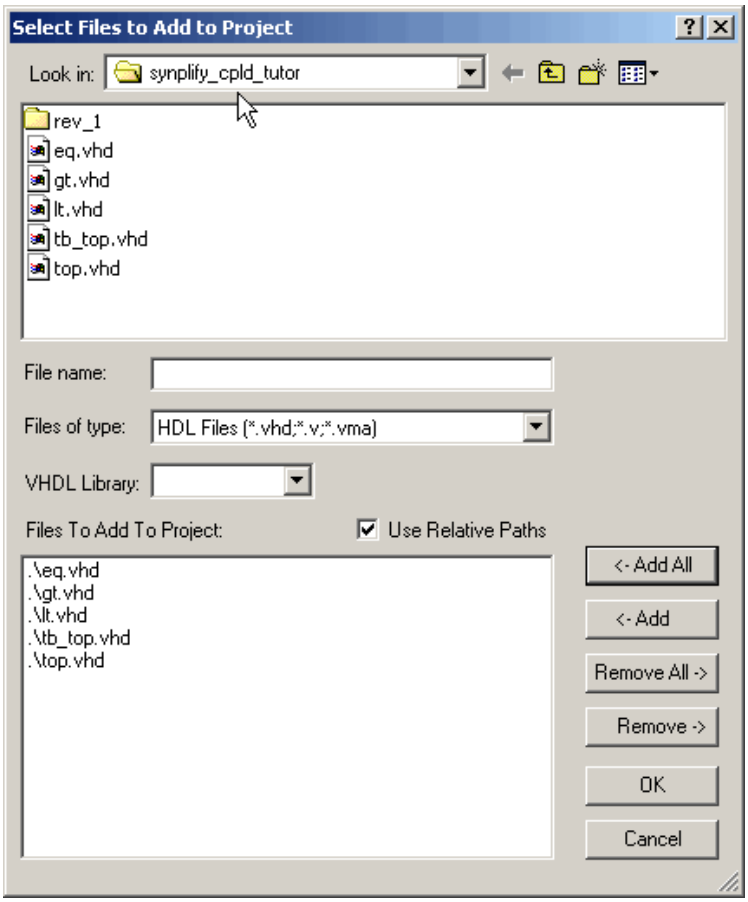
1. On the Synplify main window, click **Add** to open the Select Files to Add to Project dialog box, shown in Figure 9.

   You should see five 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.

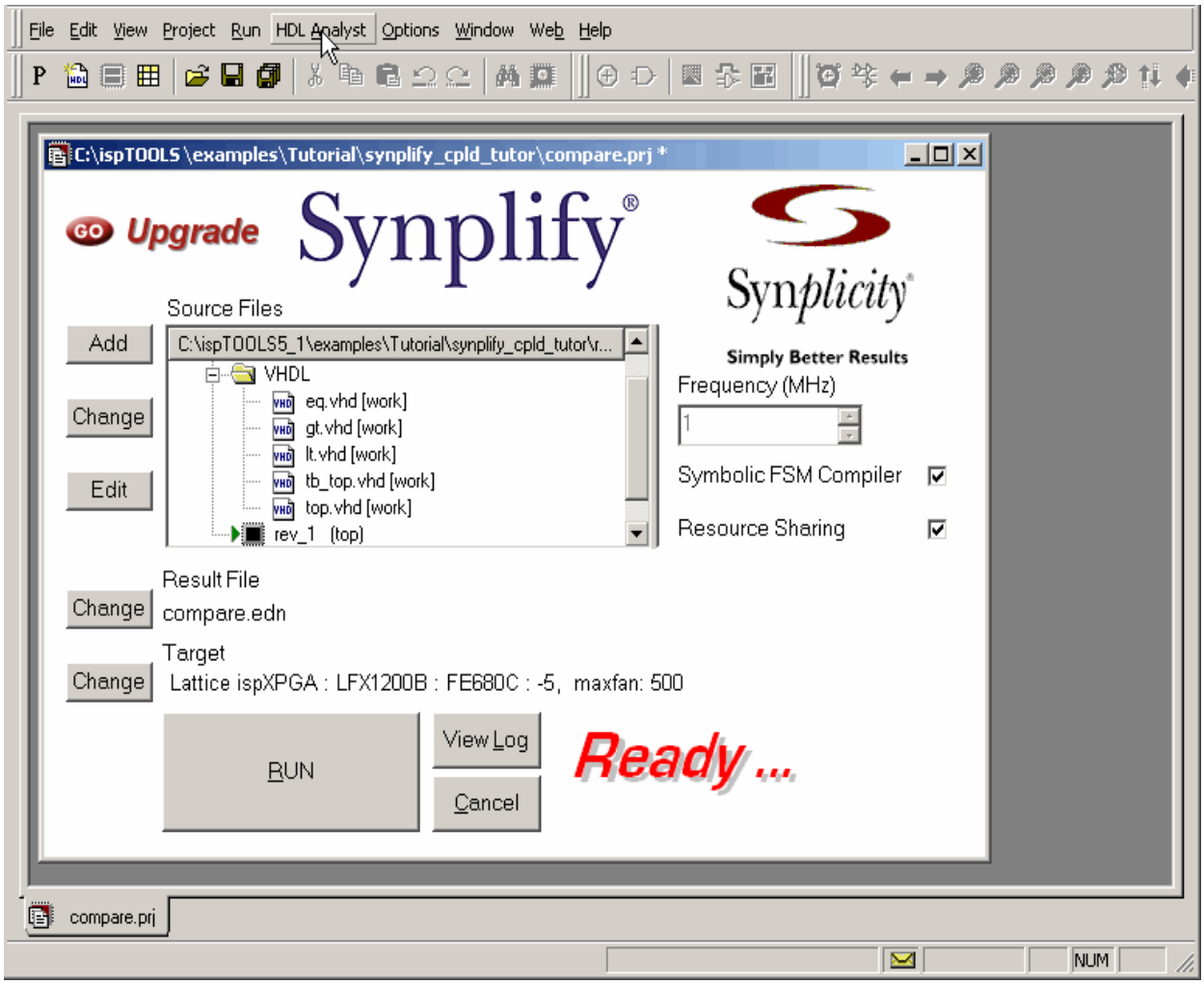
Figure 9: Select Files to Add to Project Dialog Box



