Lattice Radiant Software Tutorial with CrossLink-NX

December 17, 2019

Copyright

Trademarks

Disclaimers

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.
<*Italic*>	Variables in commands, code syntax, and path names.
Ctrl+L	Press the two keys at the same time.
`Courier`	Code examples. Messages, reports, and prompts from the software.
`...`	Omitted material in a line of code.
`.` `.` `.`	Omitted lines in code and report examples.
`[ ]`	Optional items in syntax descriptions. In bus specifications, the brackets are required.
`( )`	Grouped items in syntax descriptions.
`{ }`	Repeatable items in syntax descriptions.
`|`	A choice between items in syntax descriptions.

Contents

Lattice Radiant Software Tutorial with CrossLink-NX 7
Learning Objectives **7**
Time to Complete This Tutorial **8**
System Requirements **8**
Accessing Online Help **8**
About the Tutorial Design **8**
About the Tutorial Data Flow **8**
Task 1: Create a New Radiant Software Project **10**
- Opening the New Project Wizard **10**
- Setting the Project Name and Location **10**
- Adding Source Files **11**
- Select a Device **12**
- Finishing the Project Setup **13**
- About the File List View **13**

Task 2: Create a PLL Using IP Catalog **15**
Task 3: Verify Functionality with Simulation **16**
Task 4: Inspect Strategy Settings **17**
Task 5: Set Timing Constraints **19**
- Creating a Clock Constraint **19**
- Creating a Delay Constraint **19**

Task 6: Process the Design **20**
- About the Process Toolbar **21**
- Processing the Design **21**

Task 7: Examine the Layout **22**
Task 8: Examine Timing Analysis Results **23**
Task 9: Analyze Power Consumption **24**
Task 10: Run Export Utility Programs **26**
Summary of Accomplishments **27**
Recommended References **27**

Revision History 29

Lattice Radiant Software Tutorial with CrossLink-NX

The next generation design tool for FPGA design, the Lattice Radiant® software is designed to address the needs of high-density FPGA designs.

This tutorial leads you through all the basic steps of designing and implementing VHDL and Verilog designs targeted to the Lattice CrossLink-NX™ device family. It shows you how to use several processes, tools, and reports from the Radiant software to import sources, run design analysis, view design hierarchy, and inspect strategy settings. The tutorial then proceeds to step through the processes of adding and editing a strategy, specifying the synthesis requirements, examining the device resources, setting timing and location assignments, and editing constraints to configure the settings to implement the design to the target device.

Note

Some of the screen captures in this tutorial may have been taken from a version of Radiant software that differs from the one you are using. There may be slight differences in the graphical user interface (GUI), but the software functions the same.

Learning Objectives

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

- Create a new Radiant software project.
- Customize IP using IP Catalog.
- Verify functionality with simulation.
- Inspect strategy settings.
- Process the design.
- Examine static timing analysis.

- Analyze power consumption.
- Run Export Utility programs.

Time to Complete This Tutorial

The time to complete this tutorial is approximately 2 hours.

System Requirements

To complete the tutorial, you need:

- The Radiant software.
- The tutorial design files. You can download these from:

  https://www.latticesemi.com/view_document?document_id=52825

  The files come in a .zip file. Extract all the files to a convenient location.

Accessing Online Help

You can find online help information on any tool included in the tutorial at any time by choosing **Help > Lattice Radiant Software Help** or **Help > *<tool name>***. The Help also provides easy access to many other information sources.

