# Generating Parameterized Modules and IP Cores

## Table of Contents

Generating Parameterized Modules and IP Cores

<table>
<thead>
<tr>
<th>Module</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verilog HDL Design with LPMs Using the Module/IP Manager</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Downloading and Evaluating ispLEVERCORE Modules</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>VHDL Design with LPMs Using the Module/IP Manager</td>
<td>58</td>
</tr>
<tr>
<td>4</td>
<td>Verilog HDL Design with LPMs Using the Lattice PMI</td>
<td>82</td>
</tr>
</tbody>
</table>

**Module 1: Verilog HDL Design with LPMs Using the Module/IP Manager**

- Task 1: Create a New Project
- Task 2: Target a Device
- Task 3: Generate an LPM Module Using the Module/IP Manager
- Task 4: Compile the Files for Functional Simulation
- Task 5: Simulate the Design and View the Waveforms
- Task 6: Instantiate the Verilog LPM Module
- Task 7: Import Source Files
- Task 8: View the Routing Results with the Floorplanner
- Task 9: Run Static Timing Analysis

**Module 2: Downloading and Evaluating ispLEVERCORE Modules**

- Task 1: Download an ispLeverCORE Evaluation Package
- Task 2: Run Functional Simulation with ModelSim
- Task 3: Run Synthesis Using Synplify
- Task 4: Create an New ispLEVER Project
- Task 5: Target the Device
- Task 6: Copy Design Files into the ispLEVER Project
- Task 7: Import Top-Level Source File in Project Navigator
- Task 8: Place and Route the Design and View the Results

**Module 3: VHDL Design with LPMs Using the Module/IP Manager**

- Task 1: Create a New Project
- Task 2: Target a Device
- Task 3: Generate an LPM Module Using the Module/IP Manager
- Task 4: Compile the Files for Functional Simulation
- Task 5: Simulate the Design and View the Waveforms
- Task 6: Instantiate the VHDL LPM Module
- Task 7: Import Source Files
- Task 8: View the Routing Results with the Floorplanner
- Task 9: Run Static Timing Analysis

**Module 4: Verilog HDL Design with LPMs Using the Lattice PMI**

- Task 1: Create a New Project
- Task 2: Target a Device
- Task 3: Import Source File
- Task 4: Instantiate the LPM Module
- Task 5: Synthesize the Design
- Task 6: Build the Database
- Task 7: Route the Design and View the Results
- Task 8: Run Static Timing Analysis
Module 5: VHDL Design with LPMs Using the Lattice PMI

Task 1: Create a New Project
Task 2: Target a Device
Task 3: Import Source Files
Task 4: Instantiate the VHDL LPM Module in Source File
Task 5: Synthesize the Design
Task 6: Build the Database
Task 7: View the Routing Results with the Floorplanner
Task 8: Run Static Timing Analysis

Congratulations
Generating Parameterized Modules and IP Cores

The ispLEVER® software offers two methods for generating parameterized modules and IP cores:

- The Module/IP Manager method, which lets you select from a list of LPMs and ispLeverCORE™ IP modules and specify the parameters from a dialog box
- The PMI (Parameterized Module Instantiation) method, which lets you embed the cores into your HDL source code using a text editor.

In both methods the software produces the instantiation template and its associated files and saves them in the project directory.

This tutorial includes 5 modules, each showing an example of one of the approaches for Verilog HDL or VHDL designs.

Module/IP Manager Method

- **Module 1** - Verilog HDL Design with LPMs Using the Module/IP Manager
- **Module 2** - Verilog HDL Design with ispLeverCORE IP Module Using the Module/IP Manager
- **Module 3** - VHDL Design with LPMs Using the Module/IP Manager

Lattice PMI Method

- **Module 4** - Verilog HDL Design with LPMs Using the Lattice PMI
- **Module 5** - VHDL Design with LPMs Using the Lattice PMI
Module 1: Verilog HDL Design with LPMs Using the Module/IP Manager

Using the Module/IP Manager, you can select an LPM from the module tree and specify parameters with the dialog box. When you click Generate, the software builds the module using your specified parameters and produces the required output files. Once you’ve declared the module and copied the boundary description to your Verilog source file, the software automatically includes it in your design.

Assumptions

You are already familiar with the Verilog HDL format hardware description language.

Learning Objectives

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

- Use ispLEVER to create a new Schematic/Verilog HDL project and target a device.
- Generate an LPM module using the Module/IP Manager.
- Compile the files for ModelSim functional simulation.
- Simulate the design and view the waveforms.
- Instantiate the Verilog LPM module.
- Import source files into the Project Navigator.
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 45 minutes
**Task 1: Create a New Project**

To begin a new project, you must create a project directory. Then you need to give the project file a name (.syn) and declare the project type (Schematic/Verilog HDL).

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

*To create a new project:*

1. Start the ispLEVER software, if it is not already running.
2. In the Project Navigator, choose **File > New Project** to open the Create New Project dialog box.
3. In the dialog box,
   - Change to the directory: `<install path>\ispcpld\examples\tutorial\tutor6\module1`.
   - In the Project name box, type `verilog_lpm.syn`.
   - In the Project type box, select **Schematic/Verilog HDL**.
   - Click **Save**.

The default project name, *Untitled*, appears in the Sources window of the Project Navigator.
4. Double-click the project title (*Untitled*) to open the Project Properties dialog box.

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

5. Type **Verilog LPM Module** as your project title and click **OK**.
Task 2: Target a Device

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

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

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

2. In the dialog box:
   - Under Family, select ispXPGA from the drop-down list.
   - Under Device, select LFX1200C.
   - Accept the remaining defaults and click OK to close the dialog box.

3. In the Confirm Change message dialog box, click Yes to continue with the operation.
4. In the next message dialog box, click **No** to retain any previous constraints.

5. Your Project Navigator should look like this:
Task 3: Generate an LPM Module Using the Module/IP Manager

Using the Module/IP Manager, you can select an LPM module from the module tree and specify parameters within the dialog box. When you click Generate, the software uses the parameters you specified and produces the required output files. Once you’ve declared the module and copied the boundary description to your HDL source file, the software automatically includes it in your design.

In this module you will create a universal shift register, which is a device that performs parallel loading of data from the input port Data to the output port Q and performs serial Left/Right shifting of data from the input port ShiftIn into LSB/MSB of the output port Q. The Module/IP Manager supports most features of LPM_SHIFTREG up to 64 bits.

To generate an shift register module using the Module/IP Manager:

1. From the Project Navigator, choose Tools > Module/IP Manager, or click the button on the Project Navigator toolbar.
2. In the left pane, expand the LPM folder and the Storage Components folder, and then select LPM_SHIFTREG.
3. In the right pane, type shift_left16 in the Module Name box and click Customize.
4. In the dialog box:
   - Select the Aclr signal name box to clear the register by setting to all "0s."
   - Under Properties, set Data Width = 16.
5. Click Generate.
A message appears in the output panel, confirming that the module has been generated successfully. The Module/IP Manager creates the following files in the project folder:

- EDIF netlist (shift_left16.edf)
- Lattice parameterized netlist file (shift_left16.ldb)
- Parameter file (shift_left16.lpc), where the file extension corresponds to the type of module generated.
- Verilog HDL instantiation template (shift_left16.v)
- VHDL instantiation template (shift_left16.vhd)
- Log file (shift_left16_lpm.log)
- Verilog HDL behavioral simulation model (shift_left16_sim.v)
- VHDL behavioral simulation model (shift_left16_sim.vhd)
- Verilog HDL testbench template (shift_left16_Tb.v)
- VHDL testbench template (shift_left16_Tb.vhd)
- Verilog HDL include command (lpm_module.include)

6. Choose **File > Exit** to exit the Module/IP Manager.
Task 4: Compile the Files for Functional Simulation

Before you can simulate a Verilog design, you must first create a library and compile the Verilog source code into that library.

