Lattice Diamond Tutorial

For the AK-MACHX02-7000
Contents

About this document ............................................................................................................................................. 3
Revision history ....................................................................................................................................................... 3
Contact information ................................................................................................................................................. 3
Copyright information ........................................................................................................................................... 3
Lattice Diamond Tutorial ....................................................................................................................................... 4
Learning Objectives ............................................................................................................................................... 4
Time to Complete This Tutorial ............................................................................................................................ 4
System Requirements .............................................................................................................................................. 4
Accessing Online Help............................................................................................................................................. 4
About the Tutorial Design .................................................................................................................................... 4
About the Tutorial Data Flow ................................................................................................................................ 5
Task 1: Create a New Lattice Diamond Project .................................................................................................... 6
Task 2: Check Hardware Description Language .................................................................................................. 12
Task 3: Verify Functionality with Simulation .......................................................................................................... 15
Task 4: Run Synthesis Process ................................................................................................................................ 20
Task 5: Set Timing and Location Assignments ...................................................................................................... 21
Task 6: Running Place and Route .......................................................................................................................... 24
Task 7: Examine Post Place and Route Results ...................................................................................................... 24
Task 8: Run Export Utility Programs ..................................................................................................................... 27
Task 9: Download a JEDEC programming file ...................................................................................................... 27
Summary of Accomplishments ............................................................................................................................... 28
Recommended References ...................................................................................................................................... 29
About this document

Revision history

The table below displays the revision history for the chapters in this manual.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Date</th>
<th>Revision</th>
<th>Changes made</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>August 2013</td>
<td>1.0</td>
<td>First publication</td>
</tr>
</tbody>
</table>

Contact information

For the latest news, upgrades and information about Artekit products, visit the Artekit web site at http://www.artekit.eu
For technical support on this product, visit the support page at http://www.artekit.eu/contact or use the Artekit forum.
For additional information about Artekit products, consult the sources below.

<table>
<thead>
<tr>
<th>Information type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical support</td>
<td><a href="mailto:support@artekit.eu">support@artekit.eu</a></td>
</tr>
<tr>
<td>Literature</td>
<td><a href="http://www.artekit.eu">www.artekit.eu</a></td>
</tr>
<tr>
<td>Sales</td>
<td><a href="mailto:sales@artekit.eu">sales@artekit.eu</a></td>
</tr>
<tr>
<td>Products forum</td>
<td><a href="http://www.artekit.eu/forums/">http://www.artekit.eu/forums/</a></td>
</tr>
</tbody>
</table>

Copyright information

This document is copyright © 2013 Artekit Italy. All rights reserved. Any person may view, copy, print and distribute this document or any portion of this document for informational purposes only as long as the copyright notice remains included.
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 VHDL design targeted to the MachXO2 device family and AK-MACHXO2-7000 Development Kit board. 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 examining the device resources, setting timing and location assignments, and editing preferences to implement the design to the target device.

Learning Objectives

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

- Create a new Lattice Diamond project
- Check Hardware Description Language (HDL)
- Verify functionality with simulation
- Examine resources
- Run synthesis process
- Set timing and location assignments
- Run place and route
- Examine post place and route results
- Download a programming file to an FPGA

Time to Complete This Tutorial

The time to complete this tutorial is about 60 minutes.

System Requirements

The following software and hardware are required to complete the tutorial:

- Lattice Diamond software
- ARTEKIT AK-MACHXO2-7000 Development Kit
(http://www.artekit.eu/products/devboards/ak-machx02-7000/)

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 VHDL file. The design that you create is targeted to MachXO2 device families.
About the Tutorial Data Flow

Figure 1 illustrates the FPGA design data flow through the system. You may find it helpful to refer to this diagram as you move through the tutorial tasks highlighted in red.

