HDL Synthesis Design with LeonardoSpectrum: ORCA Flow

Table of Contents

HDL Synthesis Design with LeonardoSpectrum: ORCA Flow

This tutorial shows you how to use LeonardoSpectrum from within ispLEVER® to synthesize a VHDL design and generate an EDIF file. You will implement the design in a Lattice ORCA FPGA device and prepare the design for ModelSim timing simulation.

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

Learning Objectives

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

- Use ispLEVER to create a new EDIF project and target a device.
- Launch LeonardoSpectrum from within ispLEVER, synthesize your VHDL design, and generate an EDIF netlist file.
- Import the EDIF source file into ispLEVER.
- Implement the design using the Map, Place, and Route processes and view the reports.
- Prepare the design for timing simulation using ModelSim.

Time to Complete This Tutorial

The time to complete this tutorial is about 20 minutes.

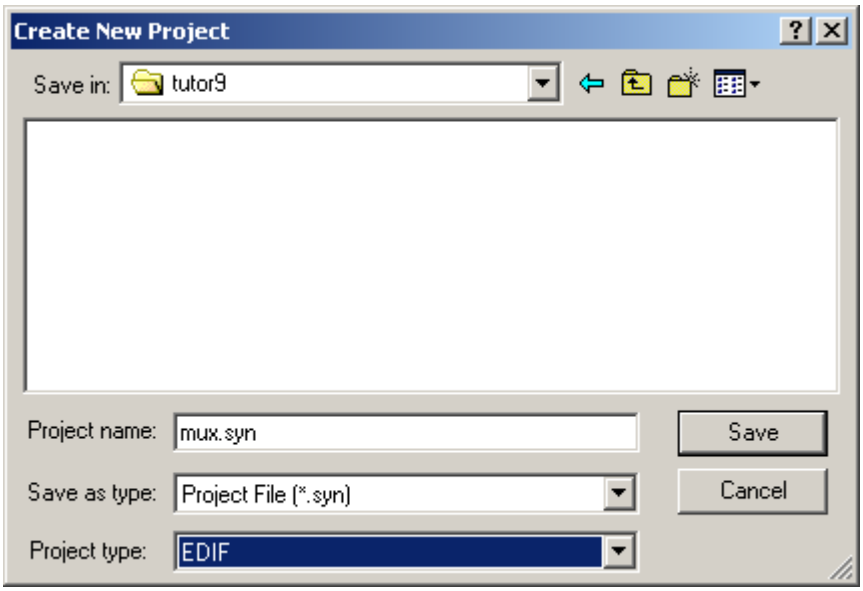
Task 1: Create a New Project

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

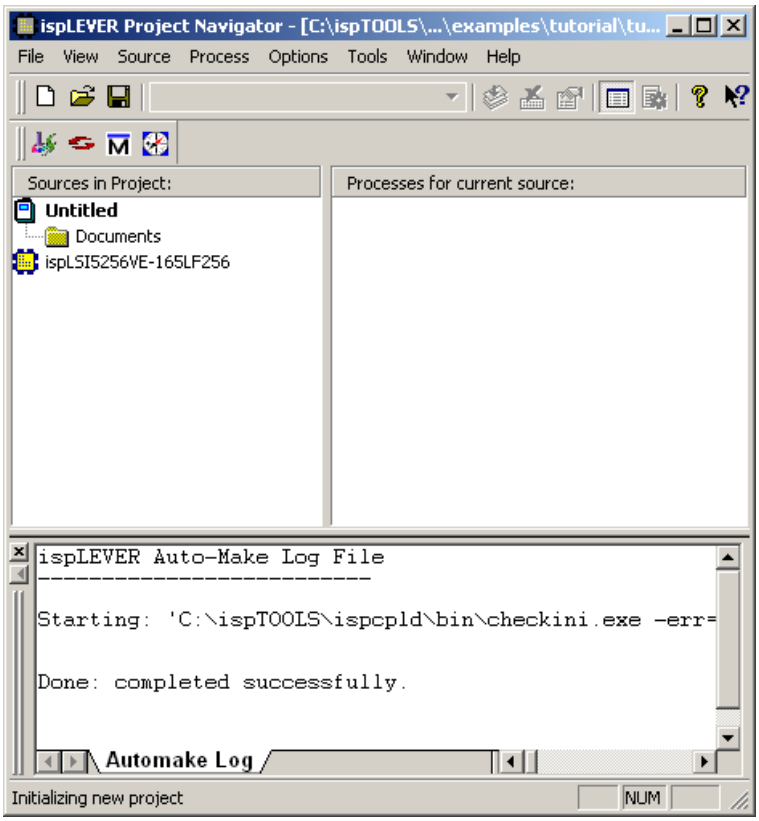
The ispLEVER software saves an initial design file with the `.syn` file extension in the directory you specify. All project files are copied to or created in this directory. The project type specifies that all design sources will be of this type.

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,
   - Change to the directory: `<install_path>\ispcpld\examples\tutorial\`**`tutor9`**.
   - In the Project name box, type **`mux.syn`**.
   - In the Project type box, select **EDIF**.
   - Click **Save**.



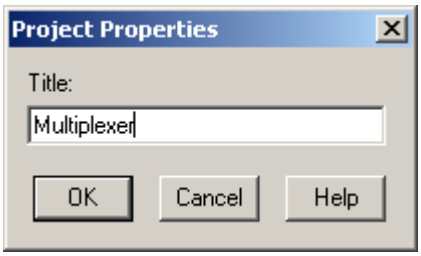
In the Sources window, the project title defaults to Untitled. You will change the project title in the next step.



4. In the Sources window, double-click the project title (**`Untitled`**) to open the Project Properties dialog box.

   The default title for a new project is "Untitled." You can create a title for the project with as many characters as you want. The title can contain spaces and any other keyboard character except tabs and returns.