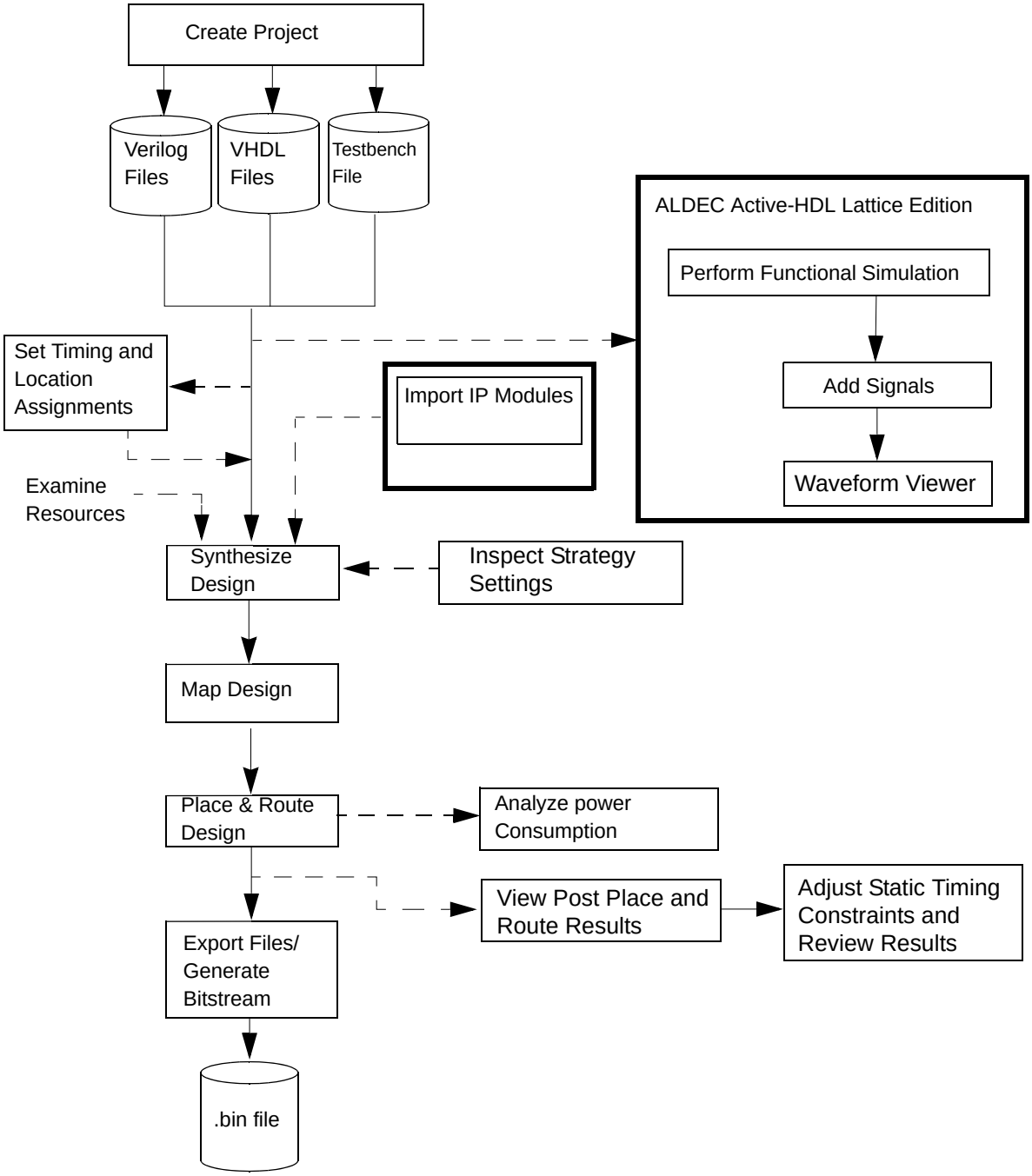
About the Tutorial Design

The design in this tutorial consists of Verilog HDL modules. The design that you create is targeted for the CrossLink-NX device.

About the Tutorial Data Flow

The following figure illustrates the tutorial data flow through the FPGA design system. You may find it helpful to refer to this diagram as you move through the tutorial tasks.

Figure 1: Tutorial Data Flow



Task 1: Create a New Radiant Software Project

A “project” is a collection of all the files and settings needed to create your design, test and analyze its behavior, and process it into a programming file for a Lattice FPGA.

Setting up a new project is done through the New Project wizard. The New Project wizard guides you through the steps of specifying a project name and location, selecting a target device, and adding existing source files to the new project. We will walk through each page of the wizard one by one. At the end, we’ll introduce the Radiant main window and its parts.

Opening the New Project Wizard

Open the Radiant software and open the New Project wizard.

To open the New Project wizard:

1. If you haven’t already, start the Radiant software by doing one of the following:
   - On Windows, go to the Start menu and choose **Lattice Radiant Software > Radiant Software**.
   - On Linux, enter the following on a command line:

     <*install_path*>/bin/lin64/**radiant**

   The main window of the Radiant software opens along with an Update dialog box. (This takes a moment.)
2. If the Update dialog box says “No update found,” click **Close**. Otherwise, install the update and restart the Radiant software.

   Now you have a clear view of the Start Page. With the Start Page you can easily open a new project, open a recent project, and access information.
3. Click the **New Project** button.

   The New Project wizard opens.
4. Click **Next**.

   The Project Name page opens.

Setting the Project Name and Location

Specify a name and location for the project files and for a design “implementation.”

An implementation is one version of your design. You can have more than one implementation to experiment with different design approaches. A project starts with one implementation. You can add more later.

To fill out the Project Name page:

1. Specify the project name: **CLNXtutorial.**

   File names for Radiant software projects and project source files must start with a letter (A-Z, a-z) and must contain only alphanumeric characters (A-Z, a-z, 0-9) and underscores (_).

2. Browse to where you want to store the project's files. This tutorial uses C:/my_radiant_tutorial. But you can use any location.

3. Make sure the **Create subdirectory** option is selected.

   The wizard automatically adds a folder for your project, which is shown immediately below the Location box.

4. Specify an implementation name. We'll use the default: impl_1.

   The directory for the implementation is automatically displayed in the Location box.

5. Click **Next**.

   The Add Source dialog box appears.

Adding Source Files

Since the tutorial comes with source files, you can add them now. Source files can be added at any time or created with the Radiant software.

To add existing source files:

1. Click **Add Source**.

   The Import File dialog box appears.

2. Browse to the tutorial design files that you downloaded. See "System Requirements" on page 8.

3. Select the following files (**Control+A** will do it.):

   - clockDivider.v
   - count32.v
   - testbench.v
   - topcount.v

4. Click **Open**.

5. Confirm that the New Project wizard is showing all of the files.

   If any files are missing, click **Add Source** again.

   If any extra files are showing, select the files and click **Remove Source**.

6. Make sure that the **Copy source to implementation source directory** option is selected.

   This makes copies of the files in your implementation instead of referring to the original files.

   The **Create empty constraint files** option is not needed for this tutorial.

7. Click **Next**.

   The Select Device dialog box appears.

Select a Device

Specify exactly which Lattice FPGA you plan to use. This selection can be changed at any time if you find a need to. We'll specify the FPGA on the CrossLink-NX Evaluation Board.

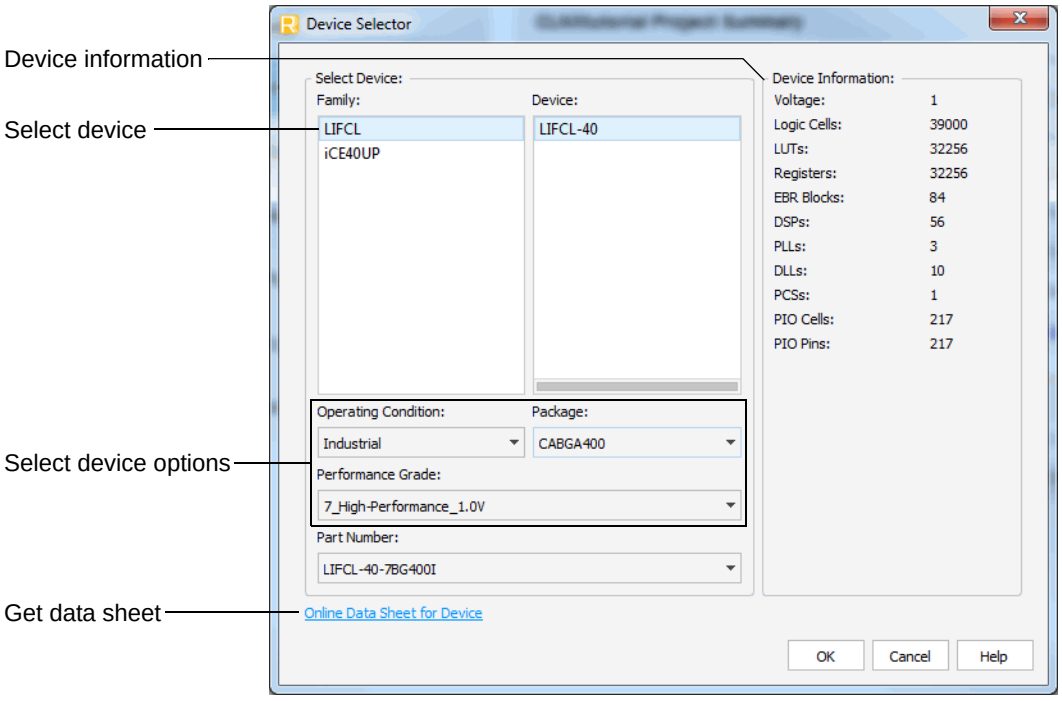
To select a device:

1. Select the device family: **LIFCL** (which is the part number for CrossLink-NX).
2. Select the specific device within the family: **LIFCL-40**.
3. Select the following device options:
   - Operating Condition: **Industrial**
   - Package: **CABGA400**
   - Performance Grade: **7_High-Performance_1.0V**

   The Part Number, at the bottom, changes as you make selections.

   The dialog box should resemble Figure 2. At the bottom is a link to get a data sheet for the device. At the right is Device Information, including a list of resources in the device such as the number of LUTs (look-up tables), registers, and PIO (programmable I/O) pins.

Figure 2: New Project Wizard's Select Device Page



4. Click **Next.**

   The Select Synthesis Tool dialog box opens.

Finishing the Project Setup

Finish by selecting a synthesis tool and confirming all the choices that you made in the New Project wizard. Then you'll see how a project looks in the Radiant main window.

To finish setting up the project:

1. Select a synthesis tool. This tutorial requires **Lattice LSE**.

   Note

   If you choose to use Synplify Pro®, some of the Radiant tools, such as Timing Constraint Editor and Physical View, will not be available. Synplify Pro may have similar tools but they are not covered in this tutorial.

2. Click **Next.**

   The Project Information dialog box appears. This dialog box summarizes the choices you made in the wizard. If you want to change any of them, click **Back**.

3. Click **Finish**.

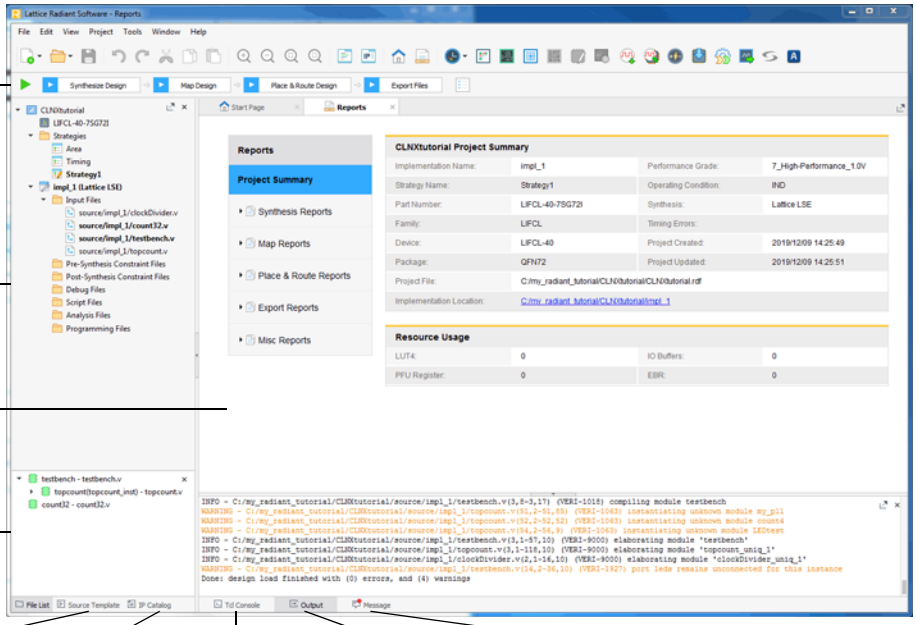
   Several views are added to the Radiant window to give you easy access to files, tools, and messages from the software. Figure 3 shows the default arrangement. On the left is the File List view showing the files and other components of the project that you just created. On the right is the Reports view showing a summary of other information about the project.

About the File List View

The File List view gives easy access to the components of the project including:

- The device.
- Strategies, which are collections of option settings for how the design is processed. Start with Strategy1, a balanced approach. If you are having trouble fitting a design into a device, try the Area strategy. If you are having trouble with timing, try the Timing strategy. You can create your own strategy by cloning one of these.
- Implementations, which are all the source files for a version of a design. A project can have several implementations so that you can experiment with different design approaches.
- Input Files, which are the design files.
- A variety of other files that may be created in the project.

Figure 3: Radiant Main Window



Bold Text Notice that some of the items, such as Strategy1 and impl_1, are written with bold text. You can have multiple components of a given type, but usually only one can be active. So impl_1 is the active implementation and Strategy1 is the active strategy for impl_1.

An exception to this rule is in the Input Files, which are the HDL design files. These are all active. In Input Files, bold text indicates a file with a top module. The Radiant software automatically analyzes the Input Files for the design hierarchy, which can be seen in the Hierarchy view. So testbench.v holds the top module in impl_1.

Commands Right-click an item to see the available commands for that item. The commands vary depending on the item. There are commands for changing properties, adding files, changing the active file, and more.

Task 2: Create a PLL Using IP Catalog

IP Catalog is an easy way to use a collection of modules from Lattice Semiconductor. With IP Catalog, these modules can be extensively customized. They can be created as part of a specific project or as a library for multiple projects.

In this task, you will generate a phase-lock loop (PLL) module to import into your design.