Figure 1: Tutorial Data Flow
Task 1: Create a New Lattice Diamond 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. Do one of the following depending on your operating system:
   - From your Windows desktop, choose Start > Programs > Lattice Diamond > Lattice Diamond
   
   **Note**
   Lattice Diamond is the default Programs folder name when you install the Lattice Diamond software. Change this name accordingly if you have chosen another folder name during installation.
   
   - From your Linux platform shell window or C-shell window, execute:
     `<install_path>/bin/lin/diamond`

   The Lattice Diamond Design Environment appears, as shown in Figure 2.

   **Figure 2: Diamond Design Environment**

   ![](image)

   The initial layout provides the 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](http://www.latticesemi.com).

   For almost all questions, the place to start is Lattice Diamond’s online Help. It describes the FPGA design flow using Diamond, the libraries of logic design elements, and the details of the Diamond design tools. The Help also provides easy access to many other information sources. The Help can be accessed from Help > Lattice Diamond Help.
2. Open a new project in one of the following ways:
   - In the Start page, under Project, click New.
   - From the Diamond main window choose File > New > Project.
   - Click the down arrow in the icon from the toolbar and then choose Project.
   
   Click Next. The New Project dialog box of the Project Wizard opens.

3. Specify Project name: LEDTutorial

   **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 (_).

4. Click Browse. In the Project Location dialog box, browse to <path_LEDTutorial>/par where we are going to store the project’s files. Click Select Folder.
   By default, when you specify the project name, the implementation name is simultaneously specified the same, as shown in Figure 3. For this tutorial, leave the default implementation name as LEDTutorial. The directory to store the implementation is automatically displayed in the Location area.

   **Figure 3: New Project, Project Name Window**

5. Click Next. The Add Source dialog box appears as shown in next figure.

   **Figure 4: New Project, Add Source Window**

6. Click Add Source. The Import File dialog box appears.

7. Navigate to the folder containing the source files, which are located in the <path_LEDTutorial>/src directory (i.e. C:/LEDTutorial/src).
   Select as reported in figure 5 LEDTutorial.vhd and LEDTutorial_tb.vhd in the directory and click Open.
The Add Source step of the Wizard appears with the selected source file added.

8. Leave unchecked Copy source to implementation directory, as reported in figure 6.
9. and click Next.

Figure 6: New Project, Add Source Window

10. In Add Source window click Next. The Device Selector dialog box appears.

11. Select the following device options:
   - Family: MachXO2
   - Device: LCMXO2-7000HC
   - Performance Grade: 6
   - Package type: TQFP144
   - Operating Conditions: Commercial
   Part Names: LCMXO2-7000HC-6TG144C

The dialog box should resemble Figure 7.
12. Click **Next**.

The Select Synthesis Tool dialog box opens: choose **Synplify Pro** as in figure 8.

**Figure 8: New Project, Select Synthesis Tool Window**

13. Click **Next**.

The Project Information dialog box appears as in figure 9. The project information includes project name, location, implementation name, device, synthesis tool, and import source.

**Figure 9: New Project, Project Information Window**
14. Click Finish.

The File List and Process views are populated and the Reports view appears as reported in figure 10. The File List view displays the components of the project. The imported VHDL file appears in the Input Files folder in the File List view. The File List view organizes project files by categories: Strategies, and Implementation including Input Files, Constraint Files, Debug Files, Script Files, and Analysis Files. You may adjust file order by dragging and 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.

**Figure 10: File List and Reports view**

Note
You can also see Area, I/O Assistant, Quick, and 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 Diamond online Help.

When you create a new project in Diamond, a logical preference file (.lpf) is automatically generated and assigned the same name as the FPGA project, LEDTutorial.lpf in this case. For this tutorial a logical preference file named LEDTutorial_AK_MACHXO2_7000.lpf is provided in `<path_LEDTutorial>/lpf` directory (i.e. C:/LEDTutorial/lpf) and contains all the pin assignments needed to program this design project onto the MachXO2 FPGA. All changes that you make to logical constraints will be saved in this file until you create a new logical preference file or add another existing one.

