ispLeverCORE™

IP Module Evaluation Tutorial

Table Of Contents

Getting Started 2
Other Tutorial Versions and Formats 2
Supported Technologies 2
Tutorial Location 2
Directory Path Description 2
Evaluation Pack Directory Structure 3
Installing IP Graphical User Interface 4
Evaluation Tutorial Flow 5
Instantiating the Evaluation Core 6
Verilog Designs 6
VHDL Designs 6
Running Functional Simulation Using ModelSim 8
Running Synthesis Using Synplify 9
Running Synthesis Using LeonardoSpectrum 10
Implementing the IP Module Using ispLEVER 11
For CPLD Devices 11
For ispXPGA Devices 11
For ORCA/FPSC/EC/ECP/XP Devices 11
Timing Report File 11
Purchasing an ispLeverCORE 13
After Purchasing 13
Configuring your ispLeverCORE 14
Getting Technical Assistance 18
Hotline 18
E-mail 18
Internet 18
Evaluating Other IP Modules 18

Getting Started

Lattice's ispLeverCORE™ IP modules are large, modular design blocks that can be reused and easily placed within your programmable logic design.

Other Tutorial Versions and Formats

This "quick start" tutorial is designed to help you evaluate and purchase the IP module package you have chosen as quickly as possible. Admittedly, it is brief and intended for users that are familiar with all the required tools and processes. Other versions and formats are available, as described below:

- A printable PDF version of this tutorial is available from the Lattice website.

Supported Technologies

Lattice ispLeverCORE IP modules are available for ispXPLD, ispXPGA, ORCA4 FPGA, ORCA FPSC, LatticeEC/ECP, and LatticeXP technologies. This tutorial is written to support all device technologies, with differences noted where important.

Tutorial Location

This tutorial contains links to additional information in the evaluation core's `readme.htm` file necessary to complete the tasks. For these links to work properly, the tutorial must be installed in the IP download folder, for example `<reeds_enco_o4_1_004>\tutorial`.

Directory Path Description

To accommodate all ispLeverCORE IP module download packages, the directory paths are described as `<variables>`. For example, the actual directory path for the Reed-Solomon Encoder for an ORCA4 device is:

```
reeds_enco_o4_1_004\orca4\ver1\source
```

However, the path name in the tutorial would be represented as:

```
<download_name>\<device>\<version>\source
```

Evaluation Pack Directory Structure

The directory structure for the Lattice ispLeverCORE evaluation package is shown below and includes the basic description of the contents.

<download_name>
|
|--\`readme.htm` (Contains specific instructions for using the ispLeverCORE.)
|--\`default.css` (Style sheet for the readme)
|--\<*tutorial*> (Empty tutorial folder)
|
|--***<device>***
|
|--***<version>***
| |
| |--**eval** (Customer RTL functional simulation directory for evaluation)
| | |
| | |--**readme.txt** file contains specific simulation information
| | |
| |--**testbench** Contents: testbench for evaluation
| |
| |--**tests** Contents: Stimulus file(s)
| |
| |--**simulation** (Run/execute functional RTL simulation here.
| | | Example: `do script\runsim_rtl.do`)
| | |
| | |--**scripts** Contents: ModelSim macro (`*.do`) to run simulation
| |
| |--**gui_script** Contents: `module_gen.zip` (See *Note* below)
| |
| |--**par** Contents:
| | (1) `.lpc` file specifying IP module configuration
| | (2) One of the following database/constraint file pairs:
| | ispXPGA – `.ld2` database file and `.lct` constraint file
| | ORCA 4/FPSC/EC/ECP/XP – `.ngo` or `.nmc` database file and `.prf` constraint file
| | ispXPLD – `.bl1` database file and `.lct` constraint file
| |
| |--**source** Contents:
| | (1) Verilog and/or VHDL top-level source files for instantiation and synthesis
| | (2) Parameter file(s) that contain IP-specific configuration values
| |
| |--**lib**
| | |
| | |--**modelsim**
| | | |
| | | |--**work** (Compiled simulation models for ModelSim)