Functional simulation is the process of simulating the functionality of your RTL design before synthesis, thus letting you find and correct basic design errors sooner. While functional simulation will verify your Boolean equations, it does not indicate timing problems.

The ispLEVER software supports third-party Verilog HDL simulation with ModelSim, an integrated, full function simulation environment.

To compile the design files for functional simulation:

1. In the Project Navigator, choose Tools > ModelSim Simulator, or click the button in the Project Navigator toolbar, to start ModelSim. If a welcome screen opens, click Close.

2. Make sure you are in the correct project directory. Choose File > Change Directory to open the dialog box. Check to see that the path is your project directory:

   <install_path>\ispcpld\examples\tutorial\tutorial6\module1

3. If the path is correct, click Cancel to close the dialog box. If the path is not correct, navigate to the correct path and click Open.

   Before you run the compiler, you must create a design library. In ModelSim, there are two kinds of design libraries: working libraries and resource libraries. A working library is the library into which a design unit is placed after compilation. A resource library contains design units that can be referenced within the design unit being compiled. Only one library can be the working library; in contrast, any number of libraries (including the working library itself) can be a resource library during a compilation.
The library named **work** has special attributes within ModelSim; it is predefined in the compiler and need not be declared explicitly (i.e., **library work**). It is also the library name used by the compiler as the default destination of compiled design units. In other words, the work library is the working library. In all other aspects it is the same as any other library.

4. Create a working design library into which a design unit is placed after compilation. Choose **File > New > Library** to open the dialog box.

5. Accept the defaults to create a library sub-directory named **work** (your design library) within the current working directory, initially mapped to itself. Once created, the mapped library is easily remapped to a different library.

   ![Create a New Library dialog box](image)

   **Create a New Library**
   - a new library and a logical mapping to it
   - a map to an existing library
   - Library Name: **work**
   - Library Maps to: **work**
   - OK
   - Cancel

   **Note:** Do not create a Library directory using Windows commands because the _info file will not be created.

6. Click OK. ModelSim creates the specified library directory and writes a specially formatted file named _info into that directory. The _info file must remain in the directory to distinguish it as a ModelSim library.

7. Compile the Verilog source files into the project’s work library. Choose **Compile > Compile** to open the Compile HDL Source Files dialog box.
8. The Module/IP Manager automatically created a Verilog simulation file for the shift_left16 module. Select `shift_left16_sim.v` and click Compile. The ModelSim software compiles the file and puts the output files in the Work Library tab.

\[\text{Note: The order in which you compile files is not important.}\]
Next you will compile the test fixture. The Module/IP Manager also automatically created a Verilog test fixture template. Normally, you would create a test fixture for your design by adding code to generate test stimulus for your design. For this tutorial, a completed Verilog test fixture file is in the sample files folder in your project.

9. In the Compile HDL Source Files dialog box, change Files of Type to All Files.
10. Navigate to the sample files folder, select shift16_tb.tf, and click Compile.

11. Click Done to close the Compile HDL Source Files dialog box.
Task 5: Simulate the Design and View the Waveforms

The ModelSim simulator can load and simulate Verilog designs, providing a uniform graphic interface and simulation control commands for debugging and analyzing your designs. Before you simulate the design, you have to load the test fixture file into ModelSim. After simulation, you can use the Wave window to display waveforms and Verilog nets and register variables you have selected.

To simulate the design and view the waveforms:

1. Choose Simulate > Simulate to open the dialog box.

2. On the Design tab, open your work directory, select a test fixture file (test).

3. Click Add to add the test fixture and the architecture to the timing simulation.

4. Click Load. ModelSim loads your design units and displays the results in the transcript window.

5. Choose View > Signals to open the Signals window. This window shows the Verilog nets and register variables in the region currently selected in the Structure window.
6. In the Signals window, choose **Add > Wave > Signals in Region** to open the Wave window. This window displays waveforms and Verilog nets and register variables you have selected. Current and past simulations can be compared side-by-side in one Wave window.

7. Now you are ready to run the simulation. In ModelSim, choose **Simulate > Run > Run All**.

8. ModelSim runs the simulation and adds the signals to the Wave window, although you will have to change the view to see them properly.
9. In the Wave window, choose View > Zoom > Zoom Full. You can use this window to check the logic.

10. Quit ModelSim.
Task 6: Instantiate the Verilog LPM Module

When the Module/IP Manager builds the Verilog HDL module, it produces output files for generating your design’s database and places them inside your project folder. Before you generate the database for your Verilog HDL design, you must port map the module signals to your top-level design signals.

**IMPORTANT:** This procedure is for reference. We have provided a completed top-level design file in the project sub-directory (\sample files\toplevel.v) that includes proper port mapping. You can copy this file into your project directory, or you can view this file for reference while you complete this task.

1. In the Project Navigator, choose **Window > Text Editor**.
2. In the Text Editor, choose **File > Open** to open the dialog.
3. In the dialog box:
   - In the Files of Type dropdown list, select **Verilog Files (*.v)**.
   - Select **toplevel.v**
   - Click **Open**.

4. Using the Text Editor, write the component declaration for the module in your top-level design file.

   **Tip:** You might want to also open the **shift_left16.v** file for reference and toggle between the two files by pressing Ctrl+F6.
5. Choose File > Save to save the toplevel.v design file.

6. Exit the Text Editor.
Task 7: Import Source Files

You "describe" a project by specifying the project files that will represent the design. You can either import an existing source or create a new one. The added source appears in hierarchical and alphabetical order in the Sources window.

1. In the Project Manager, choose Source > Import to open the Import File dialog box.

   Note: Notice the title of the dialog box identifies the project type as (Schematic/Verilog HDL). Therefore, even though the Module/IP Manager generated both Verilog HDL and VHDL files, you will see only those files that pertain to the project type.

2. Select these files below, and then click Open. The ispLEVER software imports the selected sources into the project and displays them in the Sources window.
   - shift_left16.v
   - toplevel.v

Note: An alternative to importing HDL source files generated by the Module/IP Manager into the Project is to import the module's LPC file. After the file is imported, you can double-click the LPC file to cause the Module/IP Manager to automatically open and load the module's configuration.
Task 8: View the Routing Results with the Floorplanner

After an initial Lattice internal database is built, the Pack & Place Design process packs the design instances or cells into Generic Logic Blocks (GLBs) and then places them on the FPGA device. The output of this process is the post-place design database and a Pack/Place Report.

The ispLEVER software routes the design in the FPGA after placement. The routing algorithm takes full advantage of the Lattice FPGA architecture to achieve maximum performance. It uses congestion-driven routing to achieve a fit with minimal congestion, and it uses timing-driven routing to achieve maximum performance. The output of this process is the post-route design database and a Route Report.

The Floorplanner allows you to view the results of Place/Pack Report and Route Report in a graphical form. The Floorplanner graphical display of device architecture, and design pack/placement, and design routing, includes:

- Device pin layout and associated design signals
- Customizable color-coded device utilization at the GLB level
- Customizable color-coded routing congestion at the column and row channel level
- GLB configuration in logic-equivalent function
- Design frequency (Fmax), longest arrival time (Tpd), register timing data (Tsu & Tco) and corresponding timing path at the GLB level
- Routed and un-routed signals (nets or instance terminals)

To route the design and view the results with the Floorplanner:

1. In the Project Navigator, double-click the Post-Route Design Floorplan process to pack, place, and route the design. When the processes are through running, the Floorplanner opens.

   The Floorplan View is one of the two windows available in the Main Window. The Floorplan View displays the physical placement of your logic. In this tutorial design example, only the shift_left16 module is in the design.
2. Select the **Floorplan View window** to make it active.

