Lattice Diamond Tutorial

June 2012

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 Diamond Tutorial

The next generation design tool for FPGA design, Lattice Diamond, is designed to address the needs of high-density FPGA designers.

This tutorial leads you through all the basic steps of designing and implementing a mixed VHDL, Verilog, and Edif design targeted to the Lattice ECP3 device family. It shows you how to use several processes, tools, and reports from the Lattice Diamond 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 preferences to configure the filter to implement the design to the target device.

Learning Objectives

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

- Set up a mixed VHDL, Verilog, and EDIF project
- View and Analyze the design
- Inspect Strategy Settings
- Examine Design Resources
- Set Timing and Location Assignments
- Place and Route
- Create an Implementation
- Set an Active Implementation
- Compare Multiple Place and Route Runs
- Examine Post Place and Route Results

Time to Complete This Tutorial

The time to complete this tutorial is about 60 minutes.

System Requirements

The following software is required to complete the tutorial:

- Lattice Diamond software
- (Optional) LatticeECP3 Versa Development Kit

Accessing Online Help

You can find online help information on any tool included in the tutorial at any time by choosing **Help > Lattice Diamond Help**.

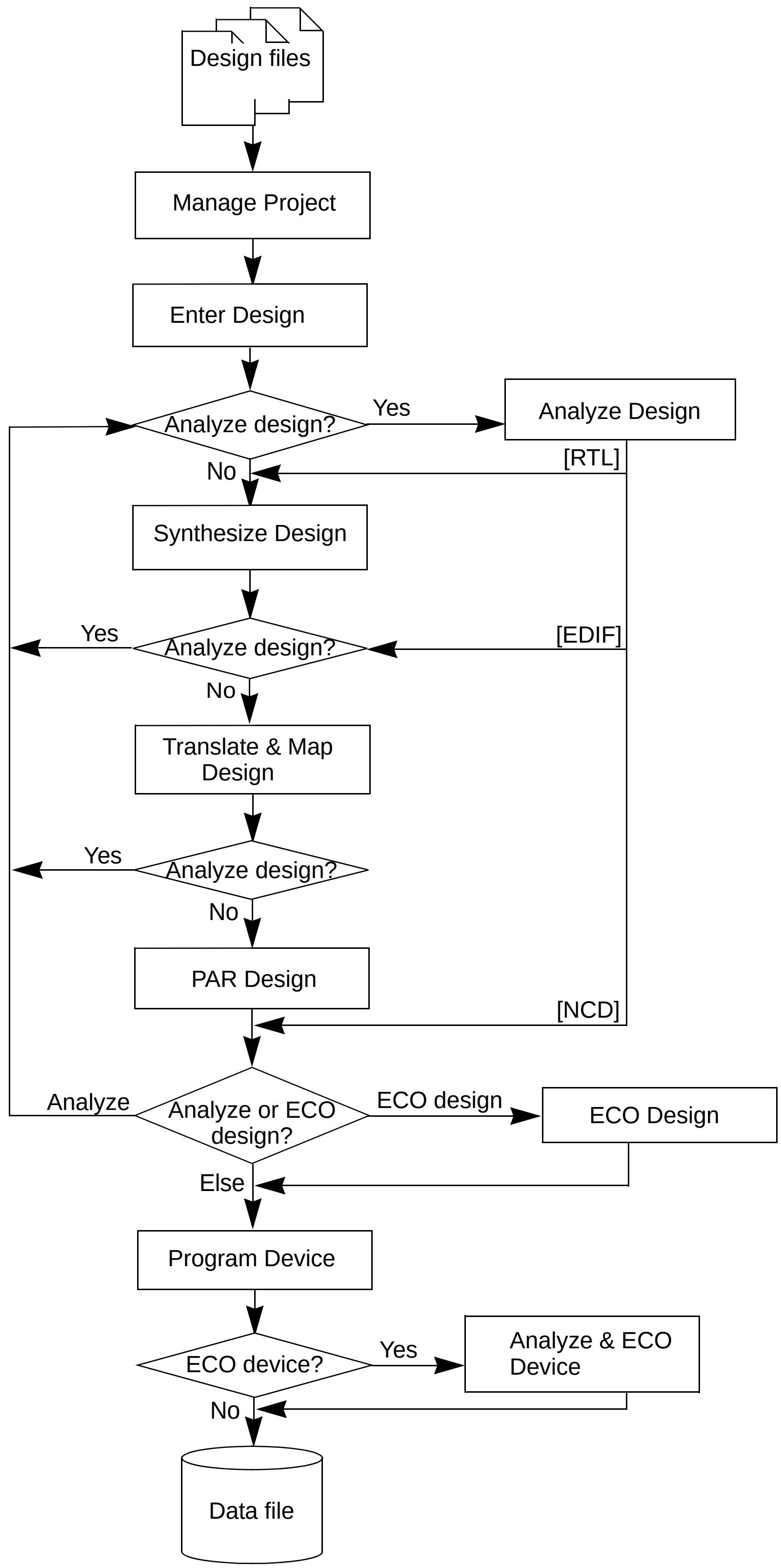
About the Tutorial Design

The design in this tutorial consists of a Verilog HDL module, two VHDL module and one EDIF module. The design that you create is targeted to Lattice ECP3 device families.

About the Tutorial Data Flow

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

Tutorial Data Flow



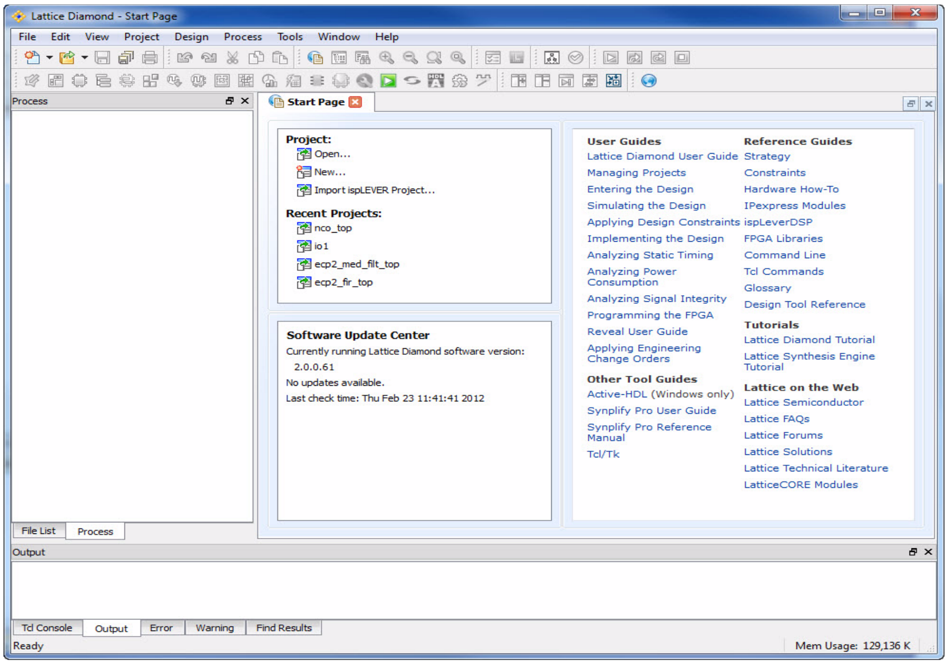
Task 1: Create a New Project

Projects are used to manage input files, preferences, and optimization options related to an FPGA implementation. While there are a number of tasks you can perform independent of a project, most designs start with creating a new project.

To create a new project:

1. On Windows, select the Lattice Diamond icon or **Start > Programs > Lattice Diamond 2.0 > Lattice Diamond**.

   The Lattice Diamond Design Environment appears.



   The initial layout provides a Start Page which provides a list of common Project actions like **Open...** to open a pre-existing project and **New....** to run the New Project Wizard. Hyperlinks in the right pane of the Start Page provide access to user guides, reference material, and online resources available from www.latticesemi.com.

   Note

   Several design entry and analysis features of Lattice Diamond are available without a source file as part of the project, for example, you may wish to define and generate an IP core or a microprocessor platform using the Diamond interface and use the result later in one or more projects. Also the power analysis features in Diamond do not require source files to perform estimation.

2. From the Start Page, click **Project > New**, or from the Diamond main window choose **File > New > Project**. You can also click the **New** icon from the toolbar and then choose **Project**  .

   The New Project overview dialog box appears.

3. Click **Next**. The **New Project** dialog appears.

4. Specify Project name: **mixedcounter**

   Note

   File names for Diamond 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 (_).

5. Click **Browse...** to specify a directory on your local PC other than the Diamond installation directory, for example, <*drive:\my_diamond_tutorial*>.

   Note

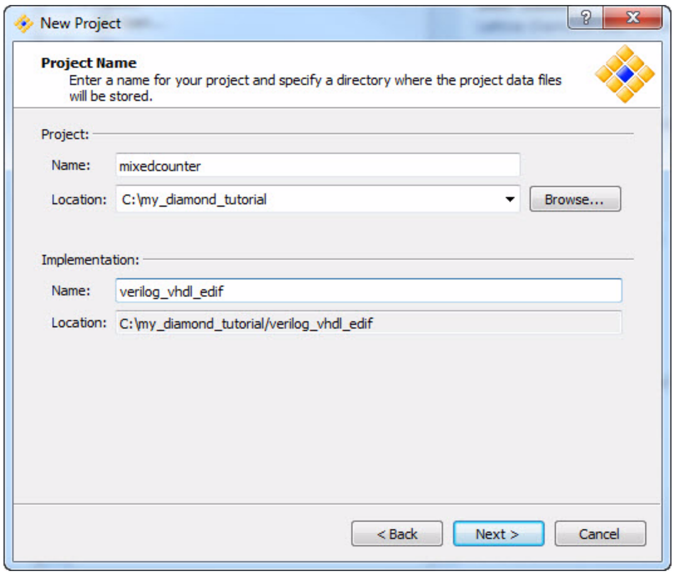
   You will have to store your tutorial files in a directory other than your Lattice Diamond installation directory. You do not have the write permission to the Diamond installation directory.