5. Type **`Multiplexer`** as your project title and click **OK**.

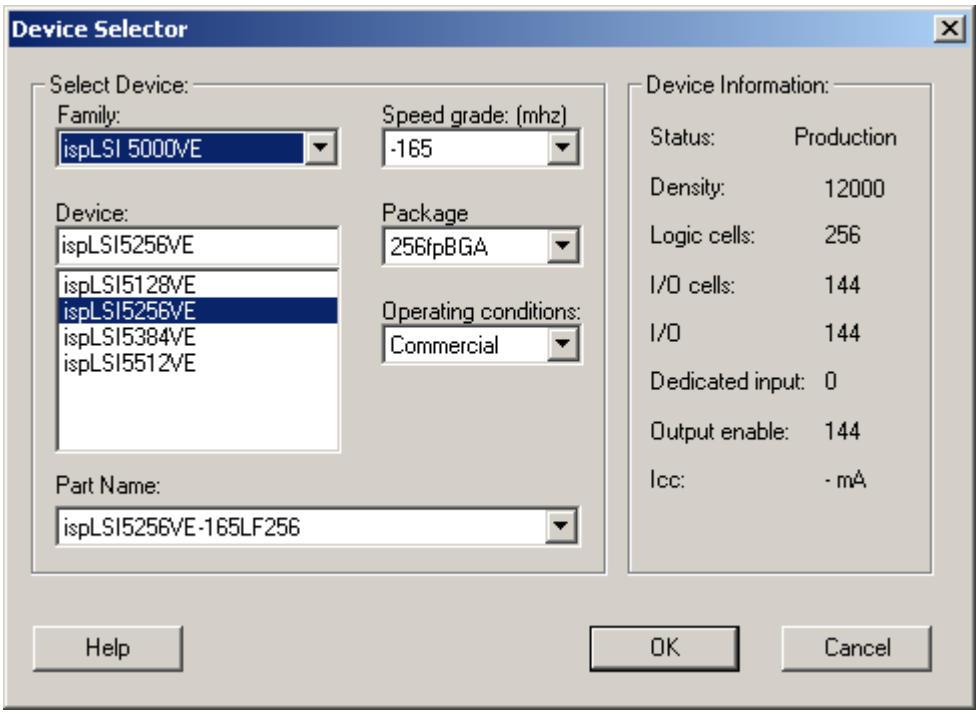


Task 2: Target a Device

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

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

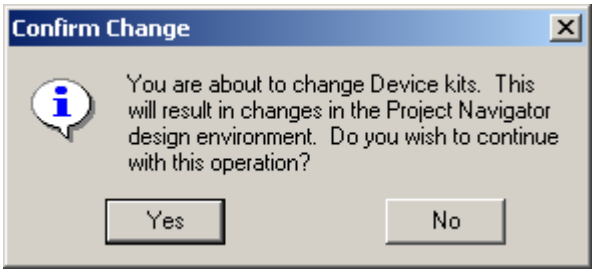
1. In the Sources window, double-click the **target device name** to open the Device Selector dialog box. The Device Selector dialog box shows the default device as well as all available devices and their options.



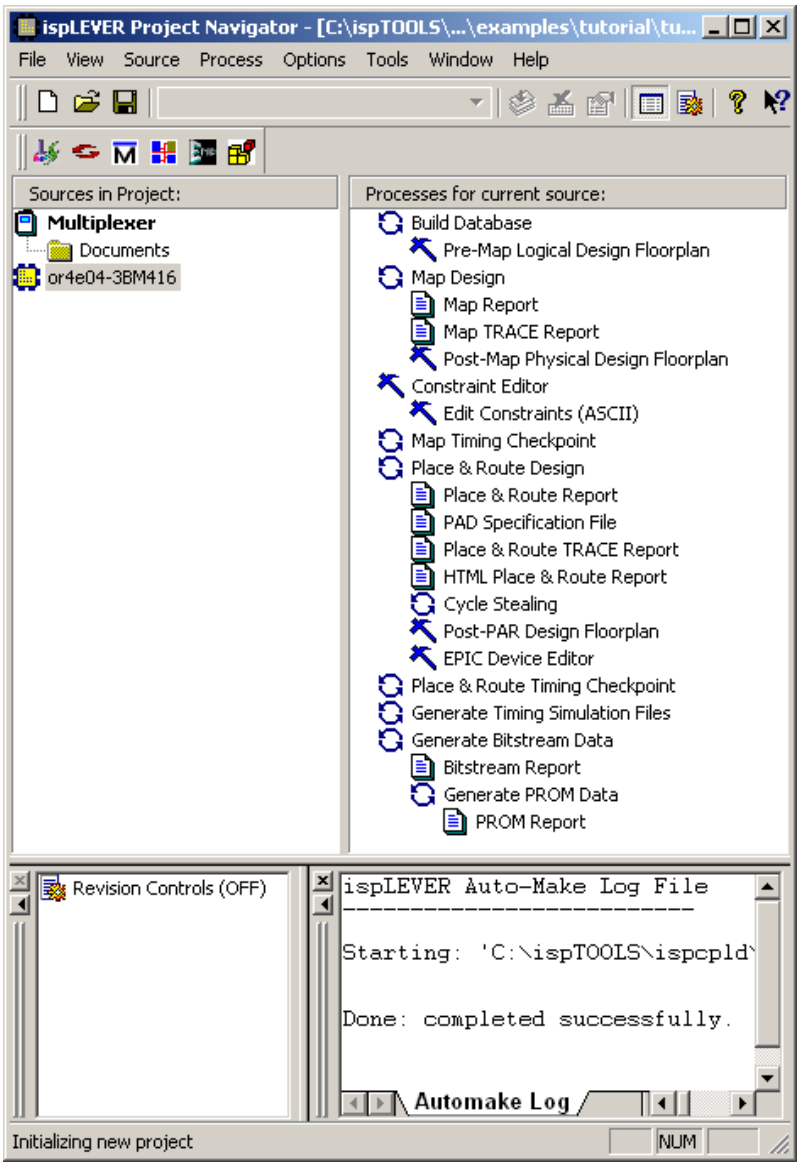
In the dialog box:

- Under Family, select **or4e00** from the drop-down list.
- Under Device, select **or4e04**.
- Under Part Name, select **or4e04-3BM416**.
- Accept the default settings and click **OK**.