To customize and generate a PLL module:

1. Click the IP Catalog tab (under Source Template).

   IP Catalog replaces Source Template.

   IP Catalog comes with a large variety of architecture, arithmetic, and memory modules. These are under the IP on Local tab. Click the IP on Server tab to see more specialized modules that you can download. Take this opportunity to expand the folders and see what is available to you.

2. On the IP on Local tab, Expand Module > Architecture_Modules and hover over **PLL**.

   To the right, a blue circle with a question mark appears.

3. Click the blue circle.

   A brief description of the module appears in the tool area. To get more information about this module, click **User Guide** in the description.

4. Double-click **PLL**.

   The Module/IP Block Wizard opens.

5. For Component name, enter **my_pll**. Use the default for the Create In location.

6. Click **Next**.

   The wizard changes to a block diagram of the module and a table of properties and values. The PLL module has a few tabs covering different types of properties.

7. As you can see, there are a lot of ways that you can customize this module, but we will keep it simple. In the General tab, find the “CLKI: Frequency” property. (It’s at the top.) Click in the Value cell and enter **50**. Leave all other values as default.

8. Click **Calculate**.

   After a moment a box opens in the wizard showing analog parameters. You do not need to do anything with these.

9. Click **Generate**.

   The Check Generating Result page appears. This may take a moment.

10. Ensure that **Insert to project**, in the lower-left corner, is selected and click **Finish.**

11. Go back to the File List view to see that my_pll/my_pll.ipx has been added to the list of Input Files. The module comes with a few associated files.

Task 3: Verify Functionality with Simulation

The Radiant software provides an interface to create a new simulation project file that can be imported into a stand-alone simulator. The Radiant software can export Active-HDL and ModelSim® simulation files.

Aldec® Active-HDL™ is an integrated environment designed for simulation of VHDL, Verilog/SystemVerilog, EDIF, and SystemC designs.

In this task, you will simulate the design using Active-HDL and analyze the resulting waveforms.

To simulate the design:

1. Make sure all the source files are included in the simulation. In the File List view, under Input Files, right-click on all of the source files except for testbench.v and choose **Include For > Synthesis and Simulation**.
2. In the File List view, right-click **testbench.v** and choose **Include for** > **Simulation**. You do not want the testbench when you synthesize the design later.

   You will see activity in the Output view, at the bottom of the window, as the Radiant software re-analyzes the design hierarchy. In the File List view, topcount.v is bolded to show that it holds the top module. The Hierarchy view, which is underneath the File List view, also changes.
3. Choose **Tools** > **Simulation Wizard**.

   The Simulation Wizard dialog box appears.
4. Click **Next.**

   The Simulator Project Name dialog box appears.
5. Enter the Project name: **simulationfile**.

   Leave the other settings at their defaults.
6. Click **Next**.
7. If you left the default for the project location, a dialog box opens saying, “simulationfile does not exist. Do you want to create it?” Click **Yes**. This creates a simulationfile folder.

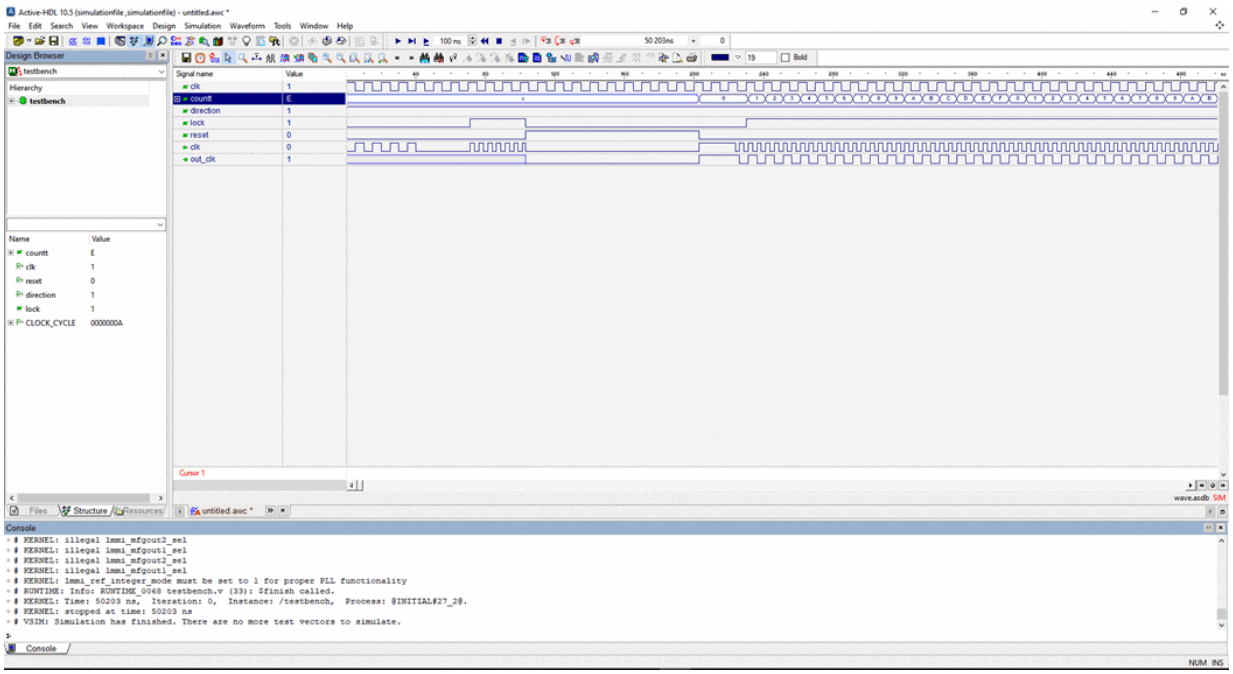
   The Add and Reorder Source dialog box appears.
8. Make sure all source files are present in the Source Files list. Leave the **Automatically set simulation compilation file order** option selected.
9. Click **Next**.