3. Choose **Edit > Select Used** to select the used (placed) cells.

4. On the toolbar, click the Show Fanin and Fanout icons.

5. This view in the Floorplan View window shows the placed logic, as well as input, output, and hardwire signals.
6. Exit the Floorplanner.

**See Also**

To learn more about other Floorplanner features, take the "Debugging ispXPGA Designs with the Floorplanner" tutorial.
Task 9: Run Static Timing Analysis

Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device’s timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To run timing analysis and view the results:

1. In the Project Navigator Sources window, select the target device.
2. In the Processes window, double-click the Performance Analyst process to pack, place, and route the design; run timing analysis; and open the Performance Analyst.
The Performance Analyst performs eight distinct analysis types. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining seven types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

3. Under Analysis, select fMAX, and then click Run. In this example, the maximum frequency is **471.7 MHz** and the longest path delay is **2.12 ns**.

*Note: Your numbers may vary slightly.*
4. Double-click the highlighted cell (2.12) in the spreadsheet window to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown are the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.
5. Exit the Performance Analyst without saving.

Congratulations

You have completed the Verilog HDL Design with LPMs Using the Module/IP Manager tutorial. In this tutorial you have learned how to:

• Use ispLEVER to create a new Schematic/Verilog HDL project and target a device.
• Generate an LPM module using the Module/IP Manager.
• Compile the files for ModelSim functional simulation.
• Simulate the design and view the waveforms.
• Instantiate the Verilog LPM module.
• Import source files into the Project Navigator.
• Pack, place, and route the design and use the Floorplanner to view the results
• Run static timing analysis using the Performance Analyst and view the results.
Module 2: Downloading and Evaluating ispLeverCORE Modules

Most ispLeverCORE Modules have evaluation packages that you can download from the Lattice website. These evaluation packages contain one or more configurations that you can test within the ispLEVER software tool suite. This tutorial module explains how to download an ispLeverCORE evaluation package, do a functional simulation of the ispLeverCORE module with ModelSim SE, and test the ispLeverCORE module through static timing analysis using the ispLEVER software.

Assumptions

You have a licensed version of ModelSim SE. (The Lattice OEM version of ModelSim cannot be used to perform functional analysis on the ispLeverCORE module.)

Learning Objectives

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

- Download an ispLeverCORE Evaluation package from the Lattice website and explore the contents.
- Perform a functional simulation of the ispLeverCORE module using the ModelSim SE software.
- Synthesize a top-level design with the ispLeverCORE as a “black box” using Synplicity.
- Use ispLEVER to create a new EDIF project and target a device.
- Import the top-level EDIF source file into the Project Navigator.
- Place and route the design, and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 45 minutes.
Task 1: Download an ispLeverCORE Evaluation Package

This task shows you how to download the ispLeverCORE evaluation package for the Reed-Solomon Encoder core for ORCA4.

To download the Reed-Solomon Encoder evaluation package:

1. Using your browser, go to the IP page on the Lattice Semiconductor website (www.latticesemi.com). Navigate to the Intellectual Property page by clicking Products > Intellectual Property > Overview. If you can’t access the site through this link, try:

2. Scroll down to Reed-Solomon Encoder and click Try.

   **Note:** to view an overview of Reed-Solomon Encoder before downloading the packet, click the Reed-Solomon Encoder link.

<table>
<thead>
<tr>
<th>Memory Control Cores</th>
<th>Try Now</th>
<th>Buy Now</th>
<th>ORCA4</th>
<th>FPGAs</th>
<th>FPCS</th>
<th>XPGA</th>
<th>XPLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR SDRAM Controller</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-Channel DMA Controller</td>
<td>TRY</td>
<td>BUY</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Digital Signal Processing</th>
<th>Try Now</th>
<th>Buy Now</th>
<th>ORCA4</th>
<th>FPGAs</th>
<th>FPCS</th>
<th>XPGA</th>
<th>XPLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reed-Solomon Encoder</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reed-Solomon Decoder</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convolutional Encoder</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turbo Encoder</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turbo Decoder</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel FIR Filter</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serial FIR Filter</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yiterbi Decoder</td>
<td>TRY</td>
<td>BUY</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. On the Reed-Solomon Encoder Downloadable Software page, click the fourth link from the top, Evaluation Package for Reed-Solomon Encoder for ORCA 4 - Configuration 4.

4. Review the trial license agreement, and then click I Agree to accept the conditions and open the File Download page.

5. In the dialog box, click Save.

6. Navigate to: <install_path>\ispTOOLS\ispcpld\examples\tutorial\tutorial6\module2 and click Save.
7. After the download process has completed, click **Close**.

8. **Go to the** `tutor6\module2` **folder and open the zipped**  
   `reeds_enco_o4_1_004_1.zip` **file. Extract its contents to the same**  
   `module2` **folder.**  
   
   The extraction produces the folder `reeds_enco_o4_1_004` **in the** `module2` **folder.**

Task 2: Run Functional Simulation with ModelSim

Functional simulation is the process of simulating the functionality of your ispLeverCORE module. Functional simulation does not test the timing performance of the core.

In order to perform this tutorial, you will need to use the ModelSim SE software. The Lattice Edition of ModelSim that comes with the ispLEVER software does not support the pre-compiled models necessary for functional simulation.

To perform functional simulation with ModelSim SE:

1. Start the ModelSim software. From your Windows desktop, choose Start > Programs > ModelSimSE > ModelSim.

2. Make sure you are in the correct project directory. Choose File > Change Directory. In the dialog box, navigate to the directory:

```
<install_path>\isptools\ispcpld\examples\tutorial\tutor6 \module2\reeds_enco_o4_1_004\orca4\ver1.0\eval\ simulation.
```

3. Click Open.
4. Run the ModelSim macro file (eval_sim_rsenc.do).
   - If you are running version 5.5e or earlier, choose Macro > Execute Macro and select the file: scripts\eval_sim_rsenc.do.
   - If you are running version 5.6a or later, choose Tools > Execute Macro and select the file: scripts\eval_sim_rsenc.do.

5. Click Open. The ModelSim macro executes an evaluation testbench designed to show some example transactions or functions associated with the core. Using the precompiled libraries, you can build your own testbenches.
6. Choose View > Wave to view the Wave Window. You will have to change the view to see the waveforms properly.

7. Choose View > Zoom > Zoom Full to enlarge the view and check the logic. Here you will see the state of each signal, and the interrelationships between the signals. This will help you gain an understanding of how the core operates.

8. Quit ModelSim in the main ModelSim Window by choosing File > Quit.
Task 3: Run Synthesis Using Synplify

This procedure shows you how to use Synplicity's Synplify outside the ispLEVER Project Navigator to synthesize a top-level design with the ispLeverCORE as a “black box” and create a top-level EDIF file.

To create an EDIF netlist of the Reed-Solomon Encoder for ORCA 4 ispLeverCORE design with Synplify:

1. In the module 2 directory, create a new directory called synthesis.
2. Start the Synplify synthesis tool. From your Windows desktop, choose Start > Programs > Lattice Semiconductor > Synplify.
3. Create a new project by choosing File > New.
4. In the New dialog box, choose Project File in the File Type box.
5. In the File Name box, type Reed_Solomon.
6. In File Location, navigate to the directory
   `<install_path>`\isp CPLD\examples\tutorial\tutor6\module2\synthesis.`
7. Click OK.
8. Click the **Add** button, and in the **Select Files to Add to Project** dialog box, navigate to the directory:

\<install_path>\ispcpld\examples\tutorial\tutor6\module2\reedsenco_o4_1_004\orca4\ver1.0\source.
9. In the order shown, select the design file, and click Add. Be sure to add the files in the following order:
   
   reeds_enco_o4_1_004_params.v  
   orca4_synplify.v  
   pll_orca.v  
   reeds_enco_o4_1_004.v  
   top_rsenc_pll.v  

![Select Files to Add to Project dialog](image)

10. Click OK.
11. Click the **Change Target** button to open the Options for Implementation dialog box.
12. In the **Device** tab,
   - Select Lattice ORCA Series 4 in the **Technology** list
   - Select 4E02 in the **Part** list, –2 in the **Speed** list
   - Select BA352 in the **Package** list
   - Enter 500 in the **Value** column of **Fanout Guide**.
   - Leave all other options as default.