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 the illustration in Figure 10.

Figure 10: Synplify Window Showing Added VHDL Files



Note

`top.vhd` must be at the bottom of the list of VHDL files for the project in the Synplicity graphical user interface, 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

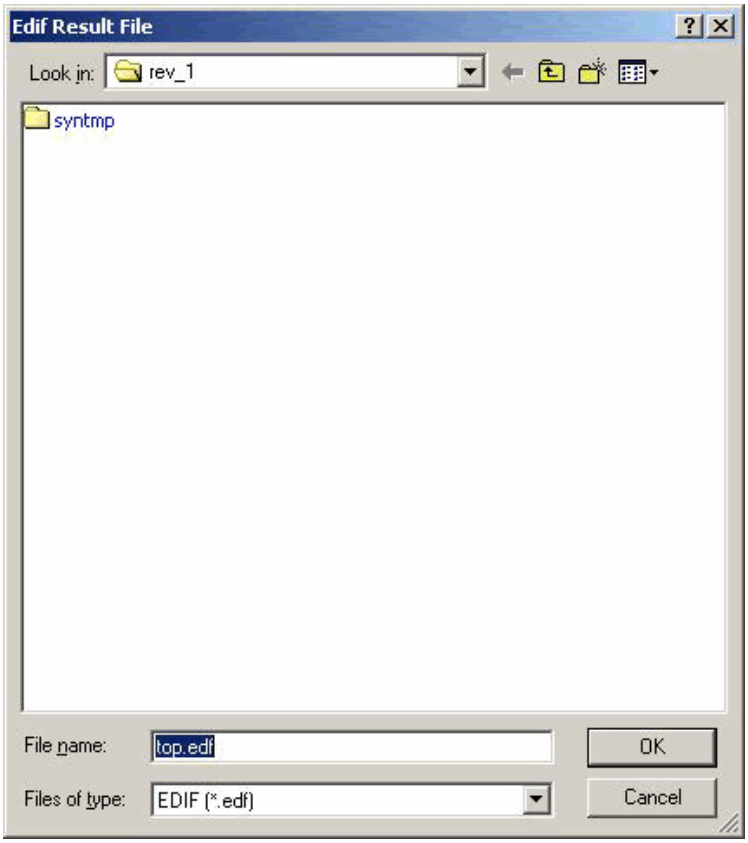
Synplify has probably set your implementation options correctly. However, it is a good practice to check them, especially if you had to change the location of top.vhd in the previous step.

To set the implementation options:

1. Click **Change (Result File)** to open the EDIF Result File dialog box, shown in Figure 11.

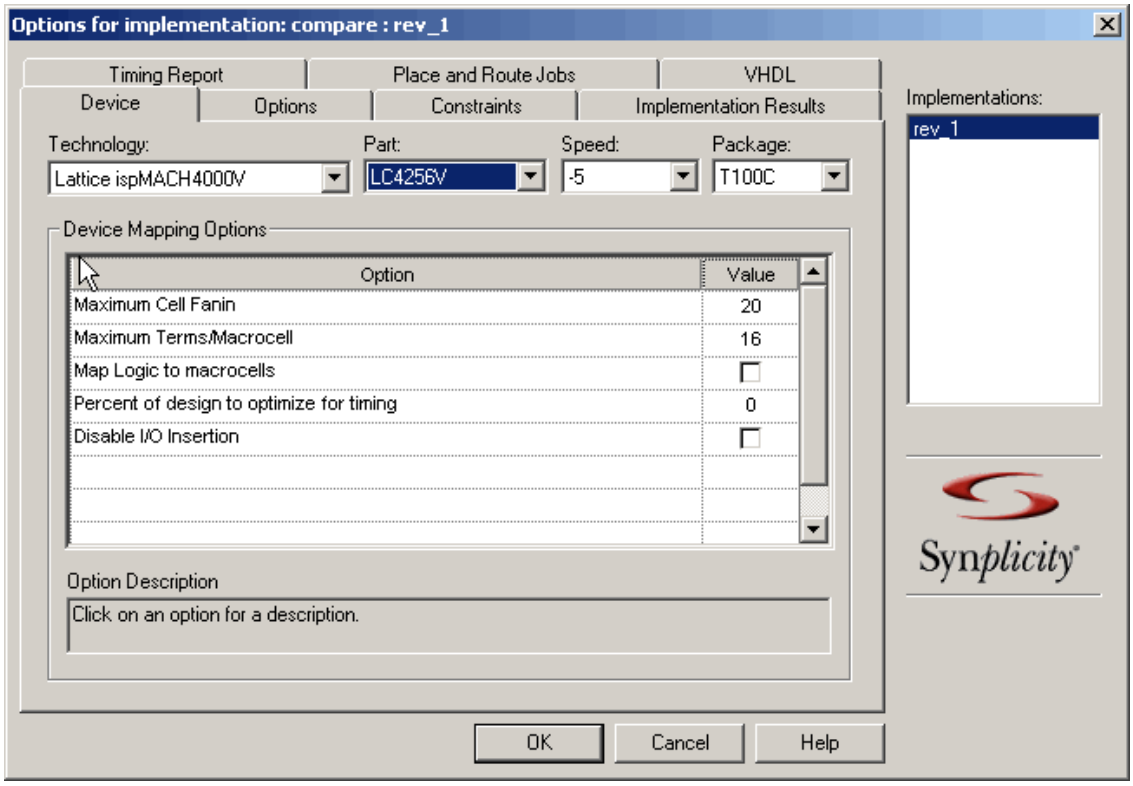
   Synplify automatically creates revision folders inside your project folder. You should be in the rev_1 folder inside your project folder.

Figure 11: EDIF Result File Dialog Box



2. Click **Cancel** to close the dialog box.
3. Click **Change (Target)** to open the Options for Implementation dialog box, shown in Figure 12.
4. In the dialog box, do the following:
   a. In the Technology box, make sure that Lattice **ispMACH4000V** is selected.
   b. In the Part box, select **LC4256V**.
   c. In the rest of the fields, accept the defaults.
   d. Click **OK**.

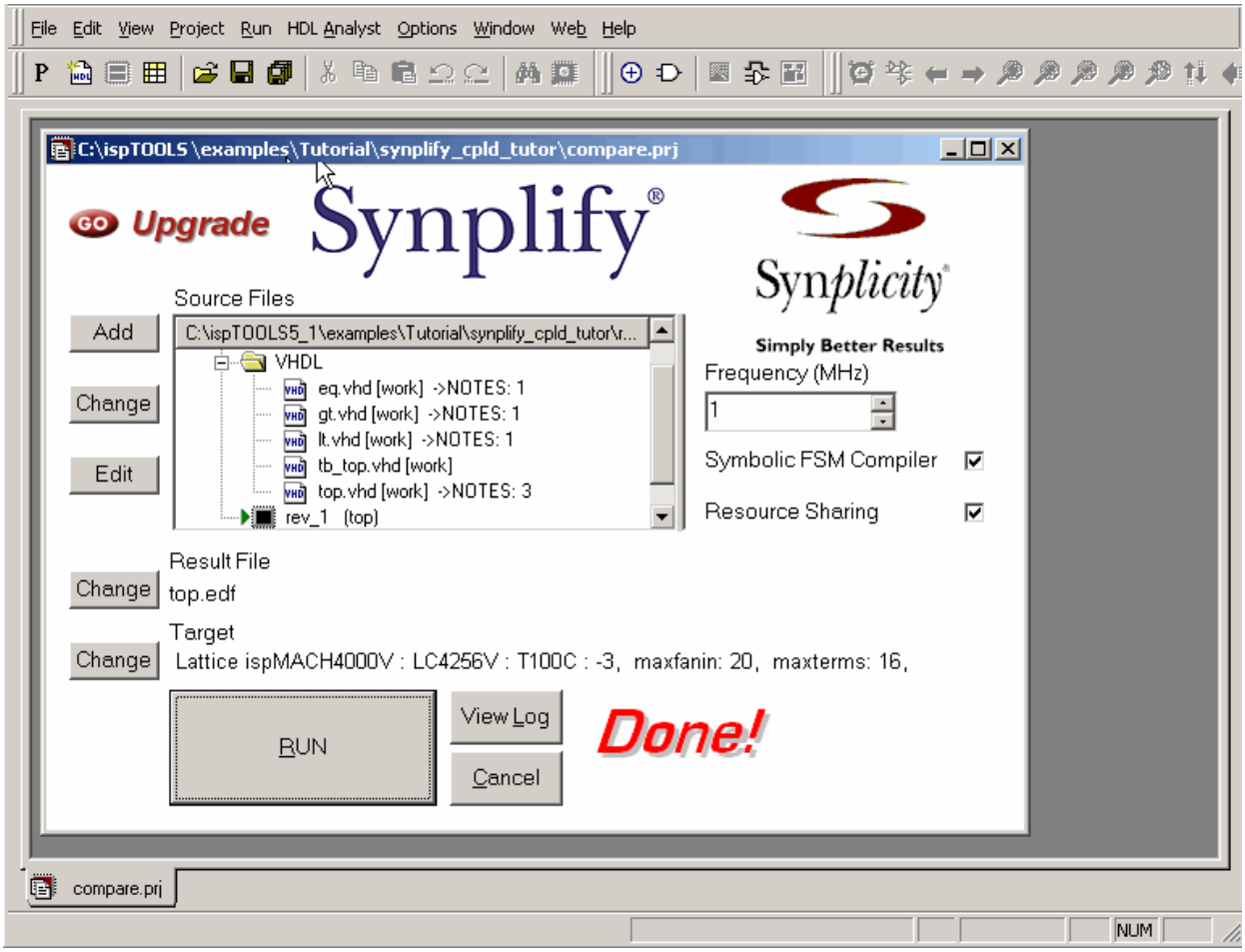
Figure 12: Options for Implementation Dialog Box



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, as shown in Figure 13. If you like, you can double-click the different files and view them. When you are finished, close the files.

Figure 13: Synplify Window Showing Completed Synthesis



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

Task 6: Import the EDIF File into Your Project

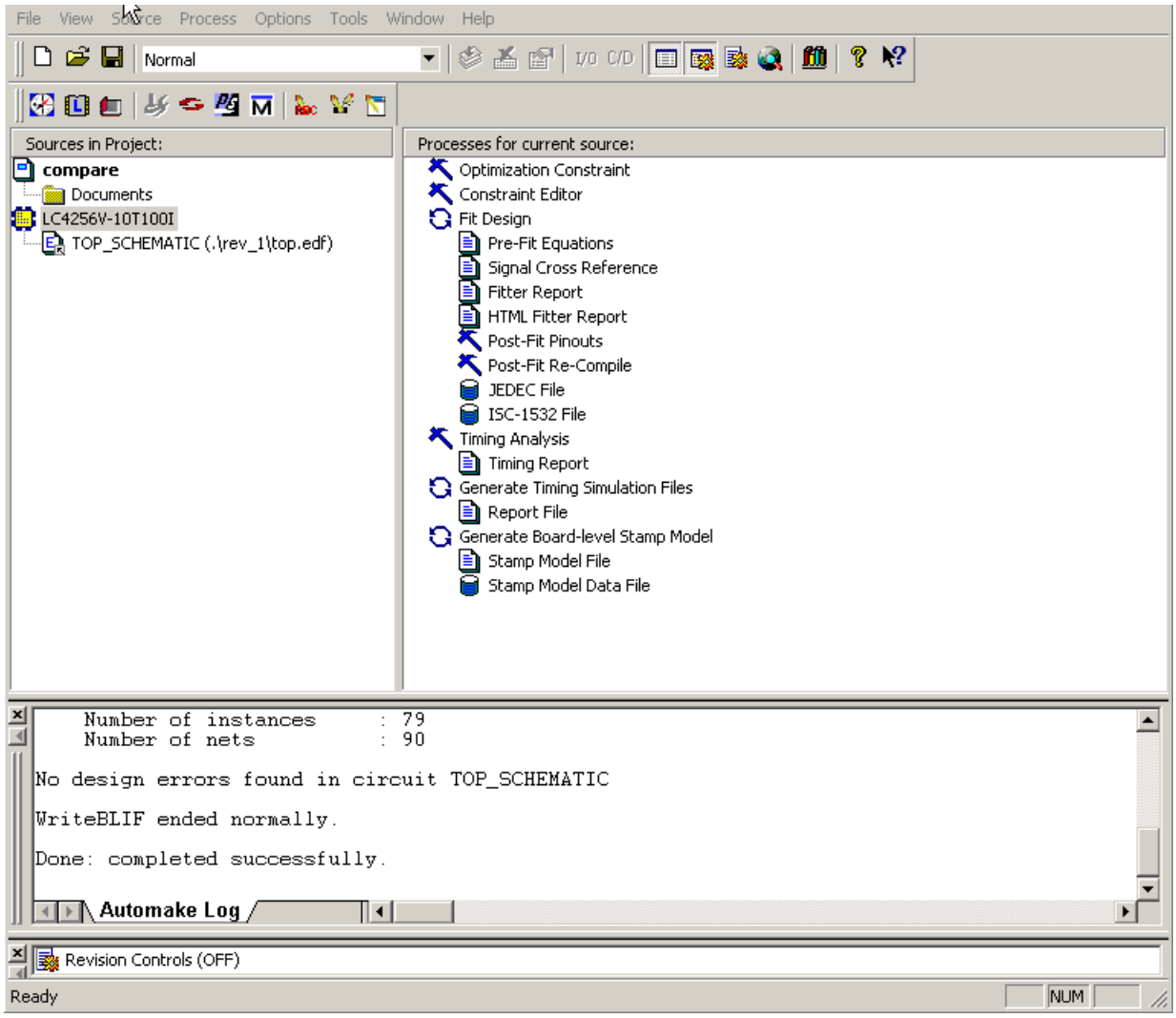
Now you are ready to import the EDIF file into ispLEVER project. You can import EDIF 2 0 0 netlists from third-party synthesis tools, such as Synplicity Synplify, into ispLEVER.

To import an EDIF netlist into your project:

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

   The software adds the selected EDIF file (`top.edf`) to the project sources, as shown in Figure 14.

Figure 14: Project Navigator Showing Added EDIF File



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: Fit the Design and View the Report

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.

At the end of a successful fitter run, the ispLEVER software generates a JEDEC file, as well as a fitter report so that you can see how the ispLEVER software has utilized and routed the part.

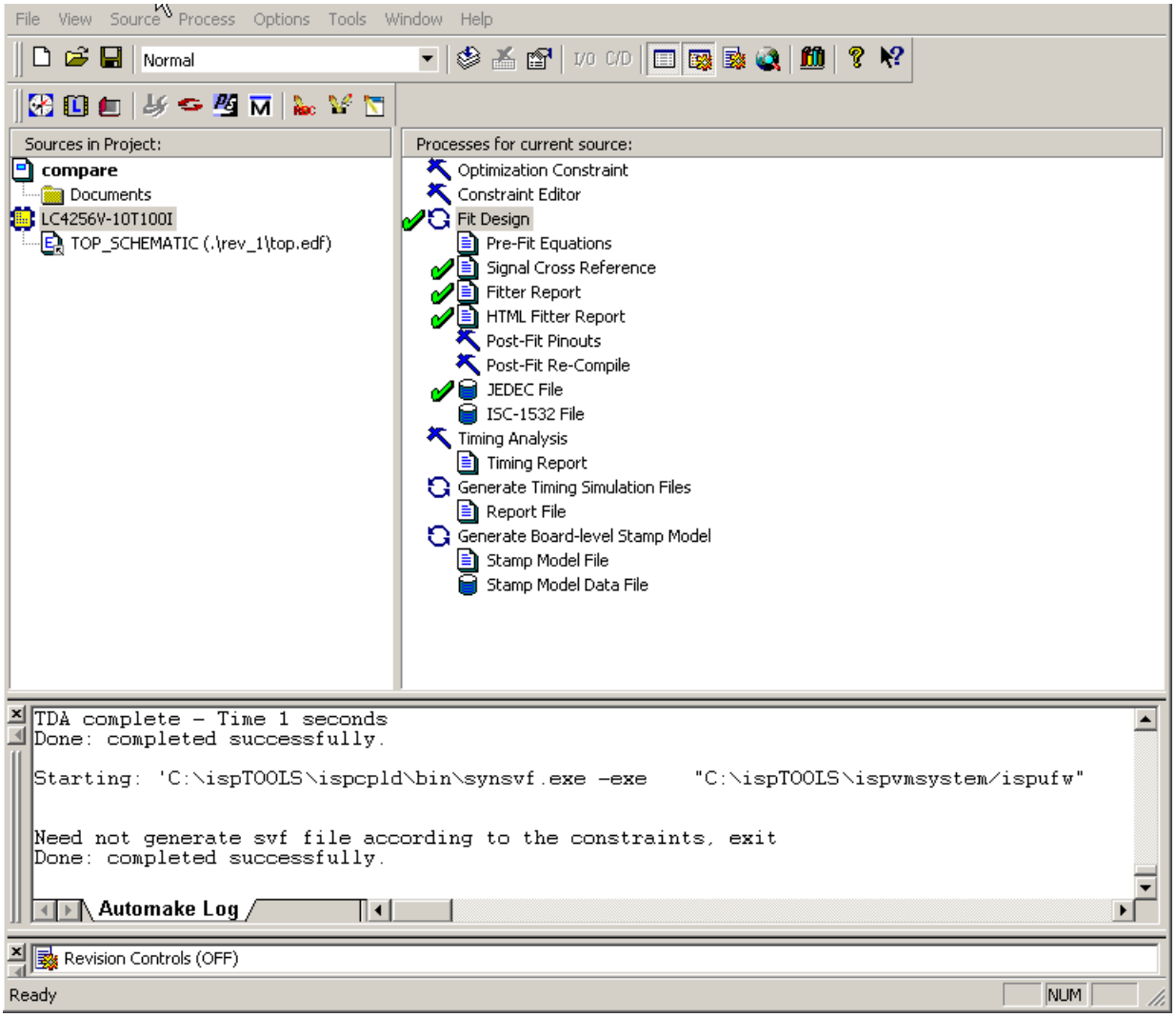
To run the Fitter and view the report:

1. With the target device selected in the Sources in project window, double-click **Fit Design** in the Processes for Current Source window to run the Fitter.

   The ispLEVER software successfully fits the design in the specified device and generates a JEDEC file. The results are shown in Figure 15.

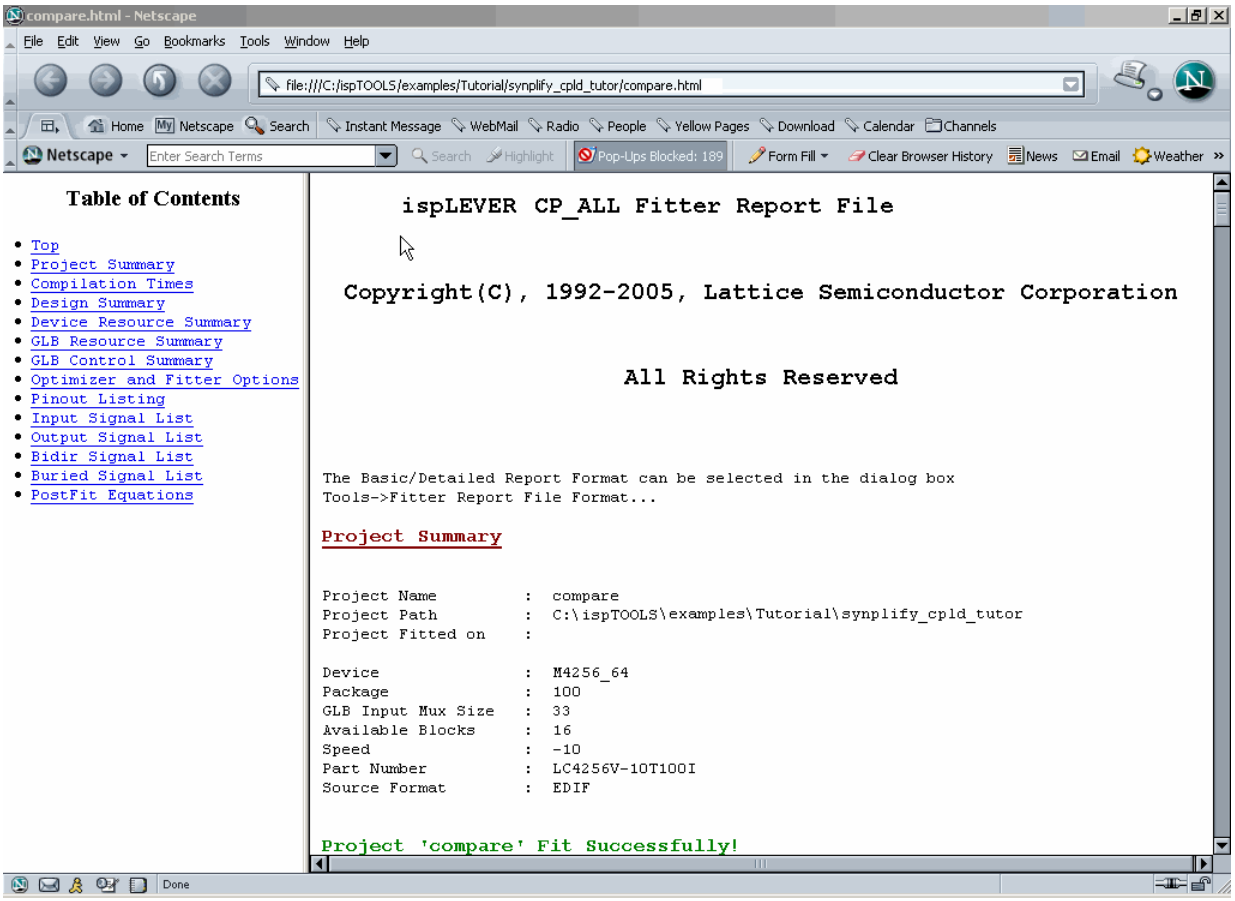
   Optional: If you like, you can right-click on the **JEDEC File** process and select **View** to look at the contents of the JEDEC file. Close the file when you are through.

Figure 15: Project Navigator Showing Fitted Design



2. Double-click on the **HTML Fitter Report** process to open the report in your browser, as shown in Figure 16. View the contents and then close the report.

Figure 16: HTML Fitter Report



Task 8: 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.

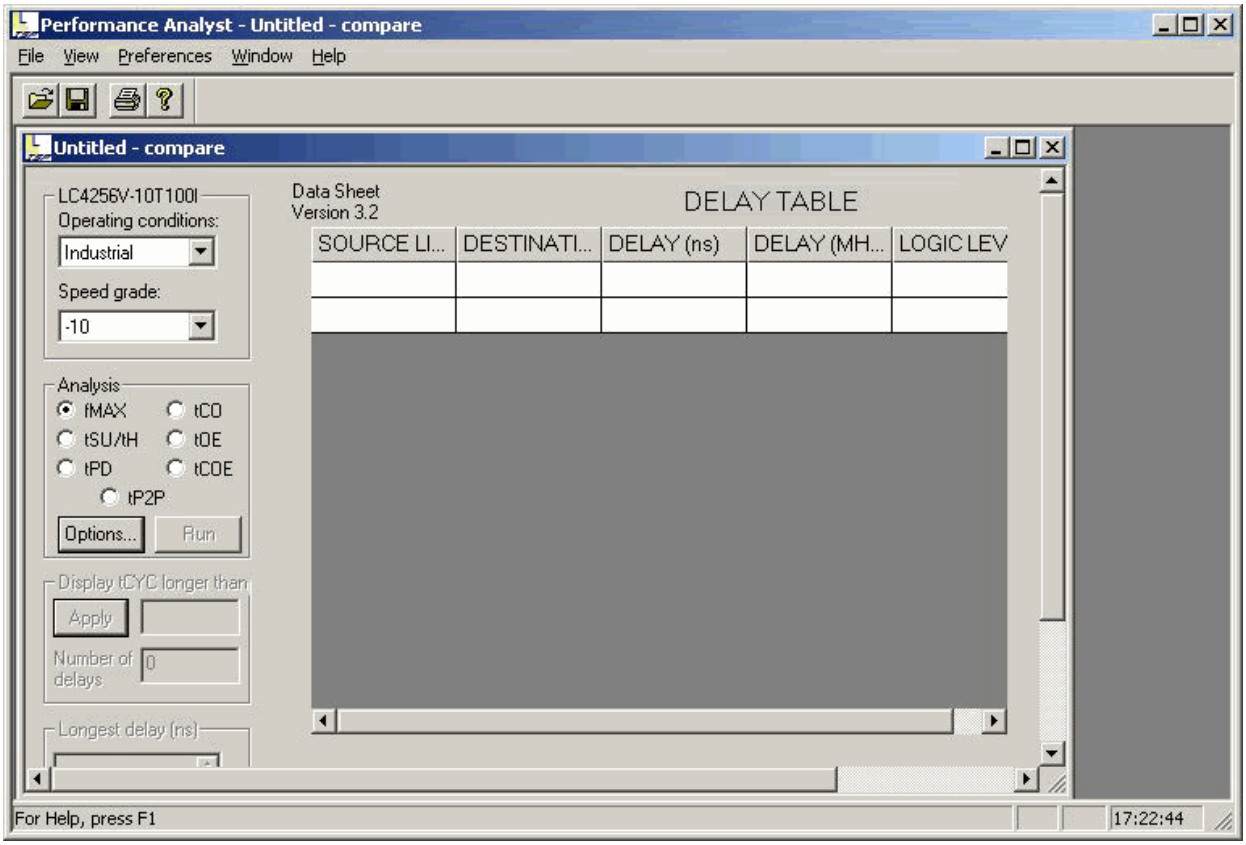
The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and

destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To perform 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 **Timing Analysis** process to run the timing analysis and open the Performance Analyst, shown in Figure 17.

Figure 17: Performance Analyst Window



The Performance Analyst performs seven distinct analysis types: fMAX, tSU/tH, tPD, tCO, tOE, tCOE and tP2P. 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 tP2P type is the path between any two user-specified pins. The remaining five 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**.

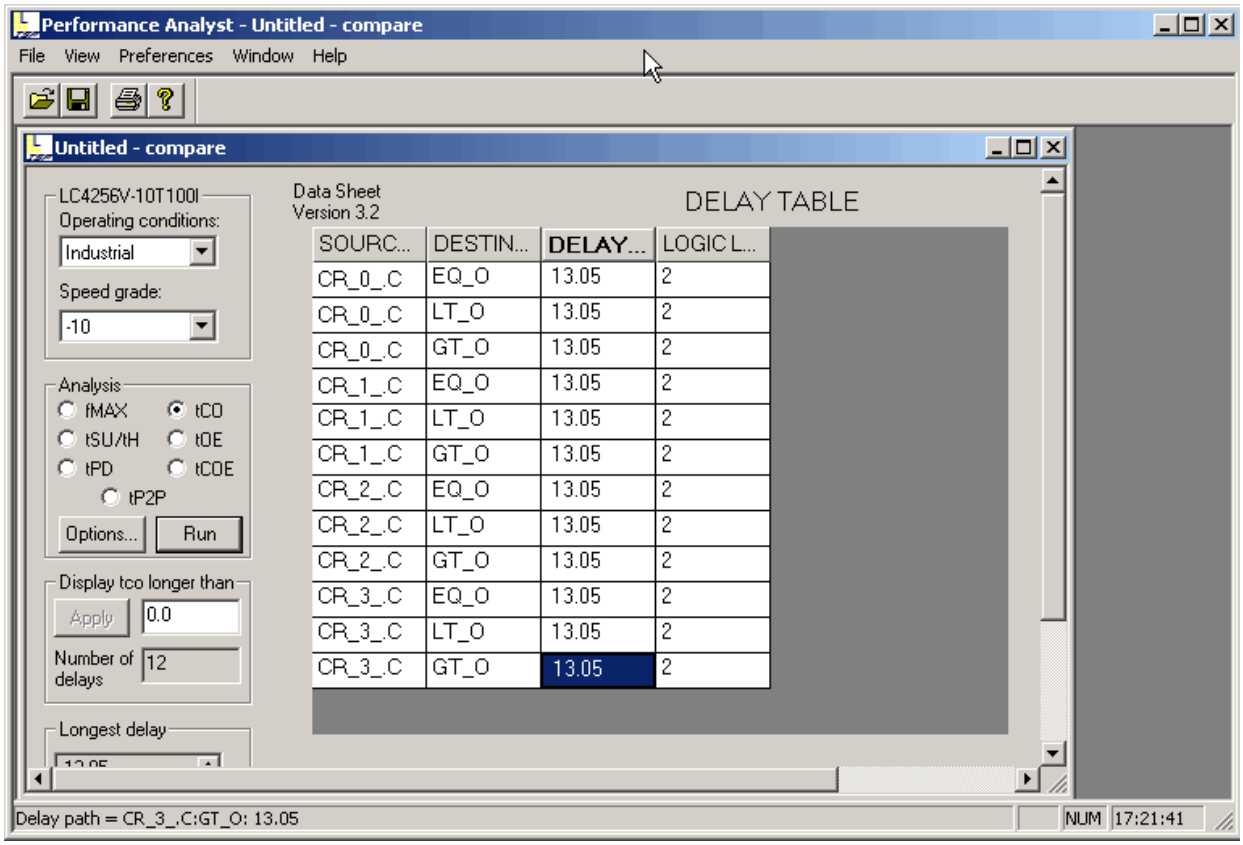
   The tCO path trace analysis reports clock-to-out delay starting from the primary input, going through the clock of flip-flops or gate of latches, and

ending at the primary output. In this case it is 13.05 ns Figure 18 shows the results of the analysis.

Note

Your timing numbers may differ slightly.

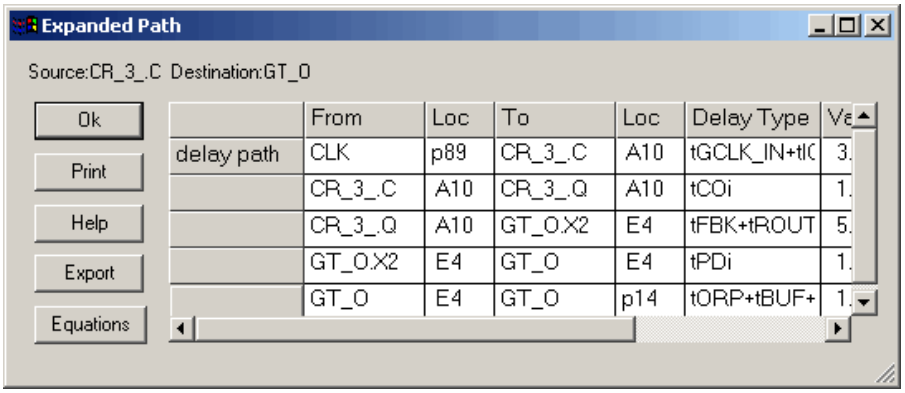
Figure 18: Performance Analyst Window Showing Clock-to-Out Delay



4. Click the highlighted cell (**13.05**) in the spreadsheet window to open the Expanded Path dialog box, shown in Figure 19.

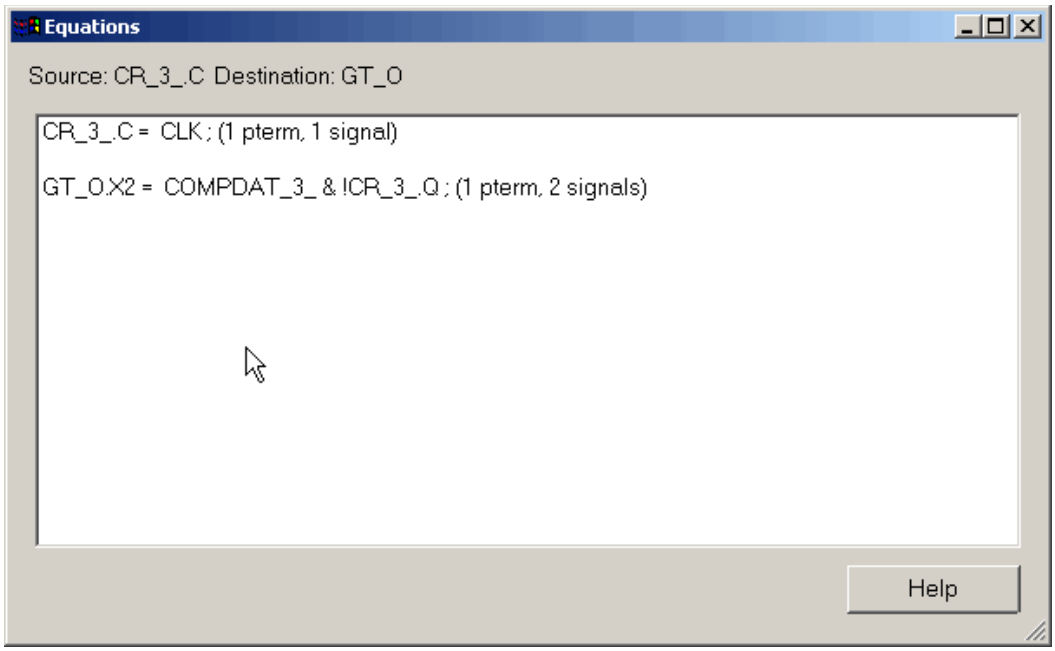
   This dialog box enables you to analyze individual timing components used to calculate the timing path. It shows a source pin (From) and a destination pin (To). It also shows the delay type, the delay of that path (value ns), and the cumulative delay of all the signals.

Figure 19: Expanded Path Dialog Box



5. Click **Equations** to open the Equations dialog box,shown in Figure 20, which shows the functional relationship between the selected source and destination.

Figure 20: Equations Dialog Box



6. Close the Performance Analyst without saving.
7. Close ispLEVER without saving.

Summary

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

- Create a new EDIF project in the ispLEVER system 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.
- Fit the design, generate a JEDEC file, and view the Fitter report.
- 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 Semiconductor ispLEVER online help
- Data sheets, technical notes, and other information on CPLDs on the Lattice Web site at http://www.latticesemi.com/search/literature.cfm. Click on **CPLDs**.