   The “Parse HDL files for simulation” dialog box appears.
10. Click **Next.**

    The Summary dialog box appears.
11. Make sure that the **Run simulator**, **Add top-level signals to waveform display**, and **Run simulation** options are all selected.
12. Click **Finish**.

    The Aldec Active-HDL software launches and the simulation starts automatically. After completing the simulation, the waveform appears, as shown in Figure 4.

Figure 4: Simulated Waveform



13. In Active-HDL, click **Simulation > Run** or click ▶ to run the simulator for some more time. Simulate the design until it stops automatically.
14. In Active-HDL, use the zoom tools (**View > Zoom**) to see how the waveform is changing.
15. Choose **File > Exit** to close Active-HDL.

    The Design Browser dialog box appears.
16. Click **OK** to stop the simulation.

    The Save File dialog box appears.
17. Click **No**.

Task 4: Inspect Strategy Settings

Implementations define the design structural elements for a project, including source code, constraint files, and debug insertion. Implementations contain all source files, constraint files, debug files, scripts, and analysis files.

A strategy is a collection of settings for controlling the different stages of the implementation process (synthesis, map, place and route, and so on). Strategies can control whether the design is optimized for area or speed, how long place and route takes, and many other factors. The Radiant software provides a default strategy, which may be a good collection to start with, and some variations that you can try. You can modify Strategy1, and create other strategies to experiment with or to use in different circumstances. Predefined strategies can also be cloned and then modified.

To view synthesis settings:

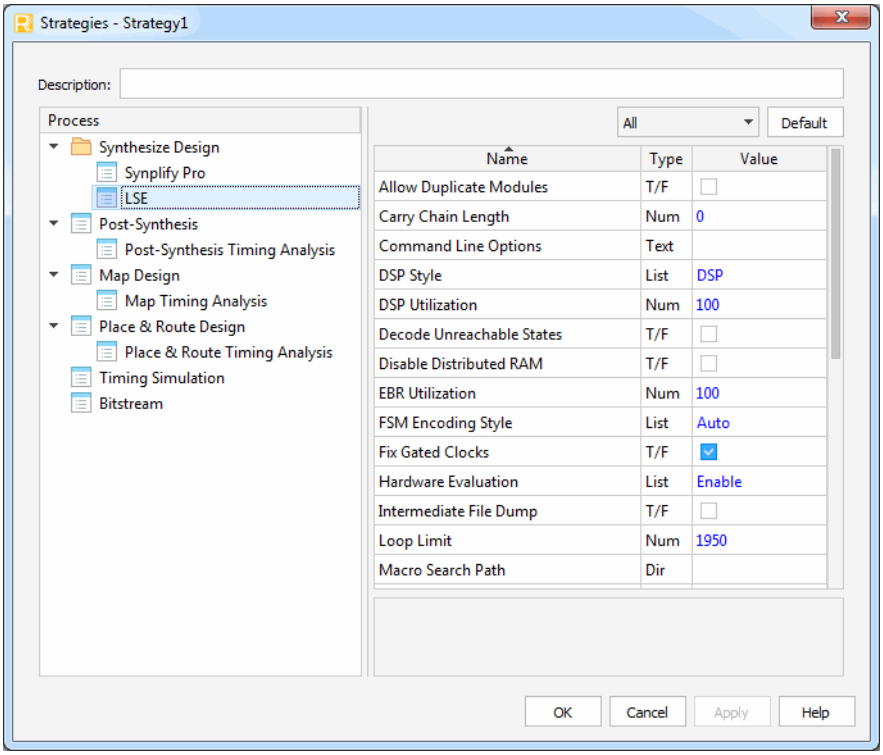
1. From the File List view, double-click **Strategy1**.

   The Strategies - Strategy1 dialog box**,** shown in Figure 5**,** appears.

2. Click **Synthesize Design > LSE**.

   A set of default global synthesis timing constraints and optimization settings appears in the panel. LSE settings are displayed as the default in the dialog box.

Figure 5: Strategies - Strategy 1



Note

When each option is selected, descriptive text appears in the lower panel of the dialog box. Default values in the strategies dialog box are shown in blue while changed values are shown in black.

3. Click **Cancel**.

Task 5: Set Timing Constraints

Timing and location assignments constrain logic synthesis, as well as backend map, place, and route programs to help meet your design requirements. A well-constrained design helps optimization algorithms work as efficiently as possible.

In this task you'll set a clock and input delay.

Creating a Clock Constraint

To create a clock constraint:

1. Choose **Tools > Timing Constraints Editor > Pre-Synthesis Timing Constraint Editor**.

   The Pre-Synthesis Timing Constraint Editor appears. The top half is an empty spreadsheet with a row of tabs beneath it. The bottom half is a box where constraint text will appear.

2. If you see a yellow bar with a message saying the “Design database in memory is outdated,” click **Reset Database**, which is to the right of the message.

3. Click the **Clock** tab.

4. Double-click in the cell in the Object Clock column.

   Some text appears in the cell and, at the right side, three periods.

5. Ignore the text and click on the three periods.

   The Object Edit dialog box opens.

6. From the Object Type menu, choose **CLOCKPORT**.

7. Under Available Objects, select **clk**.

   In the Object Command box, “get_ports clk” appears.

8. Click **OK**.

   The dialog box closes. The above command appears in the Object Clock column.

9. Click in the **Frequency** column and enter **50**.

   Other columns of the row are filled in. In the lower half of the editor, a create_clock constraint appears.

Creating a Delay Constraint

To create an input delay:

1. Click the **Input/Output Delay** tab.

2. Double-click in the Constraint Type column and choose **set_input_delay**.

3. Double-click in the Port column and click on the three periods.

   The Object Edit dialog box opens.

4. From the Object Type menu, choose **INPUTPORT**.

5. Under Available Objects, select **direction**.

In the Object Command box, “get_ports direction” appears.

6. Click **OK**.

   The dialog box closes. The above command appears in the Port column.

7. Tab to the Clock column and click on the three periods.

   The Object Edit dialog box opens.

8. From the Object Type menu, choose **CLOCK**.
9. Under Available Objects, select **clk**.

   In the Object Command box, “get_ports clk” appears.