13. In the **Options** tab, enable **FSM Compiler** and **Resource Sharing**.

15. In the Implementation Results tab, ensure that
   - The Implementation Name is rev_1
• **Results Directory** is set to:
  `<install_path>\ispcpld\examples\tutorial\tutor6\module2\synthesis\rev_1`

16. Change **Result File Name** to `Reed_Solomon.edn`. Leave other options as default.

![Options for implementation: Reed_Solomon : rev_1](image)

17. Click **OK**.

18. Click **Run**. The EDIF file named `reed_solomon.edn` will be output to your `synthesis/rev_1` directory.
19. Exit Synplicity by choosing **File > Exit**. It is not necessary to save the Synplicity project for this exercise, but if you want to save the project, click **Yes** in the dialog box, choose a project name, and save the project. Otherwise, click **No** in the dialog box.
Task 4: Create an New ispLEVER Project

In this task, you will create an ispLEVER project. To begin a new project, you must create a project directory. Then you need to give the project file a name (.syn) and declare the project type (EDIF).

The ispLEVER software saves an initial design file with the .syn file extension in the directory you specify. All project files are copied to or created in this directory.

To create a new ispLEVER project:

1. In the module 2 folder, create a new folder named levercore. This will be your ispLEVER project directory.

2. Start the ispLEVER system. From your Windows desktop, choose Start > Programs > Lattice Semiconductor > ispLEVER.

3. In the Project Navigator, choose File > New Project to open the Create New Project dialog box.

4. In the dialog box,
   - Change to the directory:
     \install_path\ispcpld\examples\tutorial\tutor6\module2\levercore.
   - In the Project name box, type reed_solomon.syn.
   - In the Project type box, select EDIF.
   - Click Save.

The default project property name, Untitled, appears in the Sources window of the Project Navigator.
5. Double-click the project title (Untitled) to open the Project Properties dialog box.

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

6. Type reed_solomon as your project property, and click OK.
Task 5: Target the Device

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

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

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

2. In the dialog box:
   - Under Family, select or4e00.
   - Under Speed Grade, select –2.
   - Under Operating Conditions, select Commercial.
   - Under Device, select or4e02.
   - Under Package Type, select BA352.
   - Under Part Name, select or4e02-2BA352C.
   - Click OK, to close the dialog box.
3. In the Confirm Change message dialog box, click **Yes** to continue with the operation.

4. Your Project Navigator should look like this:
Task 6: Copy Design Files into the ispLEVER Project

In this task you will import files into your ispLEVER project from the evaluation package and your synthesis/rev1 folder and place them into the project directory named levercore that you created in Task 4.

To copy the designs from the evaluation package:

1. Open the directory
   `<install_path>`\ispCpld\examples\tutorial\tutor6\module2\reeds_EncO_o4_1_004\orca4\ver1.0\par.

2. Copy the files `reeds_enco_o4_1_004.ngo` and `reeds_enco_o4_1_004.prf`.

3. Paste the files into the `levercore` folder:
   `<install_path>`\ispCpld\examples\tutorial\tutor6\module2\levercore.

4. Navigate back to the synthesis\rev1 folder and copy the following EDIF file and paste it in your project folder `tutor6\module2\levercore`:
   - `reed_solomon.edn`

5. Delete the existing `reed_solomon.prf` file.

6. Change the name of the `reeds_enco_o4_1_004.prf` file to `reed_solomon.prf`.

Your project folder should contain the following files:
Task 7: Import Top-Level Source File in Project Navigator

You "describe" a project by specifying the project files that will represent the design. You can either import an existing source or create a new one. The sources appear in hierarchical and alphabetical order in the Sources window.

For this task, you will import the `reed_solomon.edn` file that you created earlier in this tutorial.

To import the source files:

1. In the Project Navigator, choose **Source > Import** to open the Import File dialog box.
2. Select the following file from `tutor6\module2\levercore` directory:
   - `reed_solomon.edn`
3. Click **Open**.
4. In the Import EDIF dialog box, choose Synplicity as the CAE Vendor.
5. The ispLEVER software imports the selected source into the project and displays it in the Sources window.
Task 8: Place and Route the Design and View the Results

1. In the Project Navigator, choose **Tools > Timing Checkpoint Options**.

2. In the Timing Checkpoint Options dialog box, in the Checkpoint Options box, click **If Checkpoint Fails: Continue** in both places that it appears in the dialog box.

3. Click **OK**.
4. In the Project Navigator Processes for Current Source pane, right-click Place & Route Design. In the Properties dialog box, set the following property:

- Routing Passes: 15

In the Advanced Options portion of the Properties dialog box, set the following properties:

- Placement Iterations: 1
- Placement Iteration Start Pt: 4
- Placement Save Best Run: 1
- Routing Resources Optimization: 0
- Routing Delay Reduction Passes: 0

Leave other options as default.

**Note:** The properties listed above have been tested to achieve maximum results for this tutorial. The readme.html file included in the top-level ispLeverCORE directory gives suggested properties that you can try to maximize the results of your place and route.

5. Click Close.
6. In order to view the timing results in the Report Viewer, choose **Options > Environment**, and in the **Log** tab, choose **Using Report Viewer**, and then click **OK**.

![Environment Options Dialog](image)
7. Select the **Map Report** process in the Processes for Current Source pane, and then choose **Process > Start**. The ispLEVER software automatically builds a database maps the design, and generates the Map Report.

*Note: This process may take a minute. If you receive a message that warnings were generated, just click OK.*
8. View the Map Report in the Report Viewer. Here you can check the size and resource utilization of your design.
9. Select the **Place & Route Trace Report** process in the Processes for Current Source pane, and then choose **Process > Start**. The ispLEVER software places and routes the design.

**Note:** This process can take several minutes. If you receive a message that warnings were generated, just click OK.

Here you can check the speed of the Reed-Solomon ispLeverCORE module. If the size and speed are acceptable, you can proceed to purchase the core, or evaluate it further by instantiating it into your top-level project.
Congratulations
You have completed the Downloading and Evaluating ispLeverCORE Modules tutorial. In this tutorial you have learned how to:

- Download an ispLeverCORE evaluation package from the Lattice website.
- Run functional simulation of the core using ModelSim SE software.
- Use Synplicity to synthesize a top-level design with the ispLeverCORE as a “black box” using Synplicity.
- Use the ispLEVER software to create a new EDIF project and target a device.
- Import the top-level EDIF source file into the Project Navigator.
- Place and route the design, and view the results in the Report Viewer.
Module 3: VHDL Design with LPMs Using the Module/IP Manager

Using the Module/IP Manager, you can select a module or IP core from the module tree and specify parameters with the dialog box. When you click Generate, the software builds the module or IP core using your specified parameters and produces the required output files. Once you’ve declared the module or IP core and copied the boundary description to your VHDL source file, the software automatically includes it in your design.

The advantage of using the Module/IP Manager is its convenience: all available modules and IP cores are listed by type, and supported parameters are clearly indicated. The disadvantage is the additional time required for editing a design that contains more than one module or IP core.

Assumptions

You are already familiar with the VHDL format hardware description language.