2. In the Confirm Change dialog box, click **Yes** to confirm that you wish to change device kits.



3. Your Project Navigator should look like this:



Task 3: Start LeonardoSpectrum from ispLEVER

For HDL designs, the ispLEVER software provides two synthesis tools that are integrated into the Project Navigator environment: *LeonardoSpectrum* and *Synplify*. 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.

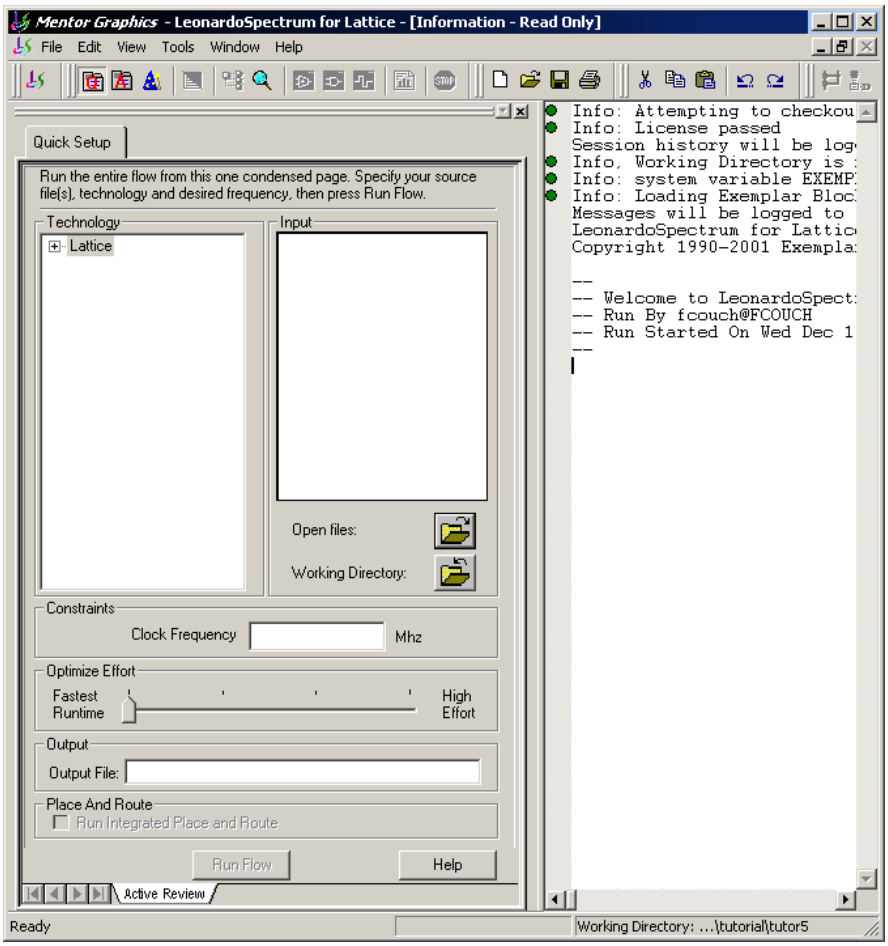
About LeonardoSpectrum for Lattice

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

When you start LeonardoSpectrum for the first time, the main window is maximized and displays the Tip of the Day and an information screen.

To start LeonardoSpectrum:

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



There are three ways to synthesize your design: *Quick Setup, Advanced Flow Tabs,* and *Synthesis Wizard*. For this tutorial you will use the Quick Setup method.

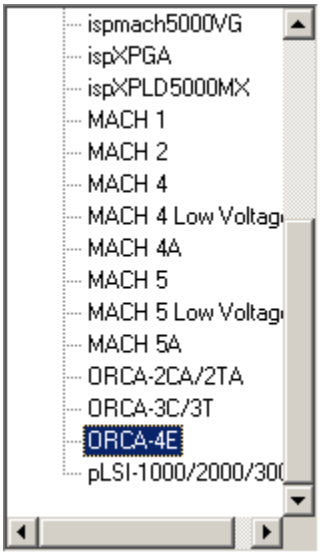
3. Make sure the **Quick Setup** tab is selected on the toolbar. Your screen should look similar to the one shown above. If not, choose **Tools > Quick Setup**.

Task 4: Use Quick Setup to Synthesize the Design

Quick Setup is a "push button" flow that you can use to achieve good first-pass synthesis results. You specify the target technology, open your input design files, optionally set the target clock frequency, and verify the name of the output netlist. When you click Run Flow, the entire synthesis flow is executed from start to finish. This includes synthesis, global constraints, optimization, and writing netlist. The output is an EDIF netlist that can be read by ispLEVER. Attributes that are placed on design objects by the HDL source code and LeonardoSpectrum are converted to properties in the EDIF netlist.

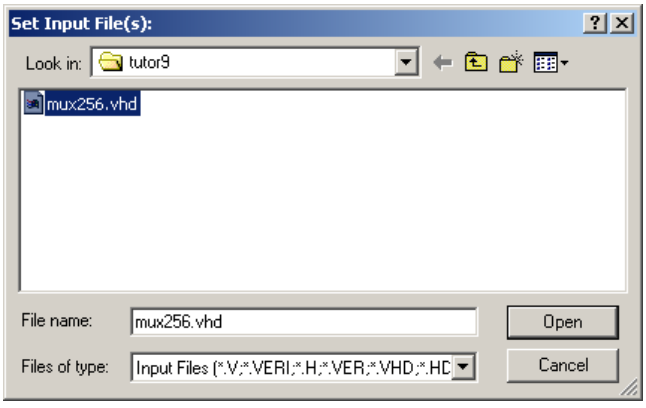
To synthesize the design:

1. On the Quick Setup tab under Technology, expand Lattice and select the **ORCA-4E** device family.



2. Under Input, click the **Open files** icon to open the Set Input File(s) dialog box.

   LeonardoSpectrum does not read pre-compiled HDL designs from disk. Instead, the source files are read directly into memory where LeonardoSpectrum builds an EDIF-like in-memory database.