6. Enter an implementation name. For this tutorial, enter **verilog_vhdl_edif**.

   Implementations are comprised of source and constraint files. You may create or add multiple Implementations per project when you want to compare multiple place and route runs.

   By default, when you specify the project name, the implementation name is simultaneously specified the same. For this tutorial, change the implementation name to **verilog_vhdl_edif**. The directory to store the implementation will be automatically displayed in the Location area.

   We will talk about creating a new implementation later in this tutorial.



7. Click **Next >**. The Add Source dialog appears.
8. Click **Add Source...** The **Import File** dialog appears.
9. Select the folder where the source files are located <*drive:\diamond_install_directory\version#*\examples\mixedcounter\source>. Choose count8.edn, count16.vhd, topcount.v, typepackge.vhd in the

directory and click **Open**. The Add Source step of the Wizard appear with all the selected source added.

10. Enable **Copy source to implementation directory** and click **Next**. The Select Device dialog appears.

11. Select the following device options:

    Family: **LatticeECP3**

    Device: **LFE3-35EA**

    Performance Grade: **8**

    Package type: **FPBGA484**

    Operating Conditions: **Commercial**

    Part Names: **LFE3-35EA-8FN484C**

    Click **Next**.

    The **Project Information** dialog appears. The project information including project name, location, implementation name, device, etc. are listed.

12. Click **Finish**.

    The File List view is populated with the Process view and the Reports view.

Note

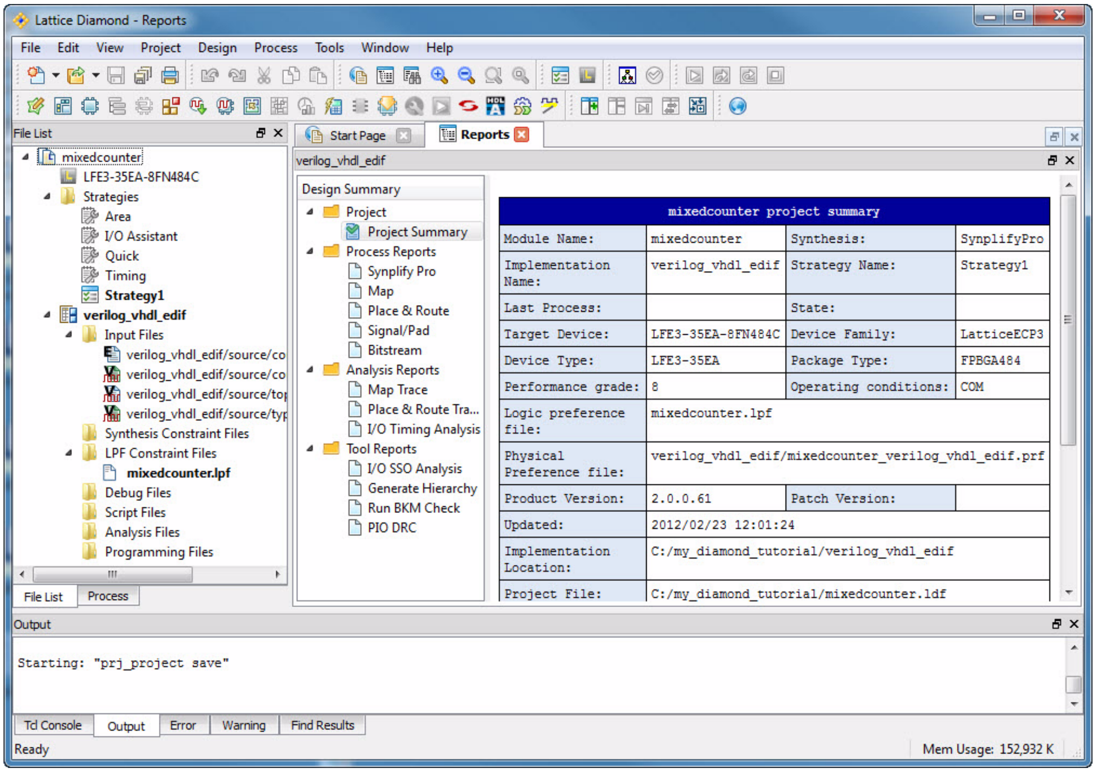
- If you have run **Design > Auto Generate Hierarchy** and **Design > Auto Run BKM Check** in a Lattice Diamond session before starting this tutorial, you will also see the Module Library view, Dictionary view, Hierarchy view displayed beside the Process view. And the Design view of the HDL Diagram will be opened to the right of the Reports view.
- The File List view, Process view, Module Library view, Dictionary view, and Hierarchy view are dockable. You can drag and drop them anywhere in the Diamond main window or even outside the main window.

The File List view displays the components of the project.

Project name:	**mixedcounter**
Target device:	**LFE3-35EA-8FN484C**
Strategies:	**Strategy1**
Design Implementation:	**verilog_vhdl_edif**

Note

You can also see Area, I/O Assistant, Quick, Timing listed in the Strategies folder in the File List view. These are predefined strategies supplied by Lattice Semiconductor. They are designed to solve particular types of design. For details of these predefined strategies, refer to the online Help.



The imported VHDL, Verilog, and EDIF files appear in the Input Files folder in the File List view. The File List organizes project files by categories: Strategies, Implementation including Input Files, Constraint Files, Debug Files, Script Files, and Analysis Files. You may adjust file order by dragging/dropping of the filenames in the list. Properties of each file are accessed by highlighting a file, clicking the right mouse button, and selecting Properties from the pop up menu.

The Process view lists all the processes available, such as synthesize design, translate design, map design, place and route design, and export files.

The Reports view provides a way to examine and print process reports. Reports view displays reports for the major processes. There are two panes in the Reports view. The left pane lists the design summary information including the reports types. The reports in detail are displayed in the right pane. You can navigate the reports quickly by using the Find function.

Section	Description
Project Summary	Lists the summary information of the project including module name, synthesis tool chosen, implementation name, strategy name, target device, device family, device type, package type, performance grade, operating conditions, logic preference file, software product version, project file name, and location, etc.

Section	Description
Process Reports	Lists the synthesis, map, place and route, signal/pad, bitstream reports in HTML format.
Analysis Reports	Lists the map trace, place and route trace, I/O timing analysis, IO SSO analysis reports.

Log messages are displayed in the Output frame of the Diamond main window.

Next you will learn how to explore and analyze the HDL design.

Task 2: Running Analysis Tools

Diamond provides an HDL visualization and rule-checks to detect coding style violations that may lead to pre-/post-synthesis simulation mismatches.

To analyze and view the HDL design:

1. Choose **Design > Generate Hierarchy** or click the Generate Hierarchy icon .

   The file list is scanned and a syntax check is performed. The Module Library, Dictionary, Hierarchy views, as well as the HDL Diagram appear in the Diamond main window and are populated with details about the VHDL modules and other symbols like signals and ports of the HDL design.

   The Module Library view is a standard tree-list view that shows all the modules in the loaded design, used and unused. You can open the entries in this view to browse the complete port, signal, behavioral block, and instance hierarchies within each module of the loaded design.

   The Dictionary view shows a sorted list of all the elements in the design. Within the list, design elements are grouped by name instead of by type. This view is very useful for finding design elements.

   The Hierarchy view is a mirror of the current view of the HDL Diagram and presents Ports, Signals, Instances, and Continuous Assignments for the current level of hierarchy.

   Note

   By default, only modules and instances are displayed in the Module Library view, Hierarchy view, and the Dictionary view. If you want to have a full display of the hierarchy of the design, turn off the **Simplified Hierarchy Display** option from the Options dialog (**Tools > Options** from the Diamond main window) the HDL Diagram section.

   The Design view shows a graphical display of the HDL module hierarchy of the design.

   Details about the scan appear in the Output frame.

2. Choose **Design > Run BKM Check**. Best Known Methods (BKM) analysis is run.

   Best Known Methods (BKM) are design guidelines that HDL Diagram uses to analyze your design. BKM checks include the following:

   Connectivity – Checks the pin connectivity of instances throughout the design.

   Synthesis – Checks for violations of the Sunburst Design coding styles, as well as other potential synthesis problems.