15. In the File List, right-click on LPF Constraint file, and select Add > Existing File. Add Existing File dialog box appears.

16. Navigate to `<path_LEDTutorial>/lpf` and select the file LEDTutorial_AK_MACHXO2_7000.lpf. Choose Copy file to Implementation’s Source directory, and click Add.

17. In the File List, right-click on LEDTutorial_AK_MACHXO2_7000.lpf and choose Set as Active Preference File. See figure 11 regarding this step.
18. As reported in figure 12, in the File List, right-click on LEDTutorial_tb.vhd and choose **Include for > Simulation**.

The Process view, shown in Figure 13, lists all the processes available, such as Synthesize Design, Translate Design, Map Design, Place & Route Design, and Export Files. The Reports view provides a way to examine
and print process reports. The Reports view displays reports for the major processes. There are two panes in the Reports view. The left pane lists the reports. The right pane displays the reports. Log messages are displayed in the Output frame of the Diamond main window.

Figure 13: File View and Reports View, Project Summary

Task 2: Check Hardware Description Language

Diamond provides hardware description language (HDL) visualization and rule-checks to detect coding style violations that may lead to pre-/post-synthesis simulation mismatches. The Hierarchy view mirrors the portion of the design hierarchy that is displayed in the active graphical views.

To analyze and view the HDL design:

1. Choose Tools > HDL Diagram. The Hierarchy view appears as shown in Figure 14.
3. Choose Design > Run BKM Check. Best Known Methods (BKM) analysis is run.

The HDL Diagram with the results of the BKM analysis appears, as shown in Figure 15. Results shown in output log and HDL diagram and Hierarchy view are color coded and based on results from the BKM analysis. 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 box (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.
4. After running the BKM check, you may encounter warning and error messages.

Error and warning messages are displayed in the Output, Warning, and Error frames. If you double-click the message in the Output, Error, or Warning frames, the source indicated in the message will be opened in the associated editor. This cross-probing function can ease your check of the source file. Double-click the rtl_ledtutorial block from the HDL Diagram Design view, or right-click the rtl_ledtutorial block and choose View Connectivity. The Connectivity view, shown in Figure 16, 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 attach it back to the Diamond main window.

**Task 3: Verify Functionality with Simulation**

Diamond provides an interface to create a new simulation project file that can be imported into a standalone simulator. Diamond supports Active-HDL and ModelSim® simulation file for file exports. 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. Select **Tool > Simulation Wizard**. The Simulation Wizard dialog box appears.

2. Click **Next**.

The Simulator Project Name dialog box appears.

3. Perform the following as shown in figure 17:
   - Specify Project name: **rtl_functional**.
   - Make sure Active-HDL is selected for Simulator.
   - Click the button to browse to `<path_LEDTutorial>/sim` where to store the project’s file
Figure 17: Simulation Project Name

4. Click Next.

5. As reported in figure 18, click Yes to create a new folder

Figure 18: Simulation Wizard, New Folder

The Process Stage dialog box of figure 19 appears.


Figure 19: Simulation Wizard, Process Stage

The Add and Reorder Source dialog box appears as in figure 20.

7. Make sure all source files are present in the Source Files list.

8. Click Next.
The Parse HDL Files for Simulation dialog box of figure 21 appears.

9. Click Next.

The Summary dialog box of figure 22 appears. Make sure that Run simulator, Add top-level signals to waveform display, and Run simulation are selected.

10. Click Finish.
The Aldec Active-HDL software is launched and the simulation starts automatically as shown in figure 22.

**Figure 23: Active-HDL 9.1 Window**

To simulate the design manually:

1. In the Active-HDL Design Browser, select **testbench** from the drop down menu.

2. Select the **Structure** tab in the Design browser.

3. In the **Structure** tab, select uut: LEDTutorial(rtl_ledtutorial), right-click and then Add to Waveform. Same operation may be done by selecting all signals and drug-and-drop to waveform under Signal name column as displayed in Figure 23.
6. In Console window, write **restart** and then press Enter

7. Always in Console window, write **run 6 us** and press Enter

After completing the simulation, the waveform appears, as shown in Figure 25.