Installing IP Graphical User Interface

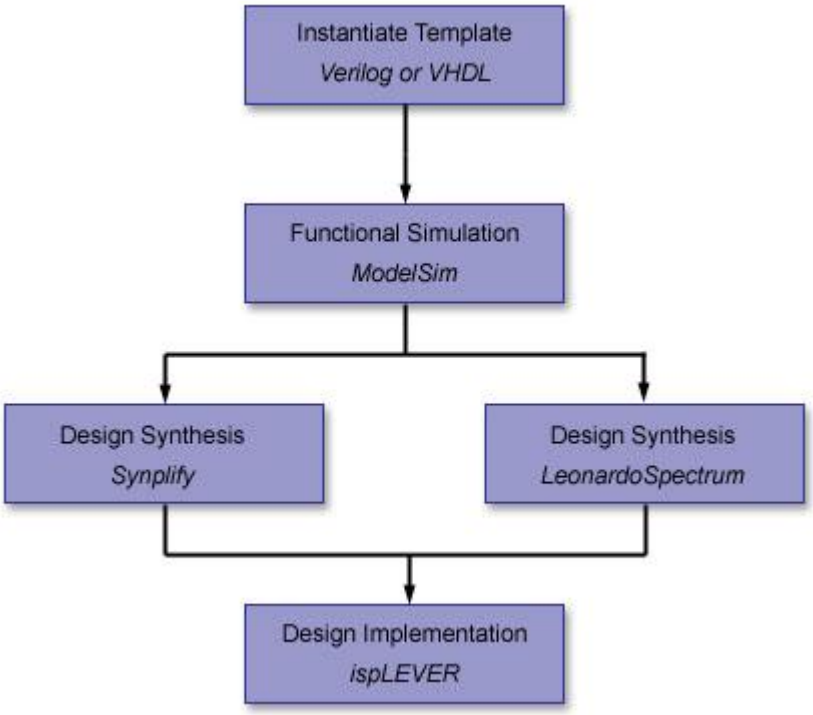
If the `gui_script\module_gen.zip` file is present in this release package, it is because the ispLEVER software does not contain the utility that allows for the configuration of this specific IP module using the Module/IP Manager. The `gui_script\module_gen.zip` file contains a directory of files that allows for the configuration of this IP module using the Module/IP Manager.

To configure the IP module using the ispLEVER Module/IP Manager:

1. Go to the directory in where the ispLEVER software is installed.
2. Unzip the `module_gen.zip` file into the `\ispcpld` folder.
3. If you are prompted to overwrite some existing files, respond "Yes."
4. If the ispLEVER Project Navigator is open, exit and rerun the tool. The IP module will be included in the list when the Module/IP Manager is run.

Evaluation Tutorial Flow

You must run this tutorial in its organized sequence. The only flow option is the synthesis tool you are using. You can return to this flow diagram by clicking the **Flow** button at the end of each lesson. You can also navigate this tutorial using the Contents pane on the left side of your browser.



Instantiating the Evaluation Core

The IP module evaluation package includes a top-level RTL source (Verilog and/or VHDL) that can be used as an instantiation template for the IP core.

Verilog Designs

To instantiate a Verilog module:

1. Using a text editor, open your top-level design file.
2. Open the top-level RTL source located in the `\source` directory of the Evaluation Pack and copy the contents into your top-level design.
3. Connect the ports to the IP module by replacing the default port names in the I/O section of the instantiation template with the actual port names from your design.

   Note: If the top-level RTL source in the package contains any instantiated PLL and/or specific I/O types, those modules must also be instantiated in your top-level design.

4. Save your top-level design file.

   Note: If you want to check the core implementation result for core evaluation purposes, the included top-level RTL source can be used as your top-level design without modification.

VHDL Designs

To instantiate a VHDL module:

1. Using a text editor, open your top-level design file.
2. Open the top-level RTL source located in the `\source` directory of the Evaluation Pack and copy the contents into your top-level design.
3. Connect the ports to the IP module by replacing the default port names in the I/O section of the instantiation template with the actual port names from your design.

   Note: If the top-level RTL source in the package contains any instantiated PLL and/or specific I/O types, those modules must also be instantiated in your top-level design.

4. Include attribute statements for Synplify or Leonardo Spectrum synthesis by typing one of the following, depending upon the synthesis tool you have chosen:

   ----------This is an attribute for Synplify----------

   ```
   attribute syn_black_box: boolean; attribute syn_black_box of
   <module_component>: component is true;
   ```

   ----------This is an attribute for Leonardo Spectrum----------

   ```
   attribute noopt: boolean; attribute noopt of <module_component>:
   component is true;
   ```

   ---

5. Save your top-level design file.

Note: If you want to check the core implementation result for core evaluation purposes, the included top-level RTL source can be used as your top-level design without modification.