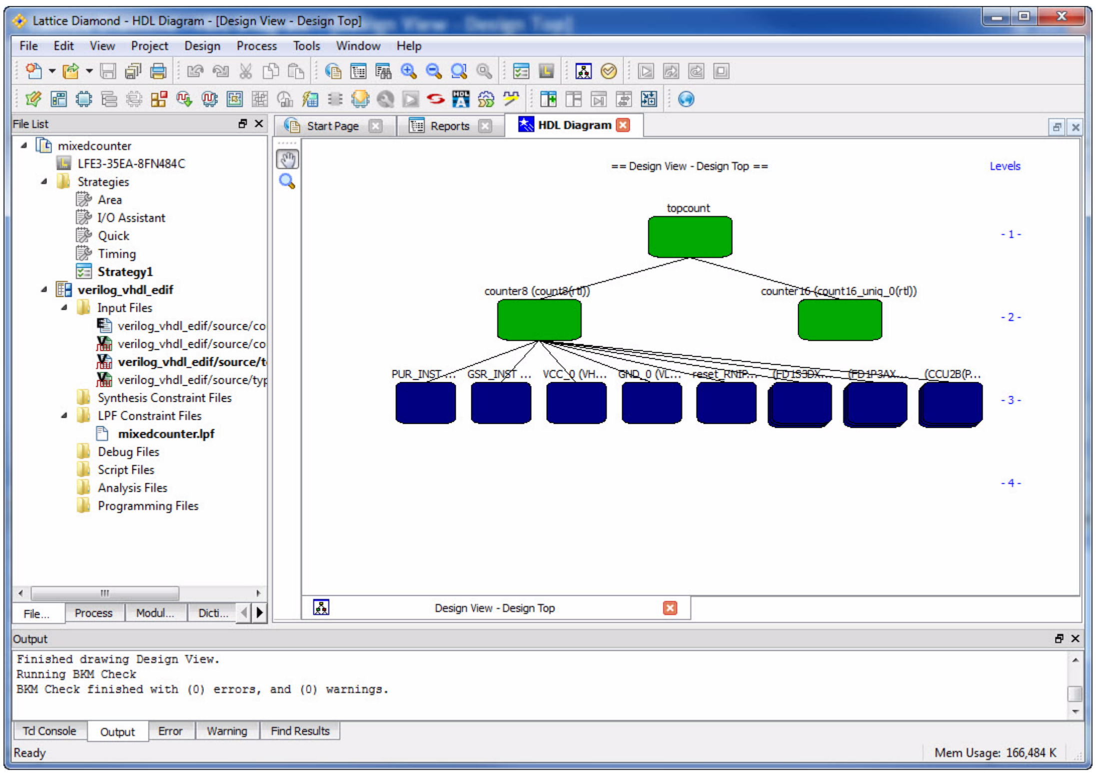
   Structural Fan-Out – Checks for maximum structural fan-out violations.

   Coding Styles – Colors modules based on their line count, colors pins and ports based on their width, validates module names, and also performs big-endian or little-endian checks on all ports.

   Verification – Validates the existence and timestamps of VCD files. A series of Lint-like RTL rule checks are run. Modules that have rule violations are color coded in the HDL Diagram view.

   The checks performed during a BKM run can be customized in the **Options** dialog (**Tools > Options** from the Diamond main window) HDL Diagram section.

   It is a good practice to run RTL analysis before synthesis to detect coding style that could lead to mismatches between pre-synthesis and post-synthesis simulation results. The analysis views are also excellent documentation output for your design.

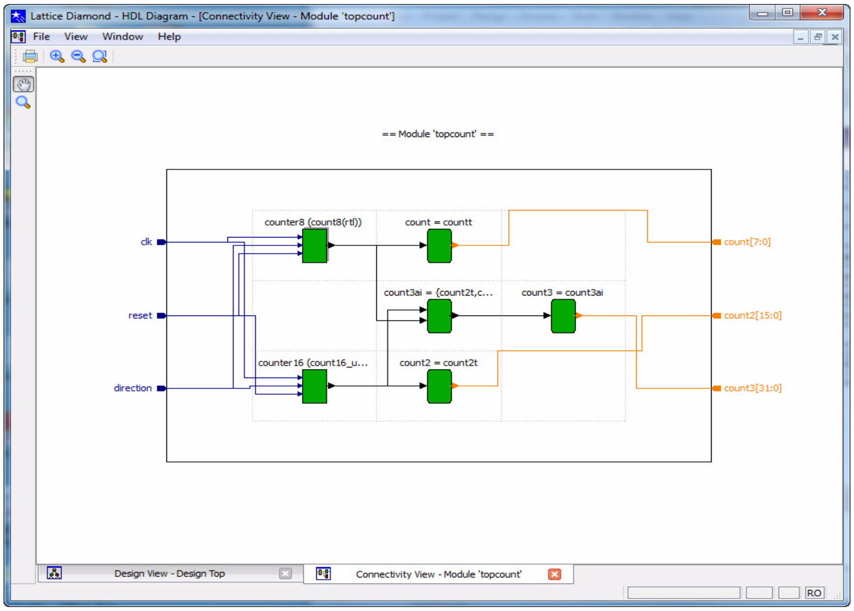


3. After running the BKM check, you might encounter warning and error messages. Error and warning messages are displayed in the Output,

Warning, and Error frames. In the Output, Error, or Warning frames, if you double click the message, the very source indicated in the message will be opened in the associated editor. This cross-probing function can ease your check of the source file.

For this tutorial, there is no error and warning after running the BKM check.

4. Double click the **topcount** block from the HDL Diagram Design view, or right-click the **topcount** block and choose **View Connectivity**. The Connectivity view appears as a new tab. You can click the Detach Tool icon on the upper right corner the HDL Diagram to make it a separate window. The Connectivity view shows signal flow between module ports, internal instances and the behavioral blocks within a particular instance or module. It enables you to explore the signal connectivity—signals and bundles—between instances and behavioral blocks within the current module.



5. When you finish checking the signal connectivity, you can choose **Window > Attach Window** from the separated HDL Diagram to make it back to the Diamond main window.

6. After running the design analysis tool, you can see the top-level source **topcount** is bold-faced in the File List view.

Task 3: Inspect Strategy Settings

A **strategy** is a collection of settings for controlling the different stages of the implementation process (synthesis, map, place & 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. Diamond 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.

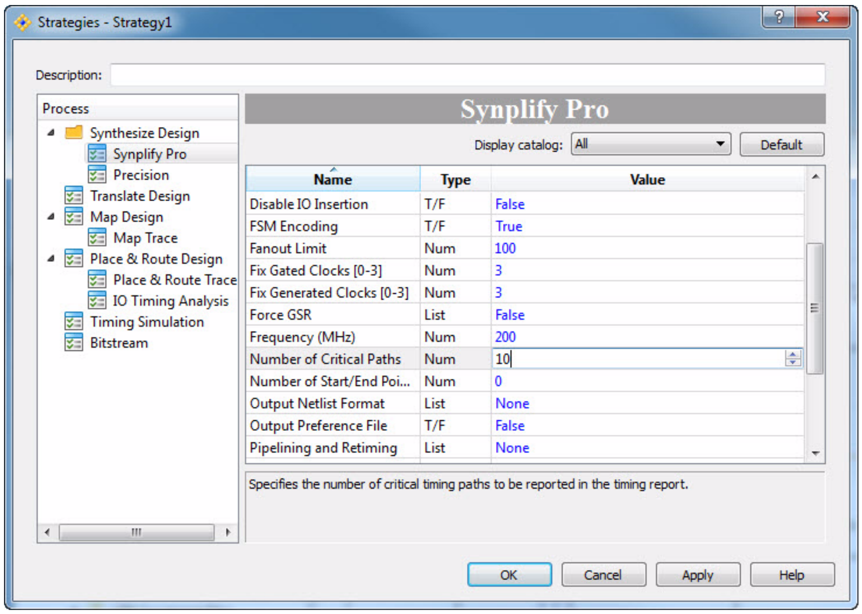
To adjust synthesis settings:

1. From the File List view, double-click **Strategy1**. The **Strategies - Strategy1** dialog appears. Browse to **Synthesize Design > Synplify Pro**. A set of default global synthesis timing constraints and optimization settings appear in the panel. Synplify Pro settings are displayed as the default in the dialog.

   For information on SDC file usage in Synplify, see the Synplify and Synplify Pro for Lattice Reference Manual in the Synplify Pro for Lattice installation directory.

2. Specify the following setting for Synplify Pro:

   Number of Critical Paths: 10



Note

When each strategy is selected, descriptive text appears in the lower panel of the dialog.

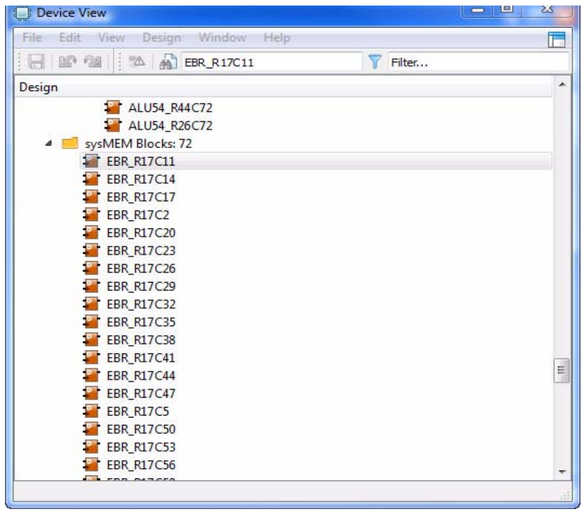
3. Choose **OK**. Global synthesis options are now set for the design.

Task 4: Examine Resources

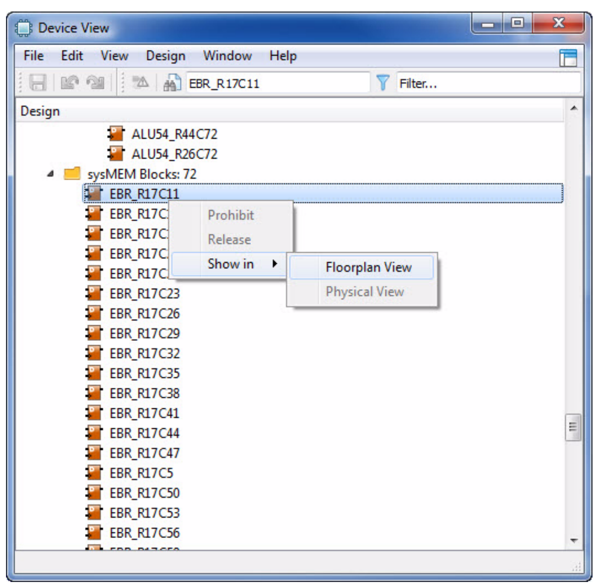
Diamond provides visualization tools to help you understand and document the physical resources of the target device and the utilization of resources. You can browse and locate device features independent of the project's source files. After synthesis, you can view the calculated resource utilization.

