Lattice Radiant Software IP User Guide

November 9, 2022

Copyright

Trademarks

Disclaimers

Type Conventions Used in This Document

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

Contents

Lattice Radiant Software IP User Guide

Lattice provides pre-tested, reusable intellectual property (IP) functions. These proven IP cores are optimized for Lattice device architecture, resulting in fast, small cores that use the latest Lattice architectures to their fullest.

Lattice Radiant software provides enhanced IP functionality, which is described in this user guide. Major topics include:

Designing with IP and PMI

IP are functional bits of design that can be re-used wherever that function is needed. Creating such components with hardware design languages is common practice. To help your design along, the Radiant software provides a variety of components for common functions. They are optimized for Radiant software device architectures and can be customized. Use these components to speed up your design work and to get the most effective results.

Radiant software components come in a variety of forms:

- Modules: These basic, configurable blocks come with IP Catalog. They provide a variety of functions including I/O, arithmetic, memory, and more. Open IP Catalog to see the full list of what's available.
- IP: Intellectual property (IP) are more complex, configurable blocks. They are accessible through IP Catalog, but they do not come with the tool. They must first be downloaded and installed as a separate step before they can be accessed from IP Catalog. To see all that's available and to learn about licensing and other vendors of IP, go to the Lattice website: www.latticesemi.com/ip.

- PMI (Parameterized Module Instantiation) is an alternate way to use some of the components that come with IP Catalog. Instead of using IP Catalog, PMI can directly instantiate a component into your HDL and customize it by setting parameters in the HDL. You may find this easier than using IP Catalog if your design requires many variations of the same component. To decide which method to use, see "PMI or IP Catalog?" on page 6.
- Reference designs provide you with a starting point on creating your own components. Lattice Reference Designs are available in Verilog and VHDL, and can be downloaded from the Lattice website: www.latticesemi.com/ip.
- Lattice library primitives are very basic functions, such as logic gates and flip-flops. They can be directly instantiated as HDL into designs. But this is an advanced technique and should usually be avoided.

IP Catalog provides a variety of functions ranging from the most basic, such as arithmetic and memory, to much more complex functions. With IP Catalog these components can be extensively customized and created as part of a specific project or as a library for multiple projects. See "Creating IP Catalog Components" on page 8.

However, many of these components can also be used with PMI (see next item). To decide which method to use, see "PMI or IP Catalog?" on page 6.

PMI or IP Catalog?

Many IP Catalog components are also available as PMI components, but PMI doesn't offer error checking. This and the next two sections discuss the pros and cons of both options to help you decide which is best for your project.

PMI is a convenient way to use components of the same type but that vary from instance to instance. This eliminates the need to create a separate component for each instance using IP Catalog.

For example, a design might require dozens of FIFOs that are functionally the same but require different address depths. With PMI, you could insert the same FIFO instantiation command wherever it's needed in the HDL and just change the address depth parameter as you go. The alternative would be to use IP Catalog to generate different components for each variation and then insert the instantiation template for each of them. In this situation, you might find the PMI method easier and faster.

Before deciding whether to use a PMI component, compare it to the equivalent IP Catalog component. Often IP Catalog provides more options than PMI does. IP Catalog also assures that all of your option selections and parameter settings are legal. PMI offers no error checking.

Parameterized Module Instantiation (PMI) The following table lists Radiant modules that support PMI. The table contains:

- PMI module name.
- IP Catalog module name.

- Links to the IP User Guides in which the PMI modules are described in detail.
- Columns that indicate PMI support for iCE40UP, LFD2NX/LFCPNX/ LFMXO5/LIFCL/UT24C/UT24CP, and LAV-AT-E devices.

Table 1: PMI Modules