Running Functional Simulation Using ModelSim

A simulation script file is provided in the `eval` directory for functional RTL simulation. The script file `<modelsim_macro_name>.do` uses pre-compiled models provided with this package. The pre-compiled library of models are located in the directory `<download_name>\<device>\<version>\lib\modelsim\work`.

NOTE: This procedure is applicable ONLY when using ModelSim SE for simulation. The Lattice Edition of ModelSim that comes with the ispLEVER software does not support the pre-compiled models necessary for functional simulation.

To run functional simulation using ModelSim:

1. Open ModelSim.
2. Choose **File > Change Directory** and go to the `<download_name>\<device>\<version>\eval\`**`simulation`** directory.
3. Run the ModelSim DO (macro) file:

- If you are running version 5.5e or earlier, choose **Macro > Execute Macro** and select the file: `scripts\<modelsim_macro_name>.do`.
- If you are running version 5.6a or later, choose **Tools > Execute Macro** and select the file: `scripts\<modelsim_macro_name>.do`.

The ModelSim macro executes an evaluation test bench designed to show some example transactions or functions associated with the core. Using the precompiled libraries, you can build your own test benches.

4. View the waveform results in the Wave window.

Running Synthesis Using Synplify

This procedure shows you how to use Synplicity's Synplify outside the ispLEVER Project Navigator to synthesize your ispLeverCORE IP module and create an EDIF file.

To run synthesis using Synplify:

1. Create a new working directory for synthesis.
2. Open Synplify.
3. Add the design files specified in your IP download readme.htm file.
4. Set the Implementation Options that are specified in your IP download readme.htm file.
5. Click **Run** to generate an EDIF file.

Note: For more detailed information, refer to the Synplify for Lattice User Guide and the ispLEVER online Help and tutorials.

Running Synthesis Using LeonardoSpectrum

This procedure shows you how to use Mentor Graphics LeonardoSpectrum outside the ispLEVER Project Navigator to synthesize your ispLeverCORE IP module and create an EDIF file.

To run synthesis using LeonardoSpectrum:

1. Open LeonardoSpectrum.
2. Select the target device and options specified in your IP download readme.htm file.
3. Set the Working Directory path pointing to the `<download_name>\<device>\<version>\source` directory.
4. Add the design files specified in your IP download readme.htm file.
5. Set any other options that are specified in your IP download readme.htm file.
6. Click **Run Flow** to generate an EDIF file.

Note: For more detailed information, refer to the LeonardoSpectrum for Lattice User's Manual and the ispLEVER online Help and tutorials.

Implementing the IP Module Using ispLEVER

This procedure shows you how to implement the IP module using the ispLEVER Project Navigator and read the timing report.

To implement the IP module and generate the timing report:

1. Open ispLEVER and create an EDIF project in a new place & route directory.
2. Select the target device specified in your IP download readme.htm file (same as you used for synthesis).
3. Delete the existing constraint file created when starting a new project.
4. Copy the evaluation database and constraint files from the `<download_name>\<device>\<version>\par` directory to your place & route directory. These files are specified in your IP download readme.htm file
5. Rename the copied constraint file to the same name as your project. For example, if your project is named "`demo.syn`," then the constraint file must be named `demo.lct` or `demo.prf`.
6. Choose **Source > Import** and select the EDIF netlist file.

For CPLD Devices

7. Double-click the **Timing Report** process to open the Timing Report.
8. You are finished with the evaluation tutorial.

For ispXPGA Devices

7. Double-click the **Post-Route Timing Report** process to open the Timing Report.
8. You are finished with the evaluation tutorial.

For ORCA/FPSC/EC/ECP/XP Devices

7. Choose **Tools > Timing Checkpoint Options** and set the options as specified in your IP download readme.htm file.
8. Select the **Place & Route Design** process, right-click to open the Properties dialog box, and then set the properties as specified in your IP download readme.htm file.
9. Double-click the **Place & Route Trace Report** process to run place and route and to open the Timing Report.

Note: For some ORCA4 ispLeverCOREs, the Cycle Stealing Process may be required to achieve the required timing specification. Check the ispLeverCORE readme file to see if this additional process step is necessary.

Timing Report File

Some cores are overconstrained to achieve maximum timing performance. The user may observe timing violations when viewing the timing report. To obtain actual results, use the Post Route Trace preference files to generate the correct timing report.