Learning Objectives

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

- Use ispLEVER to create a new Schematic/VHDL project and target a device.
- Generate an LPM module using the Module/IP Manager.
- Compile the files for ModelSim functional simulation.
- Simulate the design and view the waveforms.
- Instantiate the VHDL LPM module.
- Import source files into the Project Navigator.
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst and view the results.

Time to Complete This Tutorial

The time to complete this tutorial is about 45 minutes.
Task 1: Create a New Project

To begin a new project, you must create a project folder. Then you need to give the project file a name (.syn) and declare the project type (Schematic/VHDL).

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

To create a new project:

1. Start the ispLEVER software, if it is not already running.
2. In the Project Navigator, choose File > New Project to open the Create New Project dialog box.
3. In the dialog box,
   - Change to the folder: `<install_path>\ispcpld\examples\fpga\tutorial\tutorial6\module3`.
   - In the Project name box, type `vhdl_lpm.syn`.
   - In the Project type box, select Schematic/VHDL.
   - Click Save.
The default project title, Untitled, appears in the Sources window of the Project Navigator.

4. Double-click the project title (Untitled) to open the Project Properties dialog box.

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

5. Type VHDL LPM Module as your project title and click OK.
Task 2: Target a Device

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

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

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

2. In the dialog box:
   - Under Select Device, select ispXPGA from the Family box.
   - Under Select Device, select LFX1200C from the Device box
   - Accept the default settings and click OK.

3. In the Confirm Change message dialog box, click Yes to continue with the operation.
4. In the next message box, click **No** to retain any previous constraints.

5. Your Project Navigator should look like this after you resize the windows:
Task 3: Generate an LPM Module Using the Module/IP Manager

Using the Module/IP Manager, you can select an LPM module from the module tree and specify parameters within the dialog box. When you click Generate, the software uses the parameters you specified and produces the required output files. Once you've declared the module and copied the boundary description to your HDL source file, the software automatically includes it in your design.

In this module you will create a universal shift register, which is a device that performs parallel loading of data from the input port Data to the output port Q and performs serial Left/Right shifting of data from the input port ShiftIn into LSB/MSB of the output port Q. The Module/IP Manager supports most features of LPM_SHIFTREG up to 64 bits.

To generate a shift register module using the Module/IP Manager:

1. From the Project Navigator, choose Tools > Module/IP Manager, or click the button on the Project Navigator toolbar.
2. In the left pane, expand the LPM folder and the Storage Components folder, and then select LPM_SHIFTREG.
3. In the right pane, type shift_left16 in the Module Name box and click Customize.
4. In the dialog box:
   - Select the Aclr signal name box to clear the register by setting to all "0s"
   - Under Properties, set the Data Width = 16.
5. Click Generate.
6. The Module/IP generate displays a message in the output panel that the module has been generated successfully. The Module/IP Manager creates the following files in the project folder:
   - VHDL instantiation template (shift_left16.v)
   - VHDL behavioral simulation model (shift_left16_sim.v)
   - VHDL testbench template (shift_left16_Tb.v)
   - VHDL Include command (lpm_module.include)
   - VHDL template (shift_left16.vhd)
   - VHDL behavioral simulation model (shift_left16_sim.vhd)
   - VHDL testbench template (shift_left16_Tb.vhd)
   - Lattice parameterized netlist file (shift_left16.ldb)
   - Parameter file (shift_left16.lpc), where the file extension corresponds to the type of module generated.

7. Choose **File > Exit** to exit the Module/IP Manager.
Task 4: Compile the Files for Functional Simulation

Before you can simulate a VHDL design, you must first create a library and compile the VHDL source code into that library.

Functional simulation is the process of simulating the functionality of your RTL design before synthesis, thus letting you find and correct basic design errors sooner. While functional simulation will verify your Boolean equations, it does not indicate timing problems.

The ispLEVER software supports third-party VHDL simulation with ModelSim, an integrated, full function simulation environment.

To compile the design files for functional simulation:

1. In the Project Navigator, choose Tools > ModelSim Simulator, or click the button in the Project Navigator toolbar, to start ModelSim. If a welcome screen opens, click Close.

2. Make sure you are in the correct project directory. Choose File > Change Directory to open the dialog box. Check to see that the path is your project directory:

   `<install_path>\ispcpld\examples\tutorial\tutor6\module3`

3. If the path is correct, click Cancel to close the dialog box. If the path is not correct, navigate to the correct path and click Open.

   Before you run the compiler, you must create a design library. In ModelSim, there are two kinds of design libraries: working libraries and resource libraries. A working library is the library into which a design unit is placed after compilation. A resource library contains design units that can be referenced within the design unit being compiled. Only one library can be the working library; in contrast, any number of libraries (including the working library itself) can be a resource library during a compilation.
The library named work has special attributes within ModelSim; it is predefined in the compiler and need not be declared explicitly (i.e., library work). It is also the library name used by the compiler as the default destination of compiled design units. In other words, the work library is the working library. In all other aspects it is the same as any other library.

4. Create a working design library into which a design unit is placed after compilation. Choose File > New > Library to open the dialog box.

5. Accept the defaults to create a library sub-directory named work (your design library) within the current working directory, initially mapped to itself. Once created, the mapped library is easily remapped to a different library.

6. Click OK. ModelSim creates the specified library directory and writes a specially formatted file named _info into that directory. The _info file must remain in the directory to distinguish it as a ModelSim library.

7. Compile the VHDL source files into the project’s work library. Choose Compile > Compile to open the Compile HDL Source Files dialog box.

8. The Module/IP Manager automatically created a VHDL simulation file for the shift_left16 module. Select shift_left16_sim.vhd.

Note: Do not create a Library directory using Windows commands because the _info file will not be created.
9. Click **Compile**. The ModelSim software compiles the file and puts the output files in the Work Library tab.

Next you will compile the test bench. The Module/IP Manager also automatically created a VHDL testbench template. Normally, you would create a testbench for your design by adding code to generate test stimulus for your design. For this tutorial, a completed VHDL testbench file is in the sample files folder in your project.
10. In the Compile HDL Source Files dialog box, navigate to the sample files folder, select `shiftreg16_tb.vhd`, and click Compile. Your ModelSim view should look like this.

![ModelSim view](image)

11. Click **Done** to close the Compile HDL Source Files dialog box.
Task 5: Simulate the Design and View the Waveforms

The ModelSim simulator can load and simulate VHDL designs, providing a uniform graphic interface and simulation control commands for debugging and analyzing your designs. Before you simulate the design, you have to load the test fixture file into ModelSim. After simulation, you can use the Wave window to display waveforms and VHDL nets and register variables you have selected.

To simulate the design and view the waveforms:

1. Choose **Simulate > Simulate** to open the dialog box.

2. On the Design tab, open the work directory, and select a testbench file (**testbench**).

3. Click **Add** to add the test fixture and the architecture to the timing simulation.

4. Click **Load**. ModelSim loads your design units and displays the results on the transcript window.
5. Choose **View > Signals** to open the Signals window. This window shows the VHDL nets and register variables in the region currently selected in the Structure window.

6. In the Signals window, choose **Add > Wave > Signals in Region** to open the Wave window. This window displays waveforms and VHDL nets and register variables you have selected. Current and past simulations can be compared side-by-side in one Wave window.
7. Now you are ready to run the simulation. In the ModelSim main transcript window, type `run 1000 ns`, and then press Enter.

8. ModelSim runs the simulation and adds the signals to the Wave window, although you will have to change the view to see them properly.

9. In the Wave window, choose `View > Zoom > Zoom Full`. You can use this window to check the logic.

10. Quit ModelSim.
Task 6: Instantiate the VHDL LPM Module

When the Module/IP Manager builds the VHDL module, it produces output files for generating your design’s database and places them inside your project folder. Before you generate the database for your VHDL design, you must port map the module signals to your top-level design signals.