**Figure 25: Simulated Waveform**
Task 4: Run Synthesis Process

Synthesis is the process of translating a register-transfer-level design into a process-specific, gate-level netlist that is optimized for Lattice Semiconductor FPGAs. Diamond can be used with almost any synthesis tool. Diamond comes with two tools fully integrated: Synopsys Synplify Pro for Lattice and Lattice Synthesis Engine (LSE). “Fully integrated” means that you can set options and run synthesis entirely from within Diamond. As already set during Project wizard, you will be using Synopsis Synplify Pro for Lattice to synthesize your design for the MachXO2 FPGA. To change the synthesis tool, from the Diamond main window, choose Project > Synthesis Tool.

To synthesize the design and examine resource utilization:

1. From the Process View, double-click Synthesize Design.

When finished, check the icon next to Synthesize Design in the Process frame. A green check mark indicates success; a yellow triangle indicates success with warnings; a red X indicates failure.

2. When the synthesis process is complete, select View > Show Views > Post Synthesis Resources. Select the Hierarchy – Post Synthesis Resources tab, shown in Figure 26.

Figure 26: Post Synthesis Resources

The Post-Synthesis Hierarchy View displays the number of logical resources within each level of the design. In the Hierarchy table shown in Figure 20, LEDTutorial is the top module displaying the resource utilization.

- LUT4 8(8) – 8 represents the total LUT4 count utilization throughout the design and 8, round brackets, represents the LUT4 utilized only in the design module LEDTutorial.
- PFU Registers 8(8) – 8 represents the total PFU register utilization throughout the design and 8, round brackets, represents the PFU registers utilized only in the design module LEDTutorial. Similar utilization is shown for the I/O registers, carry cells and SLICEs.
Task 5: Set Timing and Location Assignments

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 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, click the sign before Map to display the sections in a list and then select Design Summary.

The report highlights the Design Summary section of the report. To view the contents of the entire report, click on the report to be viewed. The entire report is then displayed in the right pane of the Reports view. Use the scroll bar to navigate through the report. Some of the reports are divided into sections (for example, Map, Place & Route, and Signal/Pad). Click the sign before the report to display the sections in a list. Choose the desired section. The whole report will be displayed with the selected section displayed at the top of the right pane of the Reports view.
4. Choose **Tools > Spreadsheet View**.

The Spreadsheet View appears like in Figure 28. 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. Select the Port Assignments sheet from the Spreadsheet View.

6. Right click the IO_Type cell of the All Ports row. A pull-down menu of signal standards appears. Select **LVCMOS33**, if it is not already selected. 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.
7. Click the **Period/Frequency** icon on the Spreadsheet View tool bar. The Period/Frequency Preference dialog box appears like in figure 29.

8. Enter the following preference settings:
   - **Type**: FREQUENCY
   - **Second Type**: Clock Net
   - **Available Clock Nets**: clk_25mhz_c
   - **Frequency**: 25MHz

   Click **OK**.

**Figure 29: PERIOD/FREQUENCY Preference window**

The Timing Preferences tab of the Spreadsheet View appears with the new **FREQUENCY** preference defined.

9. Choose **File > Save LEDTutorial_AK_MACHXO2_7000.lpf** from the detached Spreadsheet View. The project Logical Preference File (.lpf) is updated. Close the Spreadsheet View.
10. From the File List view of the Diamond main window, LPF Constraints Files folder, double-click the LEDTutorial_AK_MACHXO2_7000.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.

**Task 7: Examine Post Place and Route Results**

Static timing analysis (STA) is a method for determining if your circuit design meets timing constraints. It is a method that does not require the use of simulation. The STA process employs conservative modeling of gate and interconnect delays that reflect different ranges of operating conditions on various dies, providing complete verification coverage. In this task, you will view the results of the Static timing analysis.