1. Replace the preference file (.lct or .prf) in the project directory with the following Post Route Trace file. Rename the Post Route Trace preference file to match the project name.
   - post_route_trace_synplify.lct or .prf for Synplify
   - post_route_trace_leonardo.lct or .prf for LeonardoSpectrum
2. From the GUI of the Project Navigator, right-click on **Post_Route_Timing_Report** (for XPGA) or **Place_Route_Trace_Report** (for ORCA/ORCA FPSC/EC/ECP/XP). The new timing file is generated.

Purchasing an ispLeverCORE

If you are satisfied with the results of the evaluation you may purchase Lattice's ispLeverCORE Intellectual Property modules through your local Lattice Sales Office.

Lattice's ispLeverCORE IP modules are sold in either bitstream format (for part numbers ending in "B"), or 1 configuration of a post-synthesis, gate-level netlist (part numbers ending in "N"). Each ispLeverCORE is sold with 1 year or 10 hours of technical support, whichever comes first. Extended service contracts are also available.

For IP purchasing in general, you will follow these steps:

1. Identify the IP module that you want to purchase.
2. Download Lattice's standard Intellectual Property License Agreement from the Lattice website (http://www.latticesemi.com/products/devtools/ip/levercore.cfm).
3. Review, sign, and deliver the License Agreement to your local Lattice Sales Office.
4. Select the parameters for your IP module and send the configuration file to your local Lattice Sales Office.

After Purchasing

After you have purchased and licensed your ispLeverCORE, you can continue the implementation and programming flow. You will be able to perform full timing simulation and generate a bitstream file for programming your Lattice device.

Configuring your ispLeverCORE

You can use Lattice's Module/IP Manager tool (included with Lattice's ispLEVER software) to specify the IP module parameter settings. This will generate a Lattice Parameter Configuration file (`.lpc`). You must send us this LPC file before we can send you your configured IP module. Please send it to the Lattice location listed in the confirmation e-mail. As soon as we get this file, we will e-mail you a confirmation of receipt and a scheduled ship date.

The Lattice IP parameterization tool, Module/IP Manager, is incorporated in the ispLEVER software. It provides a GUI for entering the required parameters to configure the core. After all required parameters have been entered, the following file are generated:

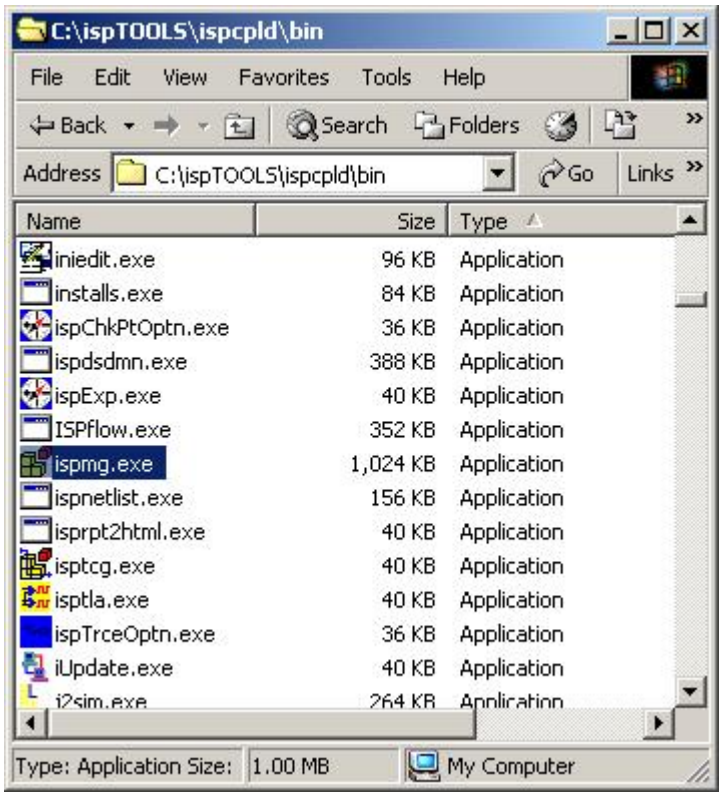
- Parameter file (`<IPname>.lpc`)

This file contains the configuration parameters you entered via the Module/IP Manager GUI. Usually the Module/IP Manager is run from the ispLEVER Project Navigator. *However, for generating your IP module configuration file, you will run the Module/IP Manager in standalone mode.*

Note: This procedure uses the Reed-Solomon Encoder for a Verilog design implemented in a Lattice ORCA 4 device as an example. Your specific settings will be different.