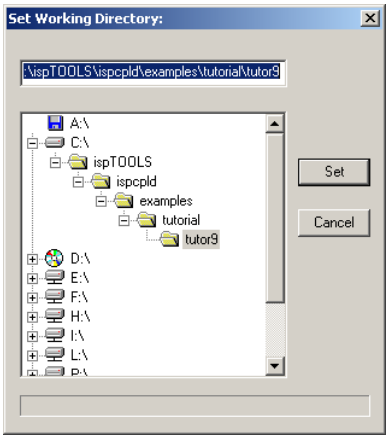


3. Make sure you're in the directory `tutorial\tutor9`. Select **`mux256.vhd`** and click **Open**.

   Notice that LeonardoSpectrum automatically points the output file to the project directory and places the file name in the Input box.

4. Directly below the Open Files icon, click the **Working Directory** icon to open the Set Working Directory dialog box. The working directory is where LeonardoSpectrum places all generated output files. These files include the output files from the synthesis process. For ispLEVER projects, you should make the working directory the same as your project directory.

5. Make sure the path is pointing to `<install_path>\ispcpld\examples\tutorial\tutor9`, and then click **Set** to close the dialog box.



***Note**: Only a few files will be generated in this tutorial. Because the number of generated files in a real project can be many, it is a good practice to separate your design source files and batch scripts into a separate sub-directory. For example, the input source files could be kept in a sub-directory named* src. *Then, if your first synthesis run generates the "fastest" possible circuit, you may want to do one or more optional runs to evaluate the tradeoffs between speed and area. You can simply copy the src sub-directory into a new working directory named "smallest", for example, and the new generated files for the next run will be placed there.*

6. At the bottom of the Quick Setup tab, click **Run Flow**.

   LeonardoSpectrum reads the opened input files and creates an in-memory EDIF style database. This is called the RTL database and the design is composed of generic gates and non-mapped (black box) modules such as operators, counters, and inferred RAMs. Next, the in memory design is mapped to the specified technology, globally optimized, and the results for each module are saved. If a timing constraint is not met at this point, additional critical path optimizations are run to try to meet the constraints. The results are kept in a second in-memory technology-mapped design database. The output EDIF netlist and support files are then automatically generated and written to the working directory.

***Note**: The **Run Flow** button is not active until you have selected your Input File(s) and target technology. When the synthesis process is complete, the Information window on the right says that the run successfully ended.*

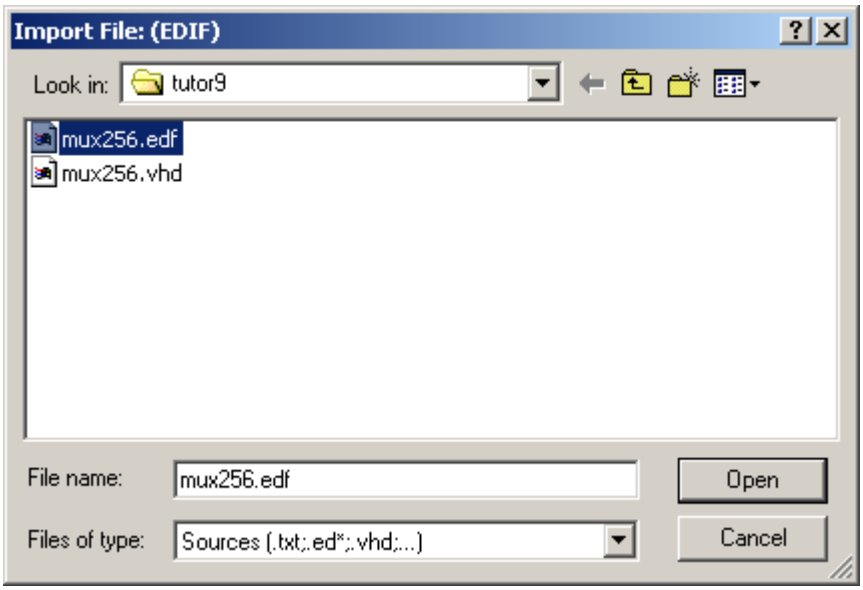
7. Choose **File > Exit** to exit LeonardoSpectrum without saving the project.

Task 5: Import the EDIF File into Your Project

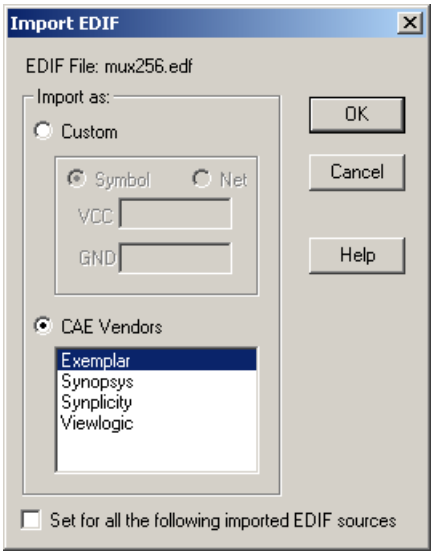
Using the Project Navigator, you can import EDIF 2.0.0 netlists from third-party synthesis tools, such as LeonardoSpectrum, into ispLEVER. In this task, you will import the EDIF netlist you synthesized in the previous task into your project.