To browse device resources:

1. Choose **Tools > Device View**. The Device view appears. Click the Detach Tool icon on the upper right corner the Device view to make it a separate window. An index of the physical resources of the target device appear.
2. Click the icon to expand the Device folder. Several folders organized by feature type appear.
3. Expand the sysDSP Blocks and sysMEM Blocks folders.
4. Type **EBR_R17C11** into the Find entry box at the top of the Device View. The first occurrence of an EBR design symbol is highlighted.



5. Select the EBR_R17C11 in the list. Right-click EBR_R43C11 and choose **Show in > Floorplan View**.



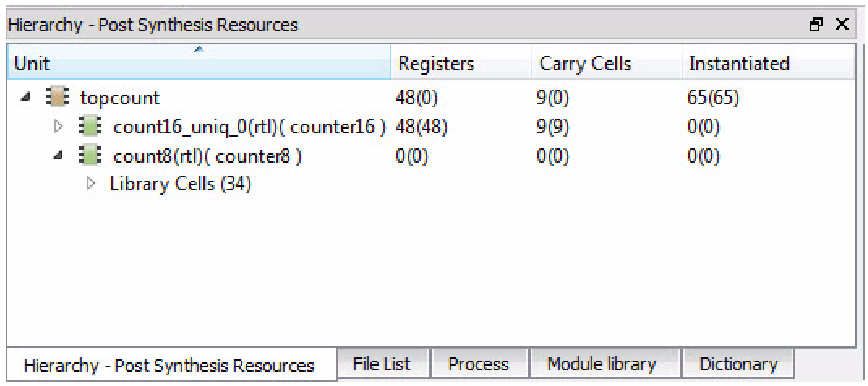
Floorplan View provides a large-component layout of your design. It displays user constraints from the logical preference file (.lpf) and placement and routing information.

6. Close the Floorplan view and the Device view.

After synthesis, you can view the calculated utilization of resources.

To synthesize the design and examine resource utilization:

1. From the Process View, double-click **Synthesize Design**.
2. When the synthesis process is complete, select **View > Show Views > Post Synthesis Resources**. Select the **Hierarchy – Post Synthesis Resources** tab.



The Post-Synthesis Hierarchy View displays the number of logical resources within each level of the design.

Task 5: Set Timing and Location Assignments

Timing and location assignments constrain logic synthesis, as well as back-end 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 section you'll set default timing constraints for the operating frequency and I/O timing then assign package pins to specific I/O signals.

To set timing and location assignments:

1. From the Process view, double-click **Translate Design** and then **Map Design**. The batch interface to logic synthesis, EDIF translation, and the design mapper run. Report files appears in the Reports view. To view each process report, select the process in the **Design Summary** pane.

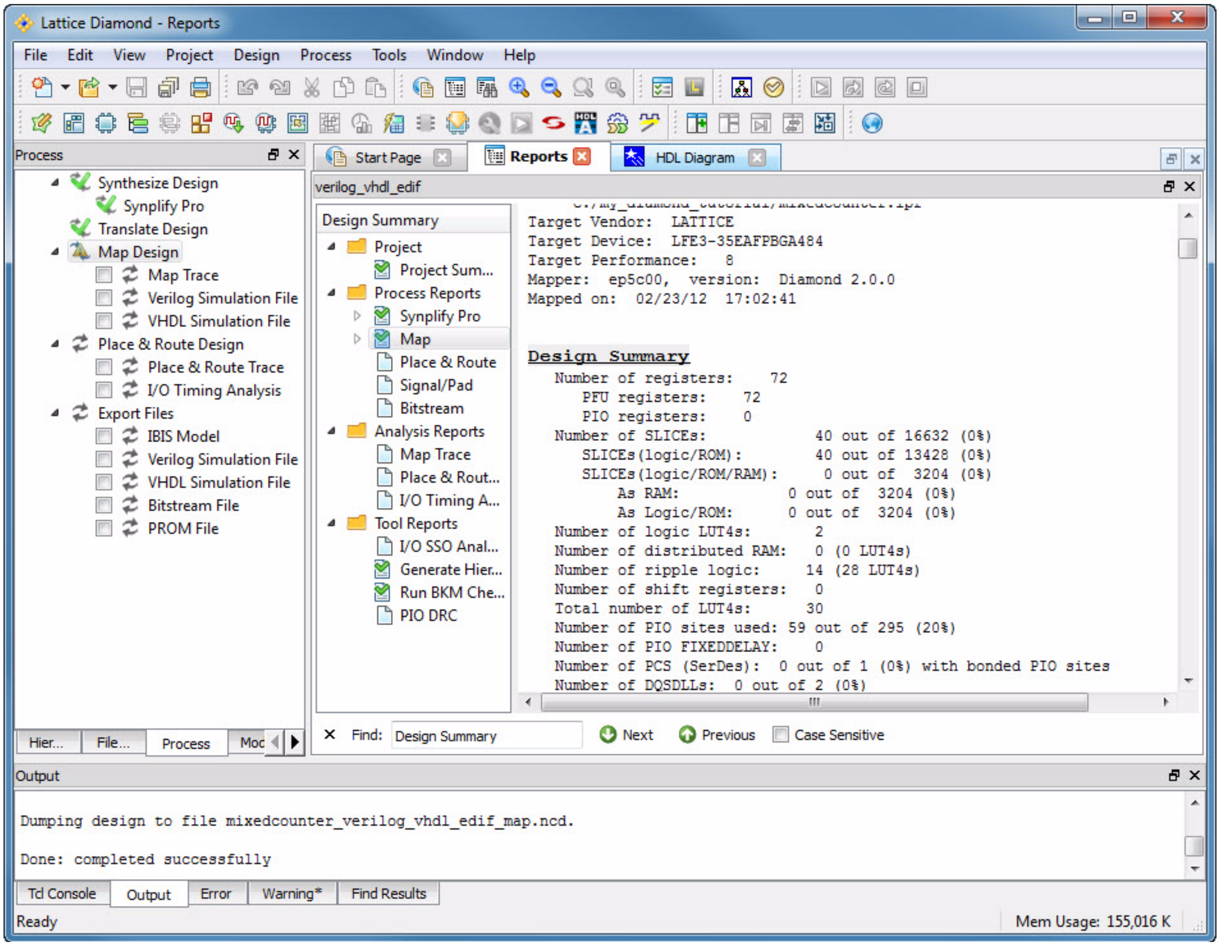
   Each major stage of an FPGA implementation is illustrated as a milestone in the Process view: Synthesize Design, Translate Design, Map Design, Place&Route Design, and Export Files. The status of any stage is represented by the following color-coded icons:

   - Completed (Green check mark) - The stage completed successfully and produced output.
   - Warning (Yellow Exclamation mark) - The stage completed with warning messages generated. You can go to the Warning panel to view the warning messages.

- Error (Red cross mark) - The stage failed. You can go to the Error panel to view the error messages.

2. From the **Design Summary** pane of the Reports view, select **Process Reports > Map**. The Map Report appears in the right panel.

3. Right click in the right pane of the Reports view, choose **Find in Text...** Type in **Design Summary**. The report highlights the Design Summary section of the report.



4. Choose **Tools > Spreadsheet View**. The Spreadsheet View appears. The Spreadsheet View is one of several preference editors available to you to define timing, I/O and floorplan constraints for the place and route tools. Preferences are organized by type into separate tabs of the Spreadsheet View.

5. Click the **Detach Tool** icon at the upper right corner of the Spreadsheet view. The Spreadsheet View is detached from the Diamond main window.

6. Click the **Period/Frequency** icon on the Spreadsheet View tool bar. The Period/Frequency Preference dialog appears.

7. Enter the following preference settings:

   Type: **FREQUENCY**

Second Type:	**Clock Net**
Available Clock Nets:	**clk_c**
Frequency:	**100MHz**

Highlight the desired clock net in the Available Clock Nets pane, and click **OK**. The Timing Preferences tab of the Spreadsheet View appears with the new **FREQUENCY** preference defined.

8. Click the **Input_setup/Clock_to_out** button on the Spreadsheet View toolbar. The INPUT_SETUP/CLOCK_TO_OUT Preference dialog appears.

9. Enter the following preference settings:

Type:	**INPUT_SETUP**
Second Type:	**All Ports**
Clock Ports/Nets:	**clk**
Time:	**10ns**

Click **OK**. The Timing Preferences tab of the Spreadsheet View appears with the new **INPUT_SETUP** preference defined. So, you can define preferences in the relevant preference dialog.

10. From the Timing Preference tab, right-click the INPUT_SETUP entry, and select **New INPUT_SETUP...**.The INPUT_SETUP/CLOCK_TO_OUTPUT Preference dialog appears.

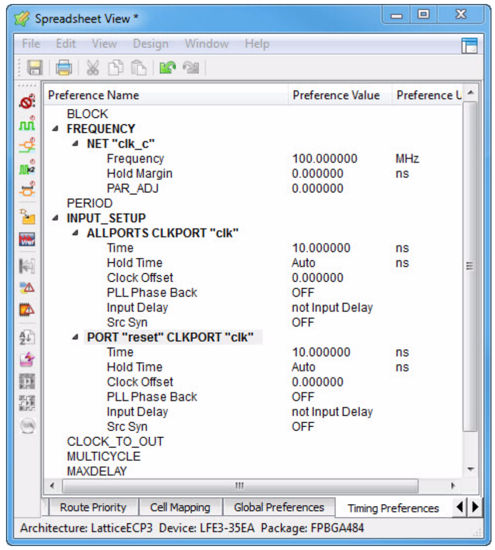
11. Enter the following settings:

Type:	**INPUT_SETUP**
Second Type:	**Individual Ports**
Available Input Ports:	**reset**
Clock Ports/Nets;	**clk**
Time:	**10ns**

Click **OK**. The Timing Preference tab of the Spreadsheet View appears with the new INPUT_SETUP preference defined.