*Examine timing analysis results:*

1. Choose Tools > Timing Analysis View

The Timing Analysis view appears like in figure 30.
A summary of the post-route static timing analysis settings such as target device information, preference file, performance grade, and environment index of the available analysis results. Related timing preferences appear in each analysis section. The lower left pane provides an index of the available analysis results. Related timing preferences appear in each analysis section.

2. From the Analysis pane (on the lower left of the Timing Analysis view), select FREQUENCY NET "clk_25mhz_c" 25.000000 MHz setup. Select the first row in the Path Table. The Detailed Path Tables are updated.

3. 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, as shown in Figure 31.
4. Select the Schematic Path view. A schematic graphic of the data path timing path appears as shown in Figure 32.

Figure 32: Schematic Path View

5. Close Spreadsheet View. In the Save dialog box, click No to discard the change.
Task 8: Run Export Utility Programs

Use the Process view to generate files for exporting. One of the files exported will be a JEDEC file (.jed) which will be used to program a MachXO2 device in the next task.

1. From the Process view, choose Export Files. A set of export files appear under the Export Files process.
2. Select JEDEC File in the Export Files section
3. Click the Run button on the Diamond toolbar or double-click on JEDEC File

Diamond generates the selected file and saves them in your project directory.

Task 9: Download a JEDEC programming file

This task requires that you have an ARTEKIT AK-MACHXO2-7000 Development Kit (http://www.artekit.eu/products/devboards/ak-machx02-7000/).

In the previous section, you have generated JEDEC file LEDTutor_LEDTutor.jed. In this section, you will use Diamond Programmer tool to download a JEDEC programming file to a MachXO2 FPGA mounted on an AK-MACHXO2-7000 Development Kit board.

To download the JEDEC file to the FPGA on the board:

1. Connect a USB cable from your computer to the AK-MACHXO2-7000 Development Kit board. Give the computer a few seconds to detect the USB device moving to next step.
2. Choose Tools > Programmer
3. In the Getting Started dialog box, choose Create a new Project from a Scan, and then like figure 33:
   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. Click OK.

Figure 33: Programmer Setup windows

Programmer scans the device database, and then the Programmer view displays in Diamond.

4. Ensure that the device LCMXO2-7000HC is selected in the Device column.
5. Double-click the Operation column to display the Device Properties dialog box and verify the following settings as shown in figure 34:
   a. For Access Mode, Flash Programming Mode from the pull-down menu.
   b. For Operation, FLASH Erase, Program, Verify from the pull-down menu.
   c. Ensure that the file named LEDTutor_LEDTutor.jed is selected as the programming file.

Figure 34: Programmer settings

6. Click OK

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

8. 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 reporting message Operation: successful.

9. At the end of this process, the FPGA is loaded with the LEDTutorial design. The programming JEDEC file allows you to test the functionality of the AK-MACHXO2-7000 Development Kit board.

To further test the design:
   a) press K2 button, LD_BUTTON, to load internal register with binary value 0110. That value is also displayed on leds: D7 and D8 are switched-on, D6 and D9 are switched-off
   b) press K3 button, SR_BUTTON, to activate a right shift of internal register. “Leds simulated movement” is from left to right
   c) press K1 button, SL_BUTTON, to activate a left shift of internal register. “Leds simulated movement” is from right to left

10. In Diamond, choose File > Save. In the “Save .xcf File As” dialog box, enter LEDTutor.xcf in the File Name box, and click Save.

Summary of Accomplishments

You have completed the Lattice Diamond Tutorial. In this tutorial, you have learned how to:

- Create a new Lattice Diamond project
- Check Hardware Description Language (HDL)
- Verify functionality with simulation
- Examine resources
- Run synthesis process
- Set timing and location assignments
- Run place and route
- Examine post place and route results
- Download a programming file to an FPGA
Recommended References

You can find additional information on the subjects covered by this tutorial in the online Help.