To import an EDIF netlist into your project:

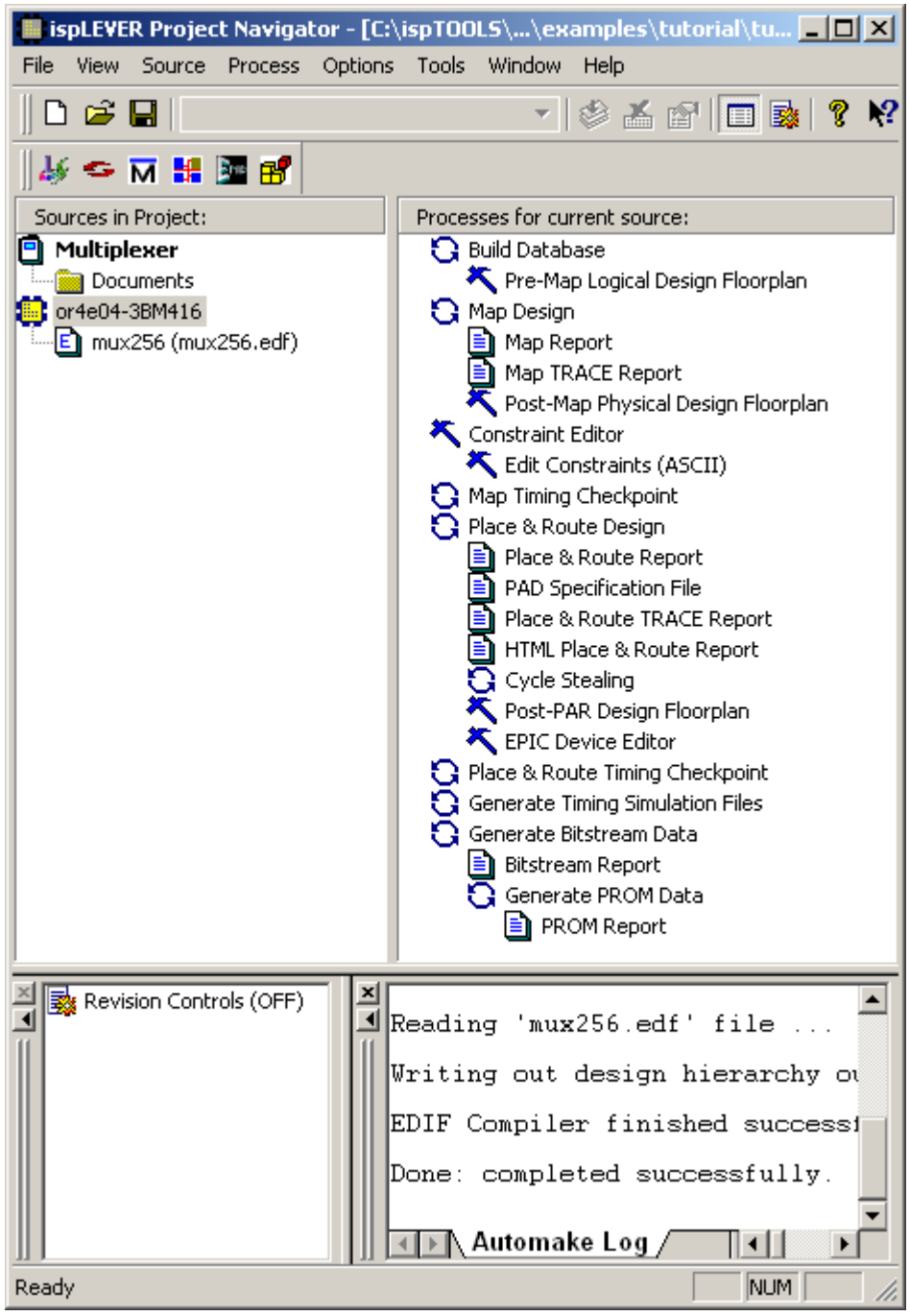
1. In the ispLEVER Project Navigator, choose **Source > Import** to open the dialog box.



2. Select **`mux256.edf`**, and then click **Open**.
3. In the Import EDIF dialog box under CAE Vendors, make sure **Exemplar** is selected, and then click **OK**.



The software adds the selected EDIF file (`mux256.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 6: Map the Design

In this task you will review Map options, run the Map program, and view the Map report. Using the EDIF netlist file you synthesized and imported into Project Navigator as input, you will actually run two processes, Build and Map. Before mapping can occur, the Build process automatically translates and builds your netlist into an ORCA-based database file. The ispLEVER software automatically does this netlist conversion when you start the Map process.

About the Map Process

Mapping is the process of converting a design represented as a logical network of device-independent components (e.g., gates and flip-flops) in the input logical design file into a network of device-specific components (e.g., configurable logic blocks) to be placed on the physical device represented in the post-map physical design (`.ncd`) file. Before mapping occurs, the EDIF netlist input is automatically converted into a logical generic database file (`.ngd`) in terms of ORCA FPGA primitives during the Build operation. After mapping, the resulting physical description is output to a Physical Design file (`.ncd`), in terms of the components in the target architecture. This physical design can then be placed and routed.