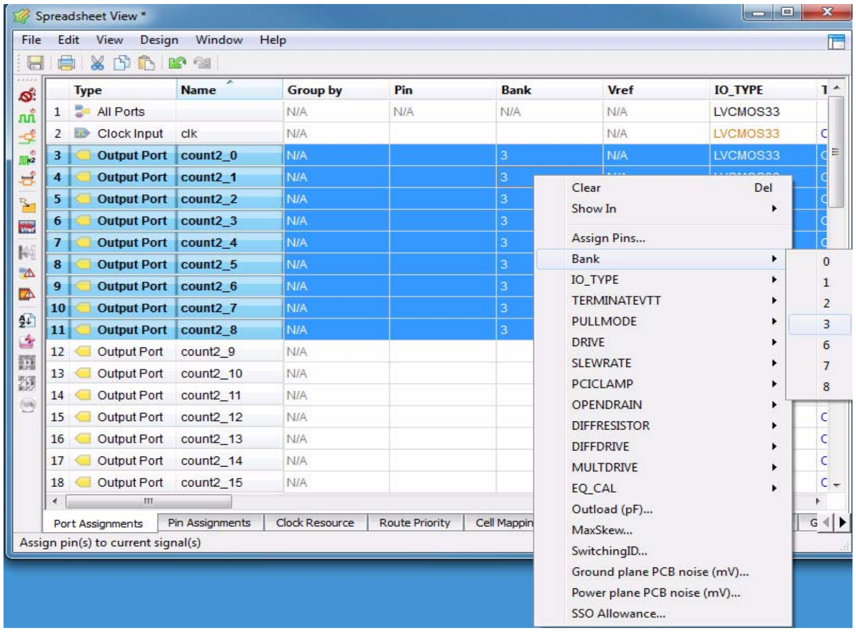
The preference dialog can be invoked from the toolbar icon, the menu item (**Edit > Preferences** from the Spreadsheet view), or from the right-click menu of the Spreadsheet view. You can also double click on a value

in Timing Preferences tab and edit the value directly.



12. Select the Port Assignments sheet from the Spreadsheet view.
13. Right click the IO_Type cell of the All Ports row. A pull-down menu of signal standards appears. Select **LVCMOS33**. The port attributes display is updated with the new IO_TYPE. Cell entries in the Spread Sheet view are color-coded to indicate the source of a preference setting:
    - Black - User-defined setting.
    - Blue - Default.
    - Orange - Implied by another user-defined setting.
14. Click the Name column to sort the port names. Select port count2_0 through count2_8. Right-click a Bank cell of the selection and choose **Bank > 3** from the pop-up menu.

The Bank cells are updated for the selected range of ports.



15. Choose **File > Save mixedcounter.lpf** from the detached Spreadsheet View. The project Logical Preference File (.lpf) is updated. Close the Spreadsheet view.

16. From the File List view of the Diamond main window, LPF Constraints Files folder, double-click the mixedcounter.lpf file. The Source Editor appears with the ASCII LPF file. Note the timing and location preferences defined so far. Close the Source Editor.

Task 6: Running Place and Route

Use the Process view to run the Translate Design, Map Design, and Place&Route Design process stages.

To run place and route:

1. From the Process List double-click **Place & Route Design**. The place and route tools are run. Intermediate results appear in the Output frame of the Diamond main window.

2. From the **Design Summary** pane of the Reports view, find the **Process Reports** section. You will find a green check mark appears before the reports generated successfully. Expand the **Process Reports** section. Select **Place & Route**. Details about Place & Route appear in the pane to the right.

3. From the Process List double-click **Place & Route Trace**. The TRACE timing analyzer is run.

4. From the **Design Summary** pane of the Reports view, expand **Analysis Reports**, and then select **Place & Route Trace** to view the report in the pane to the right.
5. From the Process List double-click **I/O Timing Analysis**. The timing analysis is run.
6. From the **Design Summary** pane of the Reports view, select the **I/O Timing Analysis** section of Analysis Reports. The I/O Timing Report appears in the right pane of the Reports view.

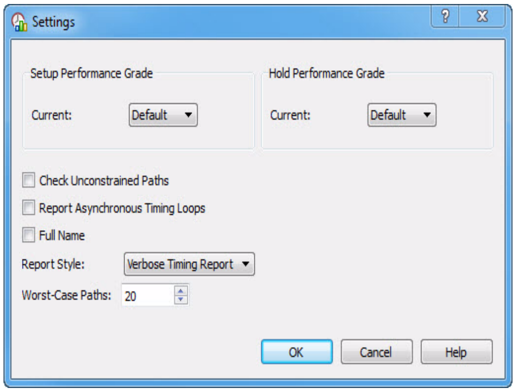
Task 7: Examine Post Place and Route Results

You can use the Timing Analysis view to examine the timing analysis results.

Examine timing analysis results:

1. Choose **Tools > Timing Analysis View**. The Timing Analysis view appears.
2. Click the **Detach Tool** icon from the right corner of the Timing Analysis view. The Timing Analysis view is detached from the Diamond main window.

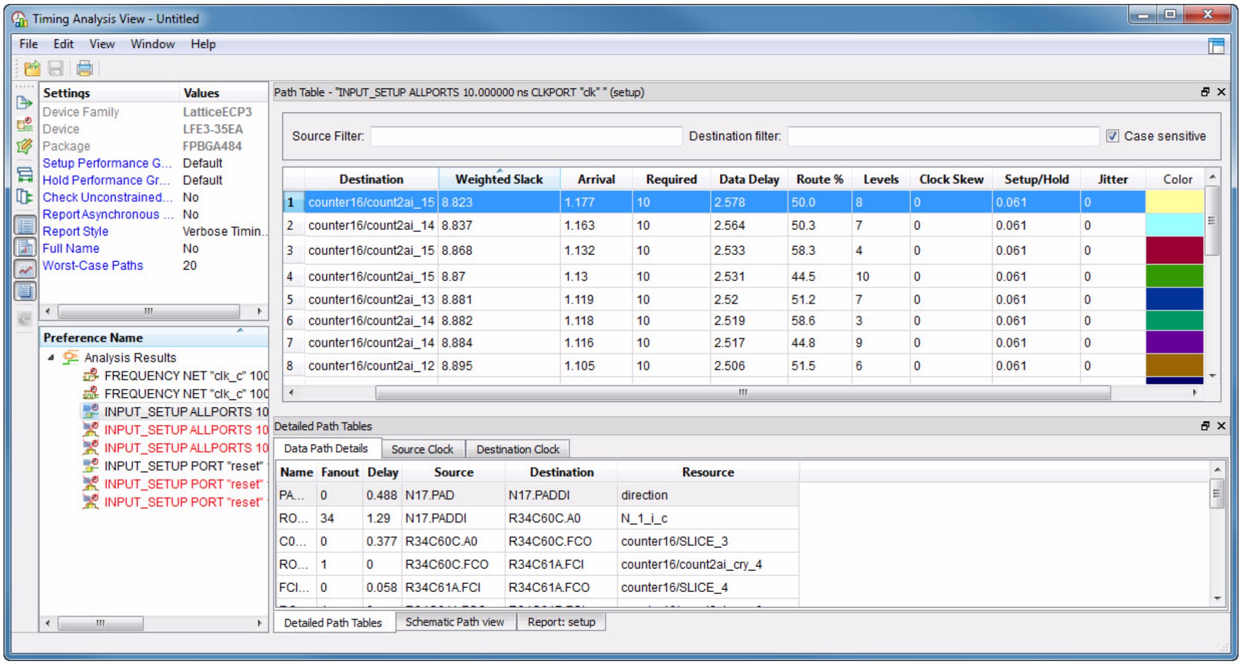
   A summary of the post-route static timing analysis settings such as target device information, preference file, performance grade, and environment conditions appear in the upper left pane. The lower left pane provides an index of the available analysis results. Related timing preferences appear in each analysis section.
3. From the Analysis pane (on the lower left of the Timing Analysis view), select **INPUT_SETUP ALLPORTS 10ns CLKPORT "clk" setup**. The Path Table is populated in the upper right of the Timing Analysis view, with the Source, Destination, Weighted Slack, Arrival, Required, Data Delay, Route %, Levels, and other details.
4. Select Row 1 of the Path Table. The Detailed Path Tables in the lower pane are populated with details.
5. Choose **Edit > Settings**, or click the Settings icon from the toolbar in the Timing Analysis view. The Settings dialog appears.
6. Enter **20** into the Worst-Case Paths field and click **OK**. The Timing Analysis view is refreshed with the additional path data.