**IMPORTANT:** This procedure is for reference. We have provided a completed top-level design file in the project sub-directory (`\sample files\toplevel.vhd`) that includes proper port mapping and an attribute statement for Synplify. You can copy this file into your project directory, or you can view this file for reference while you complete this task.

1. In the Project Navigator, choose **Window > Text Editor**.
2. In the Text Editor, choose **File > Open** to open the dialog.
3. In the dialog box:
   - In the Files of Type dropdown list, select **VHDL Files (*.vhd)**.
   - Select your own top-level design file and click **Open**.
4. Open the module file that you generated with Module/IP Manager (`shift_left16.vhd`).
5. In your top-level design file, port map the signals of the module component to your top-level design signals.

   **Note:** Make sure that the signal labels match those of the generated module. If you changed any of the signal labels in the Module/IP Manager, you will need to change them in the port map clause as well.

6. Save your design file. The Project Navigator lists the module inside the top-level design in the Sources window.
Task 7: Import Source Files

You "describe" a project by specifying the project files that will represent the design. You can either import an existing source or create a new one. The added source appears in hierarchical and alphabetical order in the Sources window.

1. In the Project Manager, choose **Source > Import** to open the Import File dialog box.

   **Note:** Notice the title of the dialog box identifies the project type as (Schematic/VHDL). Therefore, even though the Module/IP Manager generated both Verilog HDL and VHDL files, you will see only those files that pertain to the project type.

2. Select `shift_left16.vhd` and click **Open**.

3. In the Import Source Type dialog box, select **VHDL Module** and click **OK**.

4. Choose **Source > Import** again and navigate to the sample files folder.

5. Select `toplevel.vhd` and click **Open**.

6. In the Import Source Type dialog box, select **VHDL Module** and click **OK**.

   The ispLEVER software imports the sources into the project and displays them in the Sources window.

   ![Image of the Project Manager](image)

   **Note:** An alternative to importing HDL source files generated by the Module/IP Manager into the Project is to import the module's LPC file. After the file is imported, you can double-click the LPC file to cause the Module/IP Manager to automatically open and load the module's configuration.
Task 8: View the Routing Results with the Floorplanner

After an initial Lattice internal database is generated, the Pack & Place Design process packs the design instances or cells into Generic Logic Blocks (GLBs) and then places them on the ispXPGA device. The output of this process is the post-place design database and a Pack/Place Report.

The ispLEVER software routes the design in the ispXPGA after placement. The routing algorithm takes full advantage of the Lattice ispXPGA architecture to achieve maximum performance. It uses congestion-driven routing to achieve a fit with minimal congestion, and it uses timing-driven routing to achieve maximum performance. The output of this process is the post-route design database and a Route Report.

The Floorplanner allows you to view the results of Place/Pack Report and Route Report in a graphical form. The Floorplanner graphical display of device architecture, and design pack/placement, and design routing, includes:

- Device pin layout and associated design signals
- Customizable color-coded device utilization at the GLB level
- Customizable color-coded routing congestion at the column and row channel level
- GLB configuration in logic-equivalent function
- Design frequency (Fmax), longest arrival time (Tpd), register timing data (Tsu & Tco) and corresponding timing path at the GLB level
- Routed and un-routed signals (nets or instance terminals)

To route the design and view the results with the Floorplanner:

1. In the Project Navigator, double-click the Post-Route Design Floorplan process to pack, place, and route the design. When the processes are through running, the Floorplanner opens.

2. The Floorplan View is one of the two windows available in the Main Window. The Floorplan View displays the physical placement of your logic. In this tutorial design example, only the `shift_left16` module is in the design.
3. Select the **Floorplan View window** to make it active.

4. Choose **Edit > Select Used** to select the used (placed) cells.

5. On the toolbar, click the Show Fanin and Fanout icons.

This view in the Floorplan View window shows the placed logic, as well as input, output, and hardwire signals.
6. Exit the Floorplanner.

**See Also**

To learn more about other Floorplanner features, take the "Debugging Designs with the Floorplanner" tutorial.
Task 9: Run Static Timing Analysis

Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device’s timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To run timing analysis and view the results:

1. In the Project Navigator Sources window, select the target device.
2. In the Processes window, double-click the Performance Analyst process to pack, place, and route the design; run timing analysis; and open the Performance Analyst.
3. The Performance Analyst performs eight distinct analysis types. The first type, \texttt{fMAX}, is an internal register-to-register delay analysis. \texttt{fMAX} measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining seven types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

Under Analysis, select \texttt{fMAX}, and then click \texttt{Run}. In this example, the maximum frequency is \textbf{471.7 MHz} and the longest path delay is \textbf{2.12 ns}.

\textit{Note: Your numbers may vary slightly.}
4. Click the highlighted cell (2.12) in the spreadsheet window to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown are the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.
5. Exit the Performance Analyst without saving.
Congratulations

You have completed the *VHDL Design with LPMs Using the Module/IP Manager* tutorial. In this tutorial you have learned how to:

- Use ispLEVER to create a new Schematic/VHDL project and target a device.
- Generate an LPM module using the Module/IP Manager.
- Compile the files for ModelSim functional simulation.
- Simulate the design and view the waveforms.
- Instantiate the VHDL LPM module.
- Import source files into the Project Navigator.
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst and view the results.
Module 4: Verilog HDL Design with LPMs Using the Lattice PMI

The ispLEVER software lets you choose the most convenient method for placing parameterized modules and IP cores into your design database – the Module/IP Manager method and the Parameterized Module Instantiation (PMI) method. In both methods the software produces the instantiation template and its associated files and saves all of them in the project directory.

The PMI template gives you the flexibility of editing a module at any time. When your design changes or the software changes, all you have to do is edit the design file using a text editor. You don’t have to recreate the module.

With the PMI approach, you embed the core description of your module into your HDL source files. The ispLEVER Build Database process converts the EDIF file created from the synthesis program to the Lattice internal database. If a module is detected in the source code, it is automatically expanded in this process, producing the module and required output files.

Assumptions
You are already familiar with the Verilog HDL format hardware description language.

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

- Use ispLEVER to create a new Schematic/Verilog HDL project and target a device
- Import source files into the Project Navigator
- Instantiate the Verilog LPM module in source file using Lattice Text Editor
- Synthesize the design using LeonardoSpectrum
- Generate the Lattice database
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst to view the results

Time to Complete This Tutorial
The time to complete this tutorial is about 45 minutes.
Task 1: Create a New Project

To begin a new project, you must create a project directory. Then you must give the project file a name and declare the project type.

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

In this module, we have already created a project directory (module4) for you.

To create a new project:

1. Start the ispLEVER software, if it is not already running.
2. In the Project Navigator, choose File > New Project to open the Create New Project dialog box.
3. In the dialog box,
   - Change to the directory:
     `<install_path>\ispcpld\examples\tutorial\tutorial6\module4`.
   - In the Project name box, type `verilog_pmi.syn`.
   - In the Project type box, select Schematic/Verilog HDL.
   - Click Save.

The default project name, Untitled, appears in the Sources window of the Project Navigator.
4. Double-click the project title (Untitled) to open the Project Properties dialog box.

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

5. Type **Verilog PMI Module** as your project title and click **OK**.
Task 2: Target a Device

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

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

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

2. In the dialog box:
   - Under Select Device, select ispXPGA from the Family box.
   - Under Select Device, select LFX1200C from the Device box
   - Accept the default settings and click OK.

3. In the Confirm Change message dialog box, click Yes to continue with the operation.
4. In the next message dialog box, click **No** to retain any previous constraints.