10. Click **OK**.

    The dialog box closes. The above command appears in the Clock column.

11. Click in the Add Delay column.

    A check mark appears.

12. Click in the Value cell and enter **2**.

    In the lower half of the editor, you should see the following constraints:

    ```
    create_clock -name {clk} -period 20 [get_ports clk]
    set_input_delay -clock [get_clocks clk] -add_delay 2 [get_ports direction]
    ```

13. Choose **File > Save As**.
14. In the Save As dialog box, stay in the default folder (CLNXtutorial) and enter the file name, **timing_constraints**.
15. Click **Save**.

    In the File List view, timing_constraints.ldc appears under the Pre-Synthesis Constraint Files folder.

16. Close the Pre-Synthesis Timing Constraint Editor.

Task 6: Process the Design

Processing a design involves several steps that convert the high-level Verilog and VHDL description into code that can actually program a specific FPGA:

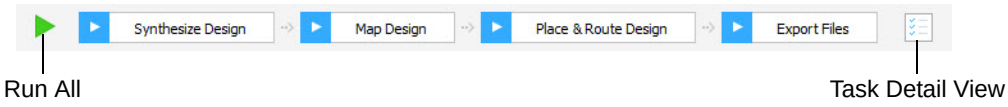
- Synthesis converts HDL into a gate-level netlist that is optimized for the FPGA.
- Map converts the netlist into a network of device-specific components, such as PFU (programmable function units) and I/O buffers.
- Place and route converts the mapped network into specific components and signal routes within the device.
- Export converts the place-and-route specifications into code to program the FPGA.

Each step also produces a set of reports that describe how the process was run and the results. If a process fails, its reports are the place to start troubleshooting.

About the Process Toolbar

Use the Process Toolbar (shown below) to run the processes.

Figure 6: Process Toolbar



With a single click you can run any individual process including any preceding processes that have not been run yet. Click the Run All button to run the whole sequence. Right-click a process button to get a menu of options for running the process.

Click the Task Detail View button to select other files to generate while running the processes. Timing analysis and simulation files are available.

While a process is running, the Run All button changes to the Stop button. Click the Stop button to stop the processing.

When a process completes, its button shows its success or failure with a green check mark or a red X.

Processing the Design

In this task, you will step through the first three processes one-by-one and check the reports after each. Then you'll examine the results of place and route before running the final Export Files process. However, in normal practice, you would probably run the whole sequence and then check the results.

To process the design:

1. In the Process Toolbar, click **Synthesize Design**.

   Task Detail View opens and tracks completion of the processes.

2. In the Reports view, click **Synthesis Reports**. These reports give details of how synthesis ran. They also give detailed information about use of device resources and timing. Hover over the Contents button in the top-right corner to get links to different sections of a report.
3. When you finish looking at the synthesis reports, click **Map Design**.
4. In the Reports view, click **Map Reports** and examine the available reports.
5. When you finish looking at the map reports, click **Place & Route Design**.
6. In the Reports view, click **Place & Route Reports** and examine the available reports.

Task 7: Examine the Layout

After routing, you can see a display of the layout using the Physical View and Floorplan View tools and cross-probing between the two views.

To see the layout:

1. Choose **Tools >** **Physical View**.

   Physical View provides a read-only detailed layout of your design that includes switch boxes and physical wire connections.

2. To the left of the diagram are lists of ports, instances, and nets. Expand the Instances list and choose one of the primitives, such as Instances > clockDivider_inst > Primitives > **out_clk.ff_inst**.

   The display zooms to the component.

3. Right-click on the component and choose **Show in > Floorplan View**.

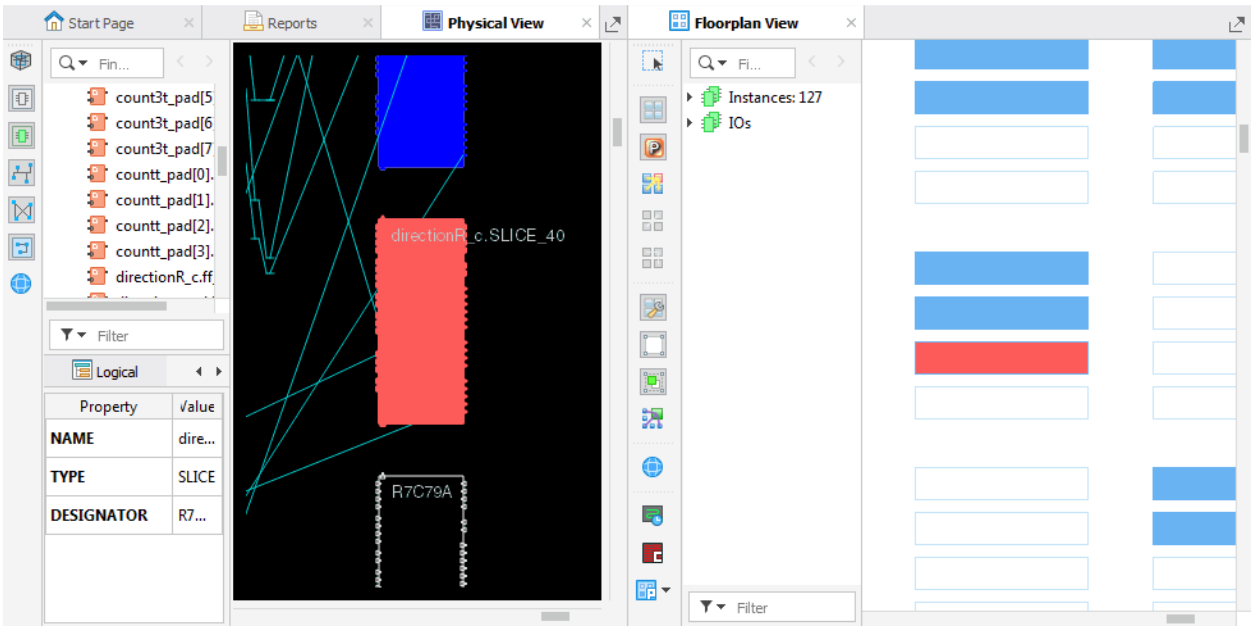
   Floorplan View opens with the display zoomed to the same component. Floorplan View provides a large-component layout of your design.