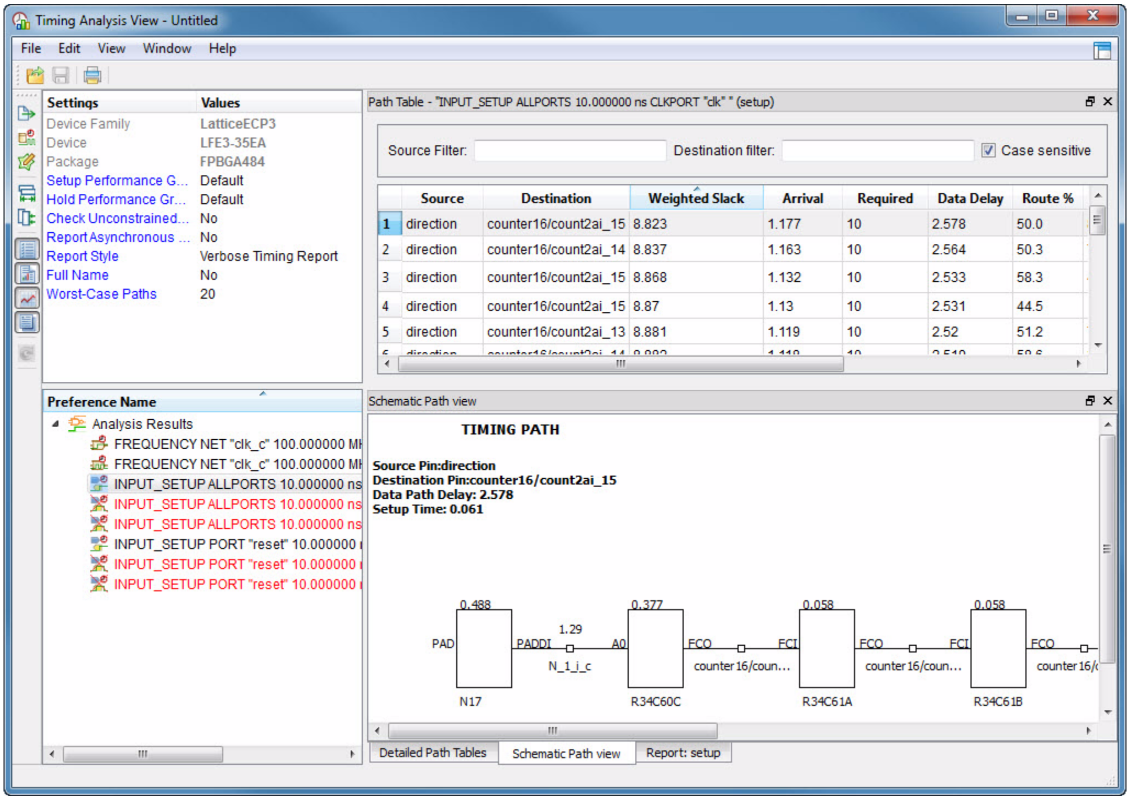
7. You can use the Source Filter field of the Path Table to filter out all the wanted paths. Delete the text from the Source Filter field, all the sources appear in the Source list again.

8. Select the first row in the **Path Table**. The Detailed Path Tables are updated.

9. Select the **Data Path Details** tab. Each component of the data path delay is identified alternating between route delays and combinatorial or clock-to-output type delays.



10. Select the **Schematic Path view.** A schematic graphic of the data path timing path appears.



Task 8: Adjust Static Timing Constraints and Review Results

In this task, you will edit timing constraints for STA (Static Timing Analysis) using the Timing Preference File (TPF) version of Spreadsheet View, and then you will use Timing Analysis view to review the results.

Timing analysis within Lattice Diamond can be performed at three points in a typical design flow: post-synthesis, post-map when the post-synthesis netlist of the design has been translated to the target device, post-placement, and post-route. Each stage provides a progressively more accurate report of delay characteristics. Timing analysis at the synthesis stage is performed by the respective synthesis tool: Synplify Pro or Precision. Additional features are provided by Diamond for post-map stages of STA.

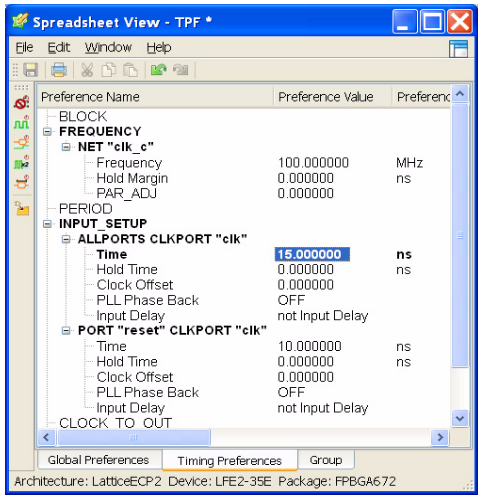
By default, the timing analysis engine, TRACE, uses those timing constraints applied by timing-driven map, place, and route. However, timing preferences can be modified, which allows you to manage the timing objectives of the implementation tools independent of static timing analysis. To accommodate an experimental static timing analysis loop, the TPF Spreadsheet View allows you to edit the timing preferences for use with the Timing Analysis view. This allows you to establish modified or additional timing preferences independent of the constraint set used for MPAR.

Tighten the timing objective of a preference and examine the results:

1. From the Preference Name list on the lower left of Timing Analysis View, select **INPUT_SETUP ALLPORTS 10ns CLKPORT “clk” Setup**, right-click and choose **TPF Preferences**.

   Spreadsheet View – TPF appears.

2. Select the Timing Preferences sheet of the TPF Spreadsheet View. Right-click 10.00000ns in the Preference Value column for the ALLPORTS CLKPORT “clk” and choose **Edit Value**. Enter **15ns** into the Preference Value field and press Enter.



3. After a few moments, return to Timing Analysis View.

   The Update button  on the toolbar is now rotating.

4. Click the **Update** button. After a short while, the indicator stops rotating and the new analysis results become available in Timing Analysis View.

   In the title bar of Timing Analysis View, “Untitled” appears with an asterisk, which indicates an in-memory change to the timing preferences. You can save the change to a Timing Preference File (.tpf) by choosing **Save > Save Untitled As** and giving it a name and location. The .tpf file will then appear in the Analysis Files folder of the File List Pane. These .tpf files enable you to experiment with different timing settings without affecting the .lpf source file. For more information, see “Using Timing Analysis View” in the “Analyzing Static Timing” section of the Diamond online Help.

5. Close Timing Analysis View and Spreadsheet View – TPF. In the Save dialog box, click **No** to discard the change.

Task 9: Comparing Multiple Place and Route Runs

Use the Run Manager to run multiple synthesis and place and route passes, compare the timing score results, and load the native circuit description (NCD) database of the best run into the workspace for further analysis.

You can create multiple strategies or implementations for the design. Then compare the runs with different implementation and strategy combination. One implementation can only be bound with one active strategy.

Now let's create a new implementation.

To create a new implementation:

1. Choose **File > New > Implementation** from the Diamond main window. Or, right click on the project name icon  from the File List view and choose **Add > New Implementation**.
2. In the New Implementation dialog box, type **verilog_vhdl** in the Name text box. By default, the directory and location will be the same name as the implementation name. You can change the directory or location to a desired one.
3. Choose **Strategy1** from the default strategy drop-down menu.
4. Click **Add Source** and choose **Browser**.
5. In the Import File dialog box, navigate to <..*diamond_install_directory**version#*\examples\mixedcounter\source>.
6. Select **topcount.v**, **typecount.vhd,** and **typepackge.vhd** and click **Open**. The selected sources are listed in the Source Files field.
7. Select the "**Copy source to implementation directory**" option and click **OK**.

   The new implementation **verilog_vhdl** is now displayed in the File List pane.

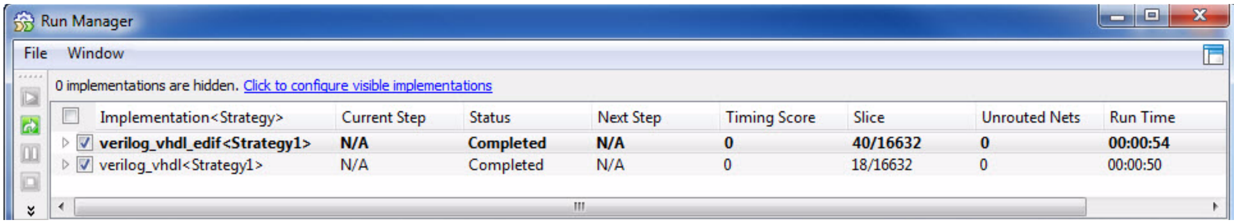
   Note

   If you want to make this new implementation active, right-click **verilog_vhdl** and choose **Set as Active Implementation**. You can have multiple implementations in your project, but you can make only one implementation active in your project at one time.

Now you will compare the run results of the **verilog_vhdl_edif** and **verilog_vhdl** implementations.

1. From the Diamond main window, choose **Tools > Run Manager**.
2. The Run Manager displays a table of implementation<strategy>: **verilog_vhdl_edif<Strategy1>** and **verilog_vhdl<Strategy1>**.
3. Enable the **verilog_vhdl_edif<Strategy1>** and **verilog_vhdl<Strategy1>** by setting the check boxes for each. You will see the verilog_vhdl_edif<Strategy1> implementation has the status of **100%** in **Progress**.

4. Click the **Rerun** button on the Run Manager toolbar. The two implementations start to run simultaneously.



In a few minutes, the results of the run appear in the table. Statistics such as Start time, Run time, Score, Unrouted, Level/Cost, and Description appear. The row in bold font indicates the active implementation that is loaded. The table provides a quick review of the quality of results produced by a particular strategy. To closely examine a particular run with analysis tools, such Timing Analysis View or Power Calculator, you can set the active strategy to be loaded.

If your system provides a multiple-core processor, you can set more implementations to be run concurrently. Go to the **Options** dialog (**Tools > Options**) of the Diamond main window, **Environment > General** tab, the **Maximum number of processes in run manager** option. Enter a number in the box in front of this option. The default value is 2. The maximum allowed value is 16.

5. Choose **View > Reports**. In Reports View, you can view results related to the run of the current active implementation. The report for **verilog_vhdl_edif** appears in Reports View.

Note

To view report of the inactive implementation, right click on the inactive implementation in the File List pane and choose **Set as Active Implementation**. Only reports of the current active implementation are displayed in the Reports view.

Task 10: Running Export Utility Programs

Use the Process view to generate files for exporting.

1. From the Process view, choose **Export Files**. A set of export files appear under the Export Files process.

2. Select the following Export Files:

   IBIS Model

   VHDL Simulation File

   Bitstream File

3. Click the **Run** button on the Diamond toolbar. Diamond generates the selected files and saves them in your project directory.