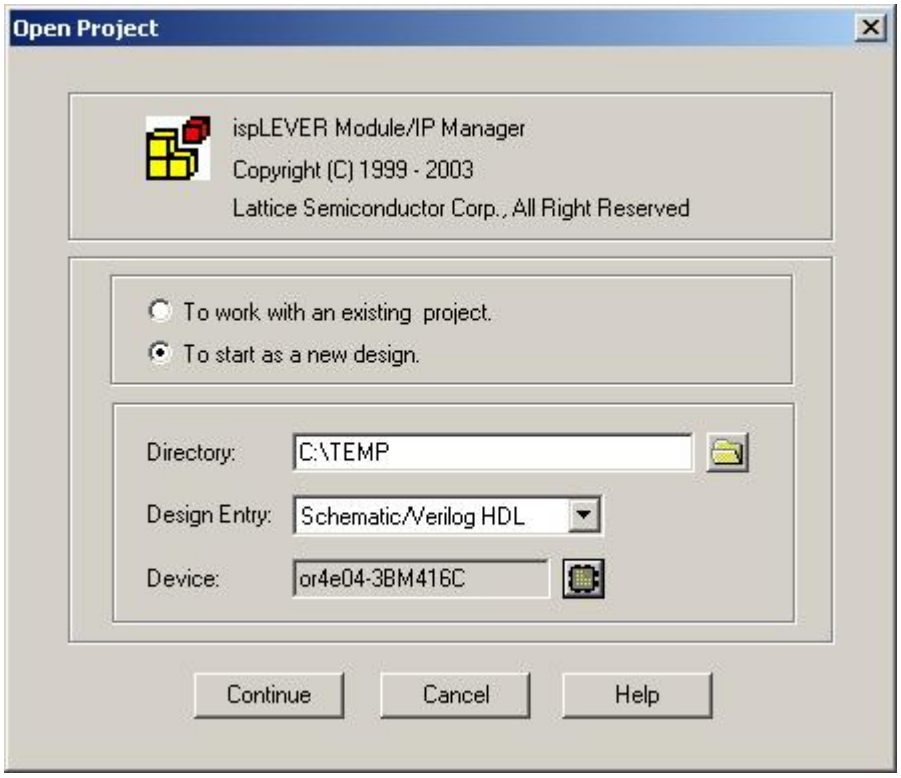
To use the Module/IP Manager to generate an LPC file:

1. Go to the directory where your Module/IP Manager application is installed, for example:
   `<isplever_install_path>\ispcpld\bin`
2. Double-click **ispmg.exe**.