4. To easily see cross-probing between Floorplan View and Physical View, ensure both views are attached to the Radiant main window. Right-click on the Floorplan View tab and choose **Split Tab Group**.

   The pane splits to show both views. See Figure 7.

   When both Floorplan View and Physical View are open, an item that you select in one of these views is automatically selected in the other. Cross-probing is especially useful for immediate examination of connections in both views.

Figure 7: Cross-Probing



5. Right-click on the Floorplan View tab and choose **Move to Another Tab Group**.

Floorplan View fills the whole pane as before.

6. In the toolbar of Floorplan View, click the arrow of the Placement button and choose **IO**.

   Floorplan View changes to show the I/O of the device.

7. In the list, expand Instances and click **clk_pad.bb_inst**.

   Floorplan View zooms in to the I/O for clk. You can do this for any of the instances labeled with "_pad" and for any of the items in the IOs list.

8. Close Floorplan View and Physical View.

Task 8: Examine Timing Analysis Results

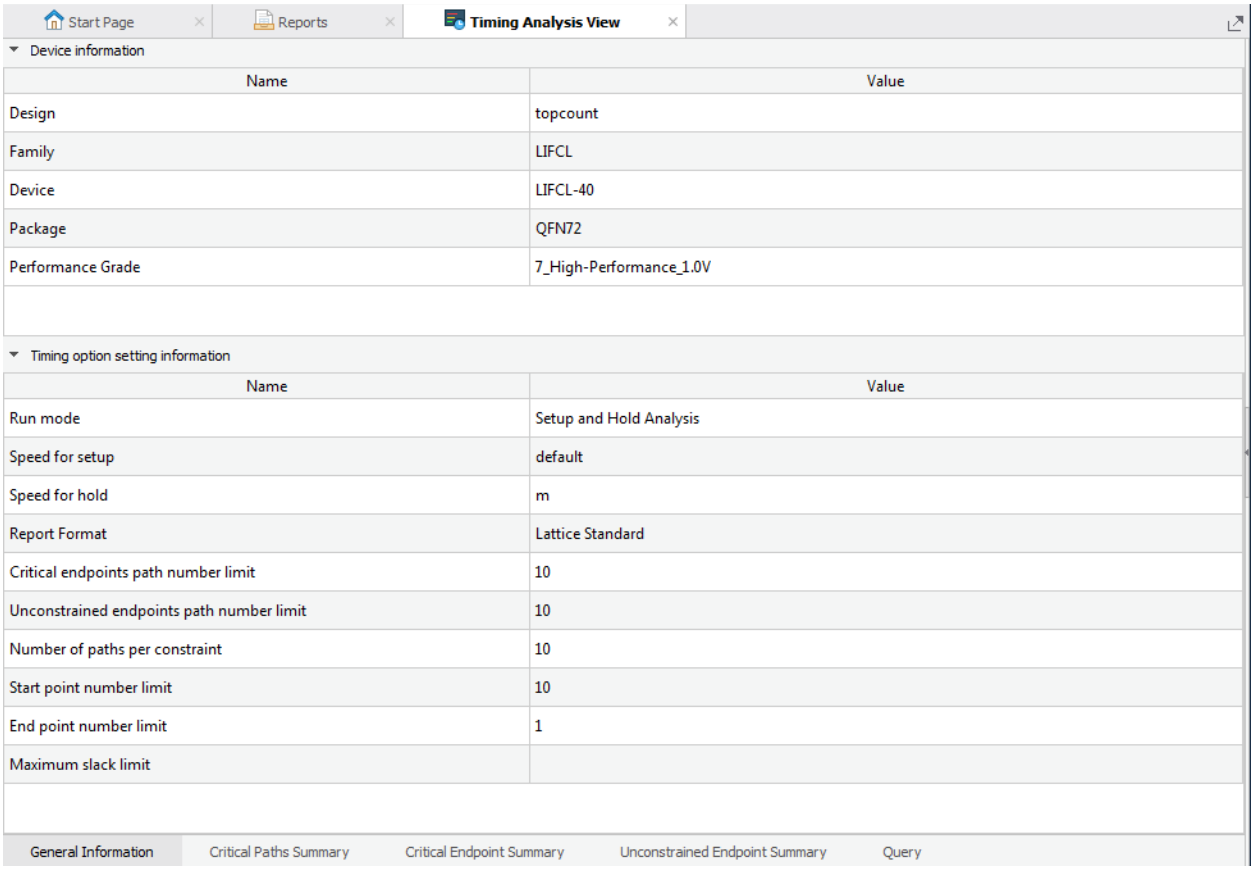
Static timing analysis can determine if your circuit design meets timing constraints. Rather than simulation, it employs conservative modeling of gate and interconnect delays that reflect different ranges of operating conditions on various dies, providing complete verification coverage.

To examine timing analysis results:

1. Choose **Tools > Timing Analysis View**.

   Timing Analysis View appears as shown in Figure 8.

Figure 8: Timing Analysis View



2. Click the **Critical Paths Summary** tab (on the lower-left of Timing Analysis View).
3. Select Row 1 of the Paths summary.

   The Path Detail tab in the right pane appears with details.
4. Close Timing Analysis View.

Task 9: Analyze Power Consumption

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

To analyze power consumption:

1. Choose **Tools > Power Calculator**.

   Power Calculator opens in Calculation mode as shown in Figure 9.