5. Your Project Navigator should look like this:
Task 3: Import Source File

You "describe" a project by specifying the project files that will represent the design. You can either import an existing source or create a new one. The added source appears in hierarchical and alphabetical order in the Sources window.

To import source files:

1. Copy the two files, `top_lpm.v` and `prepl.v`, from the sample files folder, into the module 4 folder.
2. In the Project Navigator, choose Source > Import to open the Import File dialog box.
3. Select `top_lpm.v` and click Open.

   The ispLEVER software imports the top-level design (`top_lpm.v`), which includes the LPM_Mult module `mult8x8` and the lower-level design `prepl`. All of these files are displayed in the Sources window.

Your Project Navigator should look like this.
Task 4: Instantiate the LPM Module

When you create a module with the Parameterized Module Instantiation template, you use a text editor to copy and paste the template into your lower-level design and set the parameters for the module. After saving the module with a unique Verilog file name, you then use the Text Editor to instantiate the module in your top-level design.

To instantiate a Verilog module using the PMI template, you must do the following:

- Write the boundary description of the module in your top-level design
- If you are using the Synplify synthesis tool, include a synthesis attribute statement
- Map the port names of your top-level design with the actual module names

For this exercise, you will examine the instantiation of the LPM_mult that you imported into your project in the previous task.

To examine the instantiation of the top design:

1. Double-click the `top_lpm` file to open it in the Text Editor.

   The design has been instantiated correctly, as shown in the image below.
Note: You do not have to include a synthesis statement if you are using the LeonardoSpectrum synthesis tool.

2. Choose File > Exit to close the Text Editor.
Task 5: Synthesize the Design

The Project Navigator allows you to synthesize your design automatically and seamlessly. For Verilog and VHDL designs, the Project Navigator provides two synthesis tools that are integrated into the Project Navigator: Synplicity Synplify and Mentor Graphics LeonardoSpectrum.

To synthesize the design automatically:

1. Choose Options > Select RTL Synthesis.

2. Select LeonardoSpectrum to associate the current device family with the LeonardoSpectrum synthesis tool. To use the same synthesis tool for all designs, also select As Default.

   Note: If you select the Synplify synthesis tool, the attribute statement must be included in the top-level design file, as shown in the previous task.

3. Click OK. The Project Navigator will now synthesize your design automatically prior to generating the database or implementing the design.
**Task 6: Build the Database**

This process automatically invokes the synthesis process and converts the EDIF file from the synthesis program to the Lattice internal database. Any Lattice Modules utilized in the design are expanded in this process.

*To build the database:*

1. Select the device in the Sources window of the Project Navigator.
2. In the Processes window, double-click **Build Database**.

*Note:* This process may take a minute. If you receive a message that warnings were generated, just click **OK**.

The Project Navigator generates the database and indicates, in the Processes window, that the process has been completed.
Task 7: Route the Design and View the Results

After an initial Lattice internal database is built (Build Database process), the Pack & Place Design process packs the design instances or cells into Programmable Functional Units (FPUs) and then places them on the FPGA device. The output of this process is the post-place design database and a Pack/Place Report.

The ispLEVER software routes the design in the FPGA after placement. The routing algorithm takes full advantage of the Lattice FPGA architecture to achieve maximum performance. It uses congestion-driven routing to achieve a fit with minimal congestion, and it uses timing-driven routing to achieve maximum performance. The output of this process is the post-route design database and a Route Report.

The Floorplanner allows you to view the results of Place/Pack Report and Route Report in a graphical form. The Floorplanner graphical display of device architecture, and design pack/placement, and design routing, includes:

- Device pin layout and associated design signals
- Customizable color-coded device utilization at the PFU level
- Customizable color-coded routing congestion at the column and row channel level
- PFU configuration in logic-equivalent function
- Design frequency (Fmax), longest arrival time (Tpd), register timing data (Tsu & Tco) and corresponding timing path at the PFU level
- Routed and un-routed signals (nets or instance terminals)

To route the design and view the results with the Floorplanner:

1. In the Project Navigator, double-click the Post-Route Design Floorplan process to pack, place, and route the design. When the processes are through running, the Floorplanner opens.

   Note: This process may take a few minutes. If you receive a message that warnings were generated, just click OK. Also, you can close the Report Viewer.

2. The Floorplan View is one of the two windows available in the Main Window. Select the Floorplan View window to make it active.

3. Choose View > Zoom In and click several times in the upper-left corner of the Floorplan View window until you can see the physical pattern of logic for the module. (Used PFUs are shown in blue.)
4. Choose **Edit > Select Used** to select the used (placed) cells. Notice they have changed color to visually indicate they are all selected.

5. On the toolbar, click the **Show Fanin** and **Show Fanout** icons.

This view in the Floorplan View window shows the placed logic, as well as input, output, and hardwire signals.
6. Choose **File > Exit** to exit the Floorplanner.

**See Also**

To learn more about other Floorplanner features, take the "Debugging Designs with the Floorplanner" tutorial.
Task 8: Run Static Timing Analysis

Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device’s timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To run timing analysis and view the results:

1. In the Project Navigator Sources window, select the target device.
2. In the Processes window, double-click the Performance Analyst process to run static timing analysis and open the Performance Analyst.
3. The Performance Analyst performs eight distinct analysis types. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining seven types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

Under Analysis, select fMAX, and then click Run. In this example, the maximum frequency is 327.9 MHz and the longest path delay is 3.05 ns.

**Note:** Your numbers may vary slightly.
4. Click the highlighted cell (3.05) in the spreadsheet window to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown are the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.
5. Click **OK** to close the Expanded Path dialog box.

6. Choose **File > Exit** to close the Performance Analyst without saving.

7. Choose **File > Exit** to close ispLEVER. You do not have to save the design.
Congratulations
You have completed the Verilog HDL Design with LPMs using the Lattice PMI tutorial. In this tutorial you have learned how to:

- Use ispLEVER to create a new Schematic/Verilog HDL project and target a device
- Import source files into the Project Navigator
- Instantiate the Verilog LPM module in source file using Lattice Text Editor
- Synthesize the design using LeonardoSpectrum
- Build the Lattice database
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst to view the results
Module 5: VHDL Design with LPMs Using the Lattice PMI

The ispLEVER software lets you choose the most convenient method for placing parameterized modules and IP cores into your design database – the Module/IP Manager method and the Parameterized Module Instantiation (PMI) method. In both methods the software produces the instantiation template and its associated files and saves all of them in the project directory.

The PMI template gives you the flexibility of editing a module at any time. When your design changes or the software changes, all you have to do is edit the design file using a text editor. You don’t have to recreate the module.

With the PMI approach, you embed the core description of your module into your HDL source files. The ispLEVER Build Database process converts the EDIF file created from the synthesis program to the Lattice internal database. If a module is detected in the source code, it is automatically expanded in this process, producing the module and required output files.

Assumptions
You are already familiar with the VHDL format hardware description language.

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

- Use ispLEVER to create a new Schematic/VHDL project and target a device
- Import source files into the Project Navigator
- Instantiate the VHDL LPM module in source file using Lattice Text Editor
- Synthesize the design
- Build database
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst and view the results.

Time to Complete This Tutorial
The time to complete this tutorial is about 45 minutes.
Task 1: Create a New Project

To begin a new project, you must create a project directory. Then you need to give the project file a name (.*syn) and declare the project type (Schematic/VHDL).

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

To create a new project:

1. Start the ispLEVER software, if it is not already running.
2. In the Project Navigator, choose File > New Project to open the Create New Project dialog box.
3. In the dialog box,
   - Change to the directory: <install path>\ispclbl\examples\tutorial\tutorial6\module5.
   - In the Project name box, type vhdl_pmi.syn
   - In the Project type box, select Schematic/VHDL.
   - Click Save.
The default project name, Untitled, appears in the Sources window of the Project Navigator.