Task 11: Download a Bitstream to an FPGA

This task requires that you have a LatticeECP3 Versa Development Kit.

In the previous section, you generated export files including a Bitstream File (.bit). In this section, you will use Diamond Programmer to download a bitstream to a LatticeECP3 FPGA mounted on a LatticeECP3 Versa Development Kit board.

For the purpose of this tutorial, a .bit file that is designed specifically for the LatticeECP3 Versa Development Kit board is provided in <*install_path*>/docs/tutorial/**Programmer_tutor**.

Note

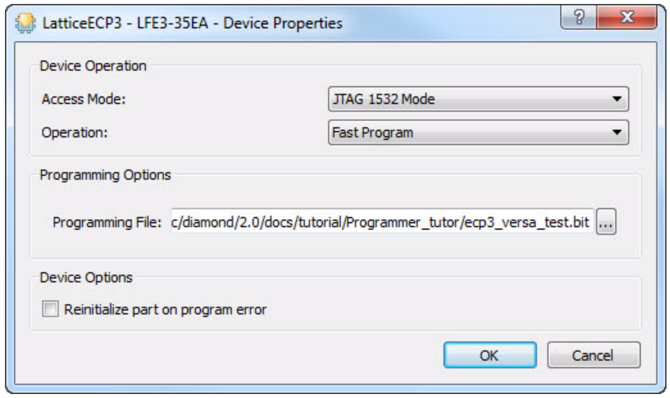
It is recommended that you do not use the bitstream created in Task 10 because this bitstream is not optimized for use with the LatticeECP3 Versa Development Kit board. For this task, use the bitstream (.bit) file provided in <*install_path*>/docs/tutorial/**Programmer_tutor**.

To download the bitstream to the FPGA on the board:

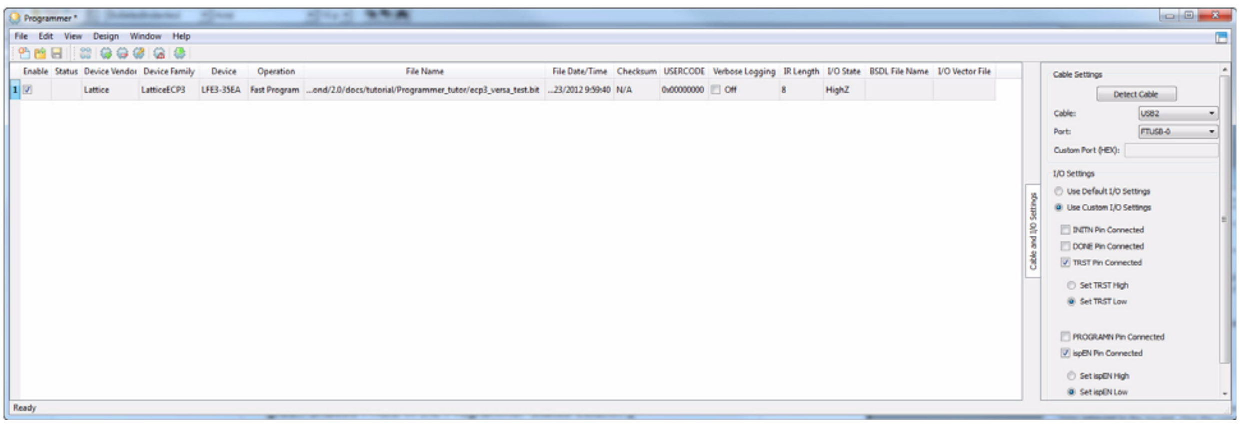
1. Remove any Lattice USB Programming cables from your system.
2. Connect the power supply to the development board.
3. Connect a USB cable from your computer to the LatticeECP3 Versa Development Kit board. Give the computer a few seconds to detect the USB device moving to step 4.
4. In Diamond, choose **Tools > Programmer**.
5. In the Getting Started dialog box, choose **Create a new Project from a Scan.**
   a. In the Cable box, select **USB2**.
   b. In the Port box, choose the only setting available in the drop-down menu, **FTUSB-0**.
   c. Leave the **Import File to Current Implementation** box unchecked.
   d. Click **OK**.

   Programmer scans the device database, and then the Programmer view displays in Diamond.
6. Click the **File Name** column, and click ... to display the Open File dialog box, and browse to the ecp3_versa_test.bit file in the following directory: <*install_path*>/lscc/diamond/2.0/docs/tutorial/**Programmer_tutor**.
7. Click **Open**.
8. Double-click the Operation column to display the Device Properties dialog box and choose the following settings:
   - For Access Mode, choose **JTAG 1532 Mode** from the pull-down menu.

- For Operation, choose **Fast Program** from the pull-down menu.

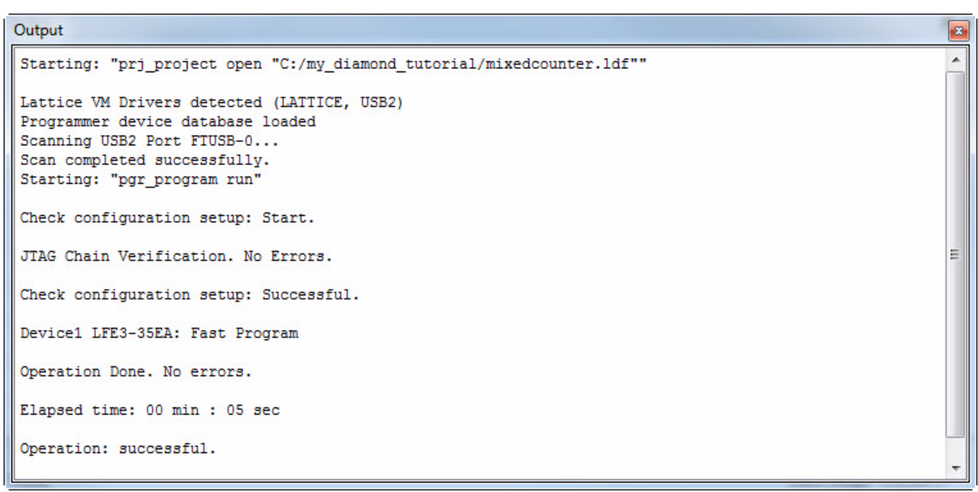


9. Click **OK**.



10. Click the Program button  on the Programmer toolbar to initiate the download.

11. If the programming process succeeded, you will see a green-shaded PASS in the Programmer Status column. Check the Programmer output console to see if the download passed.



12. At the end of this process, the FPGA is loaded with sample test bitstream. This bitstream allows you to test the functionality of the LatticeECP3 Versa Development Kit board. If the design is successfully downloaded onto the LatticeECP3 device, the period will flash in the seven segment display. To further test the design:
    a. Push the User Switch (dip switch) number 1 (closest to the edge of the board) to On to cause the Status LEDs to cycle.
    b. Push the User Switch (dip switch) number 2 (second from the edge of the board) to On to cause the LED segments to sequence. Press the top pushbutton to reset the sequence.
    c. Push the User Switch (dip switch) number 3 (third from the edge of the board) to On to cause the LED segments to spell “LATTICE*.” Press the top pushbutton to reset the sequence.
13. In Diamond, choose **File > Save**. In the Save .xcf File As dialog box, enter ecp3_versa_test.xcf in the File Name box, and click **Save.**

Task 12: Convert a File Using Deployment Tool

In the previous section, you used Diamond Programmer to download a bitstream (.bit) to a LatticeECP3 FPGA.

You will now use the Deployment Tool to convert the .bit to an industry-standard Hex file.

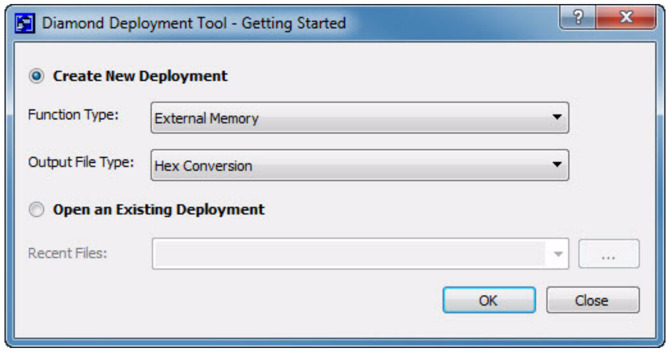
The Deployment Tool is a stand-alone tool available from the Diamond Accessories. The Deployment Tool graphical user interface is separate from the Diamond design environment. The Deployment Tool allows you to generate files for deployment for single devices and for a chain of devices. The Deployment Tool can also convert data files to other formats and use the data files it produces to generate other data file formats.

For the purpose of this tutorial, you will convert the same .bit file from Task 11 into an Intel Hex file.

To convert the .bit to an Intel Hex file using Deployment Tool:

1. Choose **Programs > Lattice Diamond Programmer 2.0 > Deployment Tool** from the Windows Start menu.
2. In the Getting Started dialog box, choose **Create New Deployment**.
    a. In the Function Type dropdown, choose **External Memory**.

b. In the Output File Type dropdown, choose **Hex Conversion**.



c. Click **OK**.