   Power Calculator provides two modes for reporting power consumption:

   - Estimation Mode:

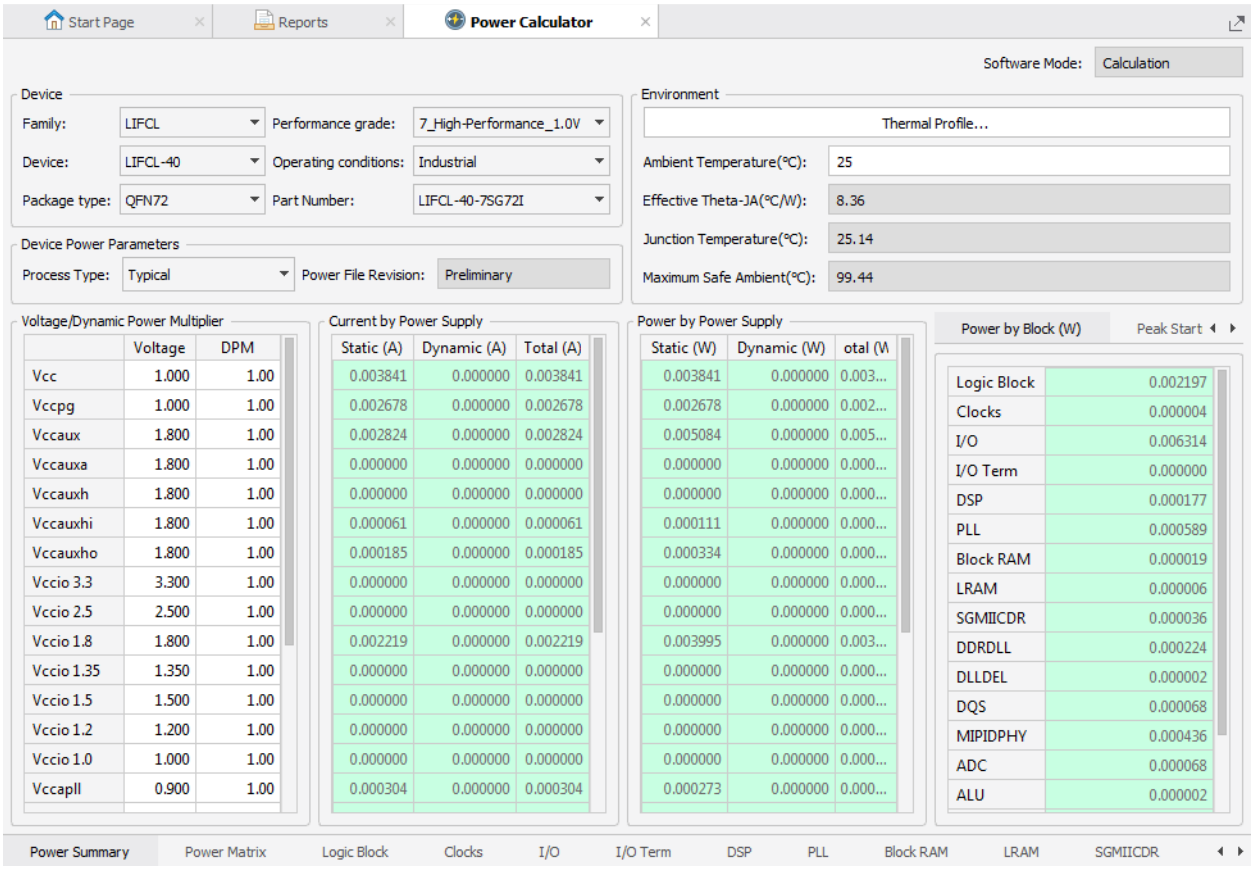
In estimation mode, Power Calculator provides estimates of power consumption based on the device resources or template that you provide. This mode enables you to estimate the power consumption for your design before the design is complete or even started.

- Calculation Mode:

  In calculation mode, Power Calculator calculates power consumption on the basis of device resources taken from a design's .udb file, or from an external file such as a value change dump (.vcd) file, after placement and routing. This mode is intended for accurate calculation of power consumption, because it is based on the actual device utilization.

  Editing data in white cells, such as voltage, frequency, activity factor, and thermal data, does not change the mode. Editing data in yellow cells, such as design data, will change calculation mode to estimation mode.

Figure 9: Power Calculator



2. In the Device Power Parameters section select the following parameter:

   Process Type: **Worst**

3. Click the **Thermal Profile...** button in the Environment section.

   The Power Calculator – Thermal Profile dialog box appears.

4. In the Board Selection section, select **Small board**.

5. Click **OK**.

   After a short while the new temperature results become available in the Environment section.

6. Close Power Calculator.

   A Save dialog box appears.

7. Click **No**.

Task 10: Run Export Utility Programs

Use the Process Toolbar to generate files for exporting. One of the files exported will be a bitstream file (.bit) that can be used to program an actual CrossLink-NX device on a circuit board.

To generate files for export:

1. In the Process Toolbar, click the Task Detail View  button.
2. Under **Export Files**, ensure the following are selected:
   - IBIS Model
   - Gate-Level Simulation File
3. Click **Export Files**.

   The Radiant software generates the selected files and saves them in the directory of the implementation.

4. In the Reports view, click **Export Reports** and examine the available reports.

   At the end of the Bitstream report is the pathname of the bitstream file.

5. In the File List view, right-click on impl_1 and choose **Open Containing Folder**.
6. Look for a file named **LEDtest_impl_1.bit**.

To close the project:

The tutorial is done and you can close the project.

1. To close the design project, choose **File > Close Project**.

   The Save Modified Files dialog box opens.

2. To save the files, click **OK**. To not save the changes, click **Deselect All** and then click **OK**.

   The design project and associated tools close. The Radiant window returns to the Start Page.

Summary of Accomplishments

You have completed the *Lattice Radiant Software Tutorial with CrossLink-NX*. In this tutorial, you have learned how to:

- Create a new Radiant software project.
- Customize IP using IP Catalog.
- Verify functionality with simulation.
- Inspect strategy settings.
- Process the design.
- Examine static timing analysis.
- Analyze power consumption.
- Run Export Utility programs.

Recommended References

You can find additional information on the subjects covered by this tutorial in the Radiant software online Help and in the *Lattice Radiant Software User Guide*.

Revision History

The following table gives the revision history for this document.

Date	Version	Description
12/17/2019	2.0	Added a link for downloading the design files. Expanded Task 1 with more information about the main window and the File List view.
11/12/2019	2.0	Initial Release.