4. Double-click the project title (Untitled) to open the Project Properties dialog box.

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

5. Type VHDL PMI Module as your project title and click OK.
Task 2: Target a Device

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

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

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

   ![Device Selector Dialog Box]

2. In the dialog box:
   - Under Family, make sure ispXPGA is selected from the drop-down list.
   - Under Device, select LFX1200C.
   - Accept the default settings and click OK to close the dialog box.
3. In the Confirm Change message dialog box, click **Yes** to continue with the operation.

![Confirm Change dialog box](image1.png)

4. In the next message dialog box, click **No** to retain any previous constraints.

![Project Navigator dialog box](image2.png)

5. Your Project Navigator should look like this:

![Project Navigator](image3.png)
Task 3: Import Source Files

You "describe" a project by specifying the project files that will represent the design. You can either import an existing source or create a new one. The added source appears in alphabetical order in the Sources window.

To import source files:
1. In the Project Navigator, choose Source > Import to open the Import File dialog box.
2. Open the sample files folder and select the following files:
   - top_lpm.vhd
   - prep1.vhd
3. Right-click the selected files and choose Copy
4. Navigate back to the tutor6/module5 directory, right-click and choose Paste, and then click Open.
5. In the Import Source Type dialog, select Module for each of these files and click OK.

The ispLEVER software imports the top-level design top_lpm.vhd and the lower-level design prep1.vhd. These designs are displayed in the Sources window. Your Project Navigator should look like this.
Task 4: Instantiate the VHDL LPM Module in Source File

When you create a module with the Parameterized Module Instantiation template, you use the text editor to copy and paste the template into your lower-level design and set the parameters for the module. After saving the module with a unique vhdl file name, you then use the text editor to instantiate the module in your top-level design.

To instantiate a vhdl module using the PMI template, you must do the following:

- Write the component declaration for the module in your top-level design file
- If you are using the Synplify synthesis tool, include a synthesis attribute statement
- Map the port names of your top-level design with the actual module names

For this exercise, you will examine the instantiation of the top_lpm.vhd design file that you imported in the previous task.

To examine the instantiation of the top design:

1. In the Sources window, double-click the top_lpm design to open it in the text editor.
   
   The design has been instantiated correctly, as shown in the image below.
**Note:** You do not have to include a synthesis statement if you are using the Leonardo Spectrum synthesis tool.

2. Close the text editor.
Task 5: Synthesize the Design

The Project Navigator allows you to synthesize your design automatically and seamlessly. For Verilog and VHDL designs, the Project Navigator provides two synthesis tools that are integrated into the Project Navigator: Synplicity Synplify and Mentor Graphics LeonardoSpectrum.

To synthesize the design automatically:

1. Choose Options > Select RTL Synthesis.

![Select RTL Synthesis dialog box]

2. Select the synthesis tool you want to use. To use the same synthesis tool for all designs, also select As Default.

   **Note:** If you select the Synplify synthesis tool, the attribute statement must be included in the top-level design file, as shown in the previous task.

3. Click OK.

   The Project Navigator will now synthesize your design automatically prior to building the database or placing and routing.
Task 6: Build the Database

This process invokes the synthesis process and converts the EDIF file from the Synthesis program to the Lattice internal database. Any Lattice Modules utilized in the design are expanded in this process.

To build the database:
1. Select the device in the Sources window of the Project Navigator.
2. In the Processes window, double-click Build Database.

Note: This process may take a minute. If you receive a message that warnings were generated, just click OK.

The Project Navigator builds the database and indicates, in the Processes window, that the process has been completed.
Task 7: View the Routing Results with the Floorplanner

After an initial Lattice internal database is built, the Pack & Place Design process packs the design instances or cells into Generic Logic Blocks (GLBs) and then places them on the FPGA device. The output of this process is the post-place design database and a Pack/Place Report.

The ispLEVER software routes the design in the FPGA after placement. The routing algorithm takes full advantage of the Lattice FPGA architecture to achieve maximum performance. It uses congestion-driven routing to achieve a fit with minimal congestion, and it uses timing-driven routing to achieve maximum performance. The output of this process is the post-route design database and a Route Report.

The Floorplanner allows you to view the results of Place/Pack Report and Route Report in a graphical form. The Floorplanner graphical display of device architecture, and design pack/placement, and design routing, includes:

- Device pin layout and associated design signals
- Customizable color-coded device utilization at the GLB level
- Customizable color-coded routing congestion at the column and row channel level
- GLB configuration in logic-equivalent function
- Design frequency (Fmax), longest arrival time (Tpd), register timing data (Tsu & Tco) and corresponding timing path at the GLB level
- Routed and un-routed signals (nets or instance terminals)

To route the design and view the results with the Floorplanner:

1. In the Project Navigator, double-click the **Post-Route Design Floorplan** process to pack, place, and route the design. When the processes are through running, the Floorplanner opens.

   **Note:** This process may take a few minutes. If you receive a message that warnings were generated, just click **OK**.

2. The Floorplan View is one of the two windows available in the Main Window. Select the **Floorplan View window** to make it active.

3. Choose **View > Zoom In** and click several times in the floorplan until you can see the physical pattern of logic for the module.
4. Choose **Edit > Select Used** to select the used (placed) cells.

5. On the toolbar, click the Show Fanin and Fanout icons.

6. This view in the Floorplan View window shows the placed logic, as well as input, output, and hardwire signals.
7. Exit the Floorplanner.

**See Also**

To learn more about other Floorplanner features, take the "Debugging Designs with the Floorplanner" tutorial.
Task 8: Run Static Timing Analysis

Static timing analysis is the process of verifying circuit timing by totaling the propagation delays along paths between clocked or combinational elements in a circuit. The analysis can determine and report timing data such as the critical path, setup/hold time requirements, and the maximum frequency.

The Performance Analyst traces each logical path in the design and calculates the path delays using the device’s timing model and worst-case AC specs supplied in the device data sheet.

The timing analysis results are displayed in a graphical spreadsheet with source signals displayed on the vertical axis and destination signals displayed on the horizontal axis. The worst-case delay value is displayed in a spreadsheet cell if there is at least one delay path between the source and destination. To more easily identify performance bottlenecks, you can double-click a cell to view the path delay details.

To run timing analysis and view the results:

1. In the Project Navigator Sources window, select the target device.
2. In the Processes window, double-click the Performance Analyst process to pack, place, and route the design; run timing analysis; and open the Performance Analyst.
3. The Performance Analyst performs eight distinct analysis types. The first type, fMAX, is an internal register-to-register delay analysis. fMAX measures the maximum clock operating frequency, limited by worst-case register-to-register delay. The remaining seven types are external pin-to-pin delay analysis. Timing threshold filters, source and destination filters, and path filters can be used to independently fine-tune each analysis.

Under Analysis, select fMAX, and then click Run. In this example, the worst fmax delay is 285.7 MHz, and the longest path delay is 350 ns.

Note: Your numbers may vary slightly.
4. Click the highlighted cell (350) in the spreadsheet window to open the Expanded Path dialog box. This dialog lets you analyze individual timing components used to calculate the timing path. There is a source pin (From) and a destination pin (To). Also shown are the delay type, the delay of that path (Value ns), and the cumulative delay of all the signals.
5. Exit the Performance Analyst without saving.
Congratulations
You have completed the VHDL Design with LPMs Using the Lattice PMI tutorial. In this tutorial you have learned how to:

- Use ispLEVER to create a new Schematic/VHDL project and target a device
- Import source files into the Project Navigator
- Instantiate the VHDL LPM module in source file using Lattice Text Editor
- Synthesize the design
- Build the database
- Pack, place, and route the design and use the Floorplanner to view the results
- Run static timing analysis using the Performance Analyst and view the results.