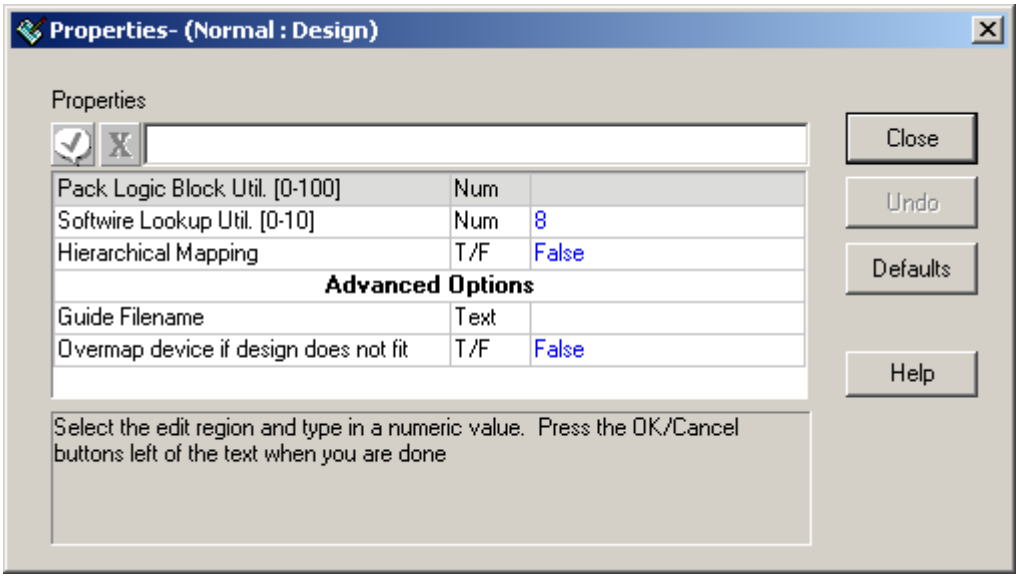
Also note that the resulting mapped `.ncd` files may include information based on user-defined attributes (ORCA properties) specified in the input netlist. For example, a property may specify where a component in the logical design must be placed in the physical device, or what nets should be given the highest priority when the design is routed. You can also use the ORCA Floorplanner to define component and regional grouping in the pre-map `.ngd` file in the logical domain and/or in the post-map `.ncd` in the physical domain.

To learn more about the Map process, refer to the "Mapping A Design" topic in the ispLEVER Help system. To learn more about using attributes (properties) in your netlist, download or view the ORCA Properties for Design Entry Desk Reference guide in PDF format using the Project Navigator: Help > ispLEVER User Documents command. See appropriate topics on floorplanning in ispLEVER for more on that subject.

To review Map properties, run the Map process, and view the Map report:

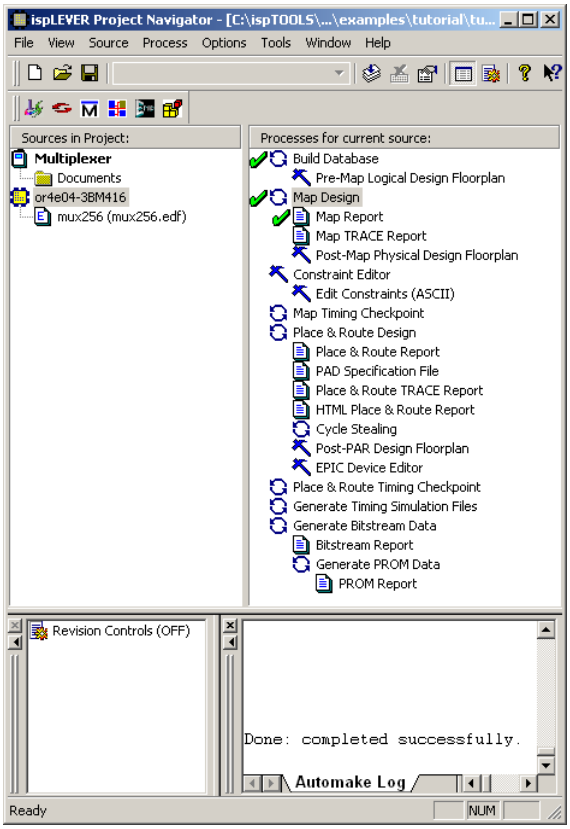
1. Before you run the Map program, you can set Map properties. Many ispLEVER processes have associated properties, which are displayed in the Properties dialog box.

   With the target device selected in the Sources window, right-click **Map Design** in the Processes window and choose **Properties** from the popup menu to open the dialog box.

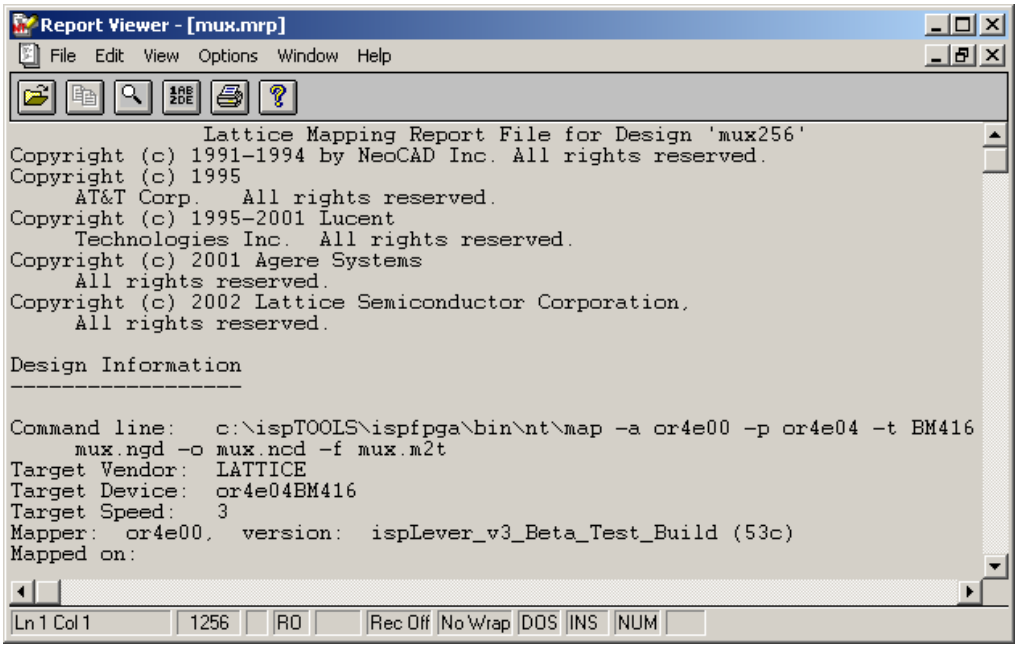


Properties can be True/False, List, or Text/Number. Accept the default properties and click **Close** to close the dialog box. Now you are ready to map the design.