3. In the Step 1 of 4: Select Input File(s) dialog box:

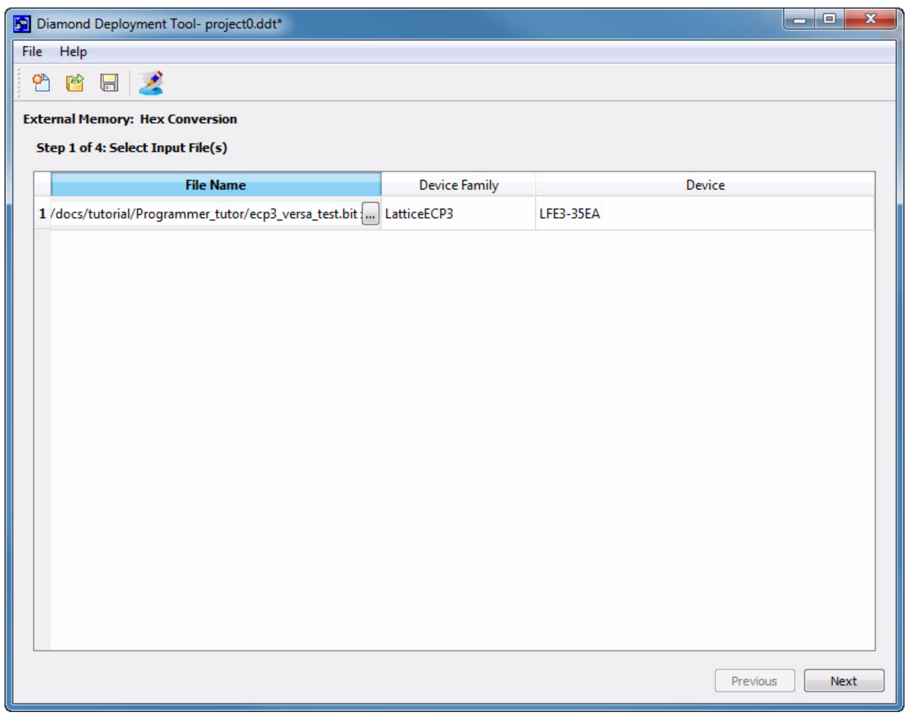
a. Click in the File Name box.

b. Click  to display the Open File dialog box.

c. Browse to the ecp3_versa_test.bit file located in <*install_path*>/docs/tutorial/Programmer_tutor.

d. Click **Open**.

e. Click **Next**.



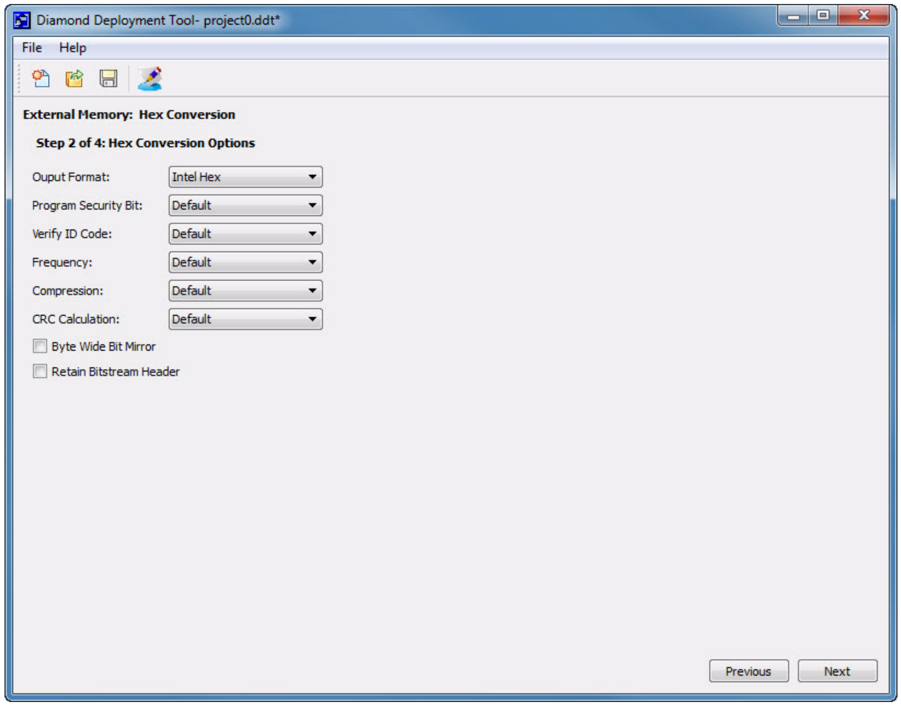
4. In the Step 2 of 4: Hex Conversion dialog box:

a. Choose Output Format as **Intel Hex**.

b. Leave all other options (Program Security Bit, Verify ID Code, Frequency, Compression, and ORC Calculation) as **Default**.

c. Leave Byte Wide Bit Mirror and Retain Bitstream Header unchecked.

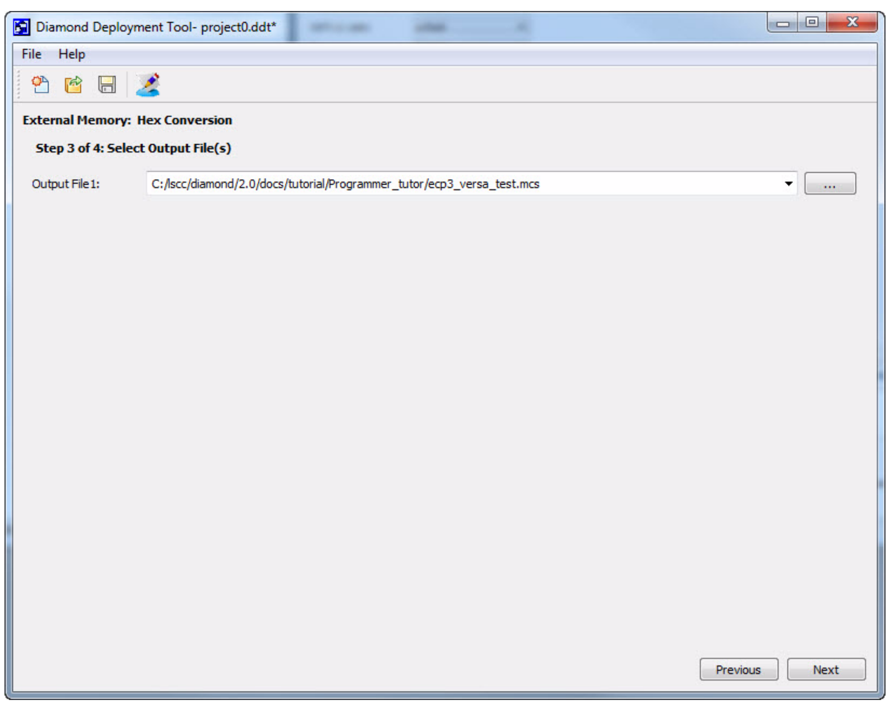
d. Click **Next.**



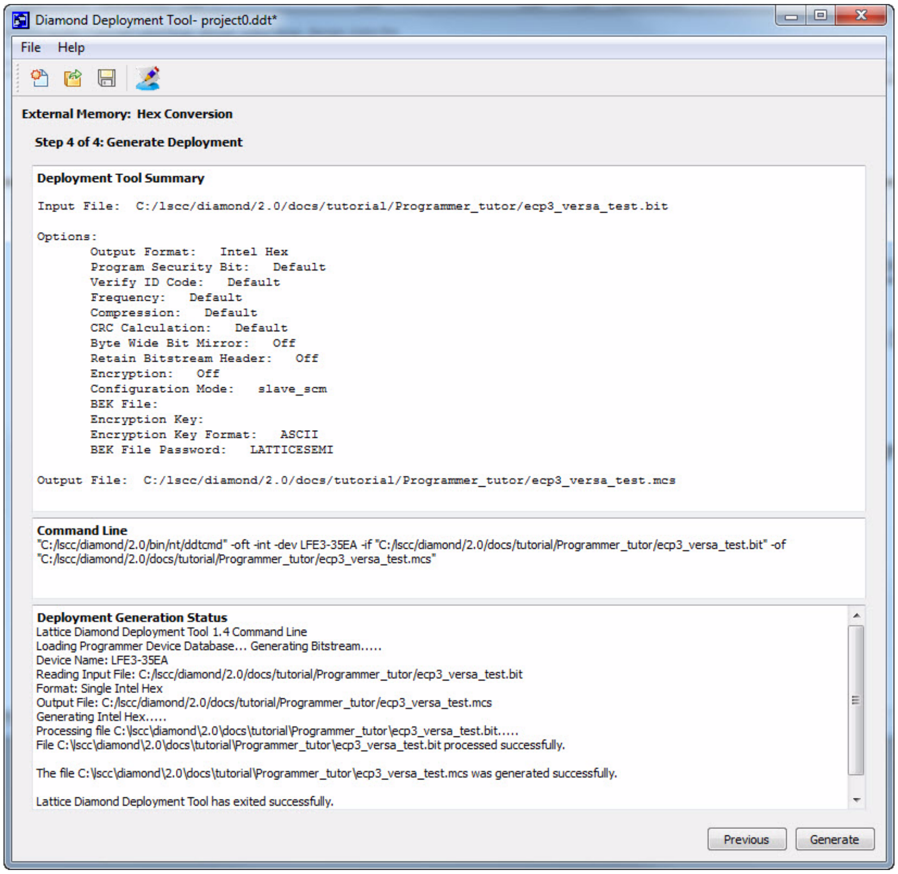
5. In the Step 3 of 4: Hex Conversion dialog box:

   a. Ensure that the Output FIle1 is ecp3_versa_test.mcs.

   b. Click **Next**.



6. In the Step 4of 4: Hex Conversion dialog box:
   a. Review the Deployment Tool Summary.
   b. Click **Generate**. The Hex file (ecp3_versa_test.mcs) is created in the Programmer_tutor directory.



The .mcs file can be used to program the SPI Flash on the LatticeECP3 Versa Development Kit board using Programmer.

To save the Deployment Tool project:

1. Choose **File > Save**,
2. In the Save As dialog box, browse to the <*install_path*>/docs/tutorial/ Programmer_tutor directory,
3. Save the Deployment Tool (.ddt) file using either the default file name or another file name.