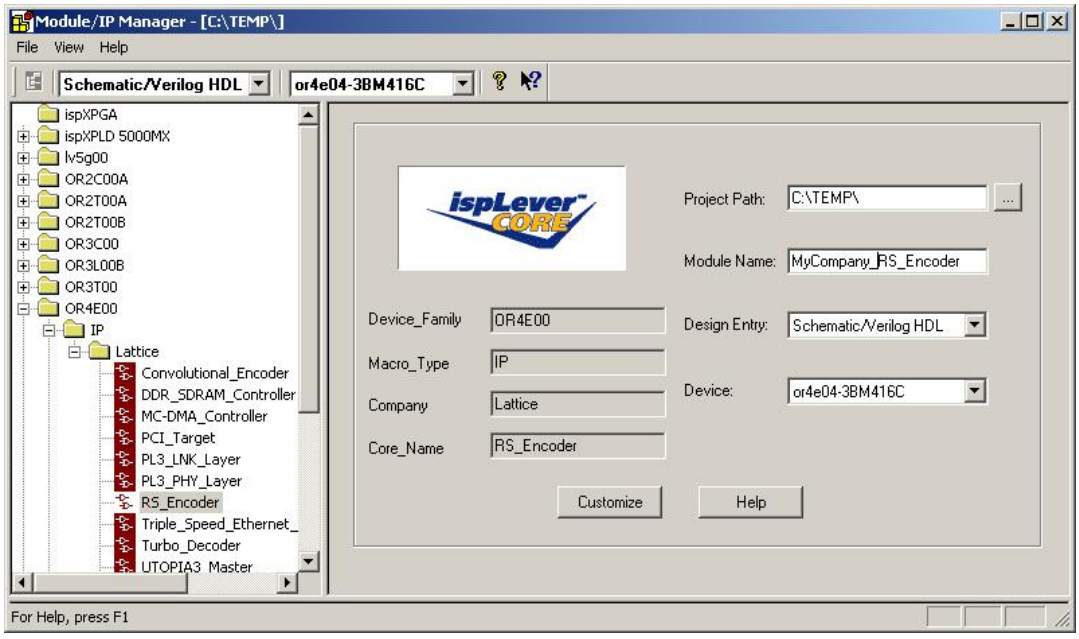


3. In the Open Project dialog box, do the following:

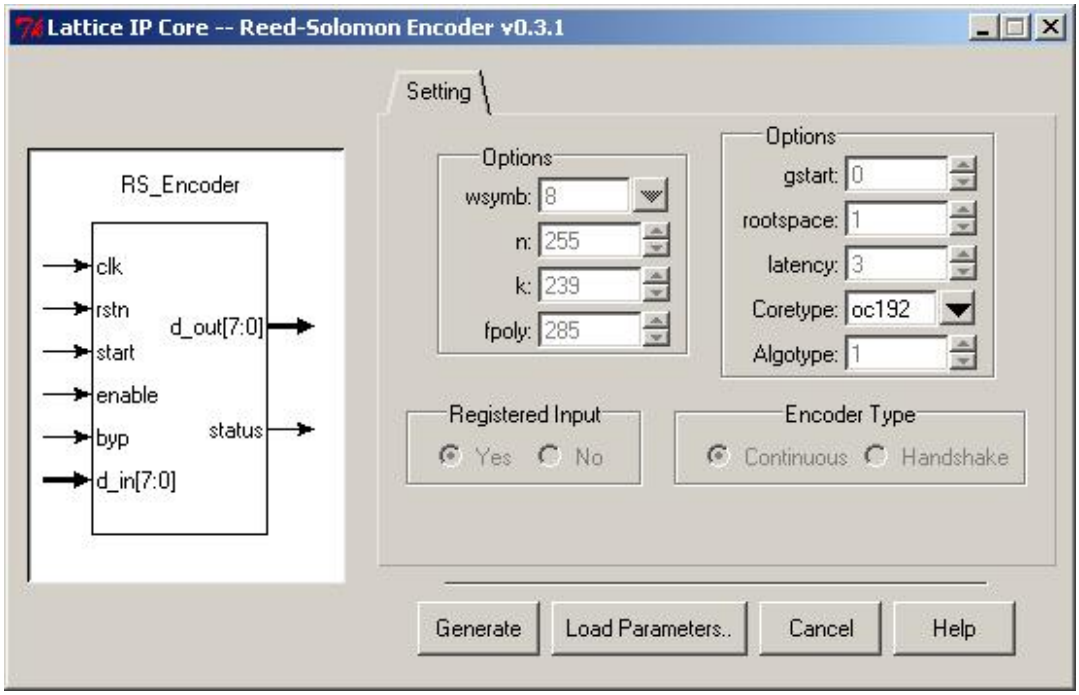
- Select **To start a new design**.
- Set the directory path for the Lattice Parameter Configuration file (`.lpc`). This can be any location.
- Select the netlist format.
- Select your target device.



4. Click **Continue** to load your selections into the Module/IP Manager and open the GUI. In the GUI, do the following:
   - Expand tree for the device family you have selected and select the IP module that you want to configure and purchase.
   - Under Module Name, type a name for your IP module. This can be any name you choose. Special characters are allowed.



5. Click **Customize** to open the configuration screen for your chosen IP module. Set the desires options and then click **Generate**.



The Module/IP Manager generates the Lattice Parameter Configuration (LPC) file to the specified location.

6. To verify the parameter settings, click on the "Load Parameters" button and select your saved `.lpc` file. Your saved parameter will load into the Module/IP Manager tool for verification.
7. You must send us the LPC file (`.lpc`) to Lattice so that we can configure the IP core and send it to you. Please send your LPC file to the Lattice location listed in your purchase

confirmation e-mail (or to your local Lattice Sales office). As soon as Lattice receives your LPC file, we will e-mail you a confirmation of receipt and the scheduled shipping date of your ispLeverCORE.

Getting Technical Assistance

For technical support assistance, please contact Lattice Applications at:

Hotline

1-800-LATTICE (Domestic)

1-408-826-6002 (International)

E-mail

techsupport@latticesemi.com

Internet

http://www.latticesemi.com

Evaluating Other IP Modules

To evaluate IP module configurations that are not included in the Evaluation Packages, contact your local Lattice sales office.