2. In the Processes window, double-click the **Map Design** process. When the process is completed, a green check mark appears next to the Build Database, Map Design, and Map Report processes.



3. Double-click the **Map Report** process to open the report in the Report Viewer.



The Map Report (`.mrp`) file is an ASCII file containing information about the MAP command run. The `.mrp` file lists any DRC errors found in the design, details how the design was mapped (e.g., the schematic constraints specified, the logic that was removed or added, and how signals and symbols in the logical design were mapped into signals and components in the physical design). It also supplies statistics about component usage in the mapped design and contains information about the Map Design process run.

4. View the contents and then choose **File > Exit** to close the report.

Task 7: Place & Route the Design

After a design has undergone the necessary translation to bring it into the physical design format, you can run the Place & Route Design process (PAR). This process takes a mapped physical design file (`.ncd`), places and routes the design, and outputs a file that can then be processed by other design implementation tools.

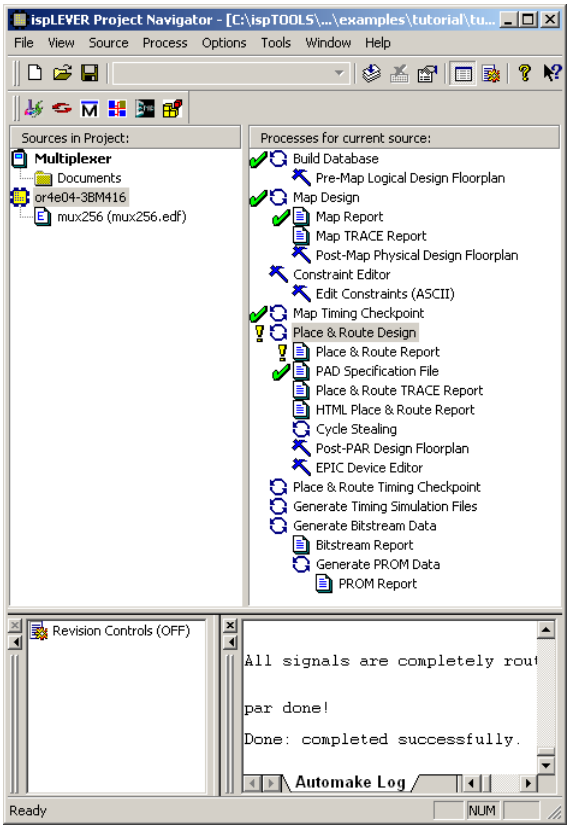
About the Place & Route Process

PAR can place and route a design in two different ways: cost-based or timing driven. The standard PAR package is a cost-based tool. This means that placement and routing are performed using various cost tables that assign weighted values to relevant factors such as constraints, length of connection, and available routing resources. First, the PAR process places the mapped physical design file (`.ncd`) in two stages: a constructive placement and an optimizing placement. PAR writes the physical design after each of these two stages completes.

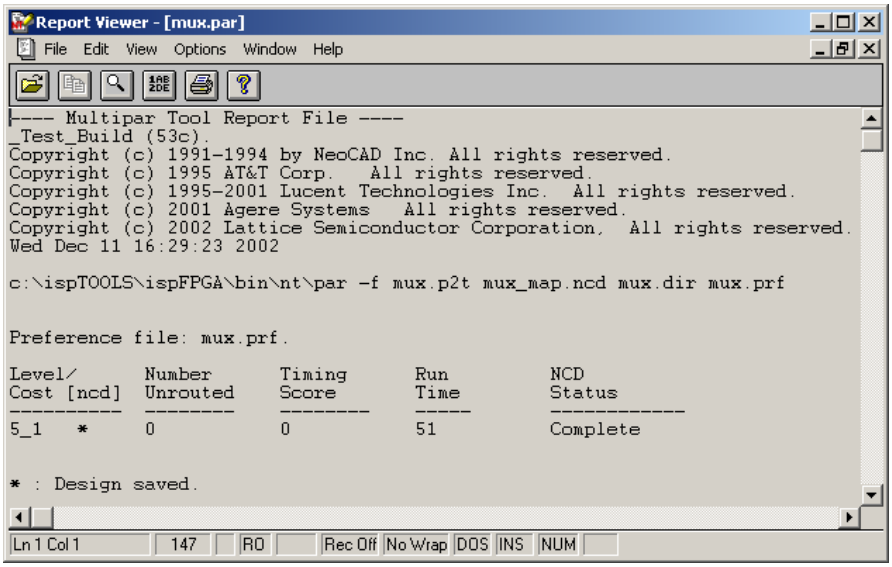
Second, routing also is done in two stages: iterative routing and delay reduction routing (also called cleanup). PAR writes the physical design file (`.ncd`) only after iterations where the routing score has improved. During iterative routing, the router performs an iterative procedure to converge on a solution that routes the design to completion or minimizes the number of unrouted nets. During reduction routing, the router takes the result of iterative routing and re-routes some connections to minimize the signal delays within the device.

To run Place & Route and view the reports:

1. Double-click the **Place & Route Design** process. The ispLEVER software places and routes the design in the specified device and generates the Place & Route Report. When finished, check marks appear beside the completed processes. (Ignore any warnings.)



2. Double-click the **Place & Route Report** process to open the PAR report file (`.par`) in the Report Viewer. The `.par` file contains execution information about the Place & Route command run. The report also shows the steps taken as the program converges on a placement and routing solution.



3. Choose **File > Exit** to close the report.

Task 8: Prepare for Simulation

The ispLEVER software supports third-party timing simulation with ModelSim™ from Mentor Graphics®. Using this integrated package, you can simulate single Verilog or VHDL designs within one environment.

About Timing Simulation

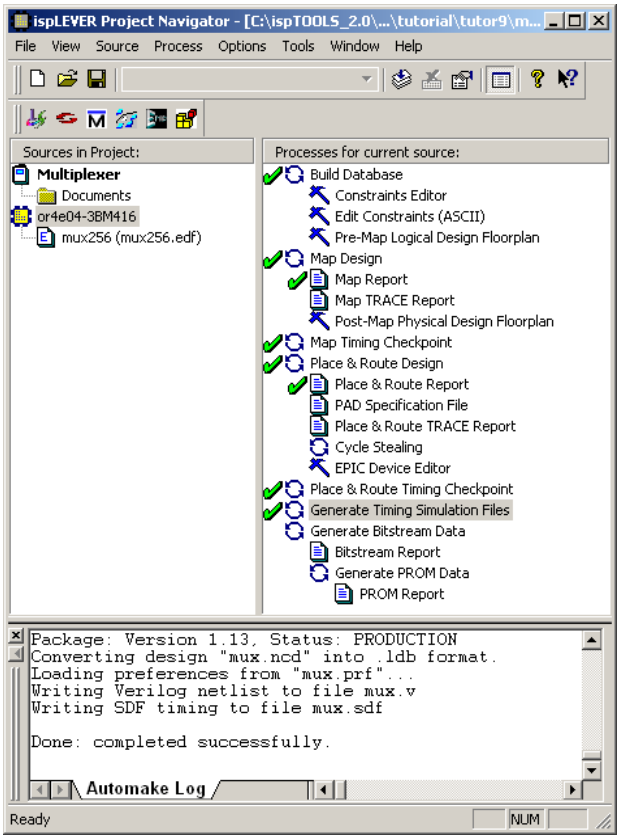
Where as timing *analysis* returns partial timing information, dynamic timing *simulation* will give you detailed information about gate delays and worst-case circuit conditions. Because total delay of a complete circuit will depend on the number of gates the signal sees and on the way the gates have been placed in the device, timing simulation can only be run after the design has been implemented.

Timing simulation also requires several input files to run - a VHDL netlist, a delay file, and a VHDL test bench. The netlist and delay files are automatically generated from within ispLEVER. The test bench must be written before hand.

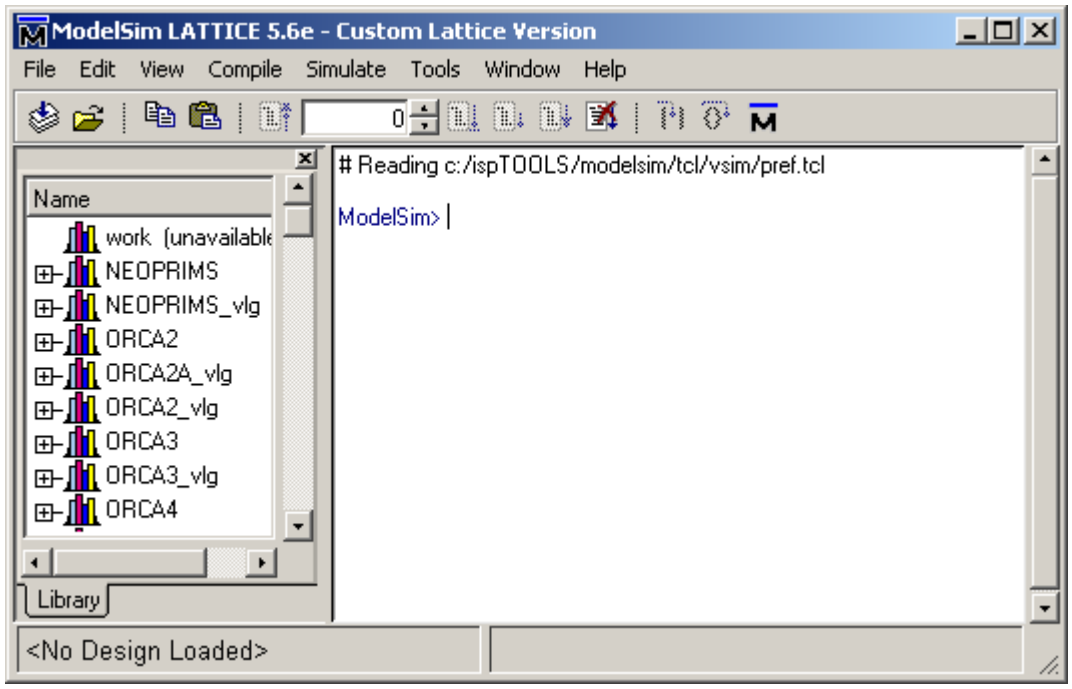
The Generate Timing Simulation Files process generates the timing simulation files for 3rd party simulators. Files generated include netlist files such as EDIF, VHDL, and Verilog, as well as delay files, such as SDF and DTB (Viewlogic).

To create the timing simulation files:

1. In the Processes window, double-click the **Generate Timing Simulation Files** process. The software writes two files to the project directory: `mux.vo` (the VHDL netlist) and `mux.sdf` (the standard delay file).



2. To run ModelSim for timing simulation, in the Project Navigator choose **Tools > ModelSim Simulator**.



Note: To complete this task, you would need a VHDL test bench file, which is not included as part of this tutorial.

Congratulations

You have completed the HDL Synthesis Design with LeonardoSpectrum tutorial. In this tutorial you have learned how to:

- Use ispLEVER to create a new EDIF project and target a device.
- Launch LeonardoSpectrum from within ispLEVER, synthesize your VHDL design, and generate an EDIF netlist file.
- Import the EDIF source file into ispLEVER.
- Implement the design using the Map, Place, and Route processes and view the reports.
- Prepare the design for timing simulation using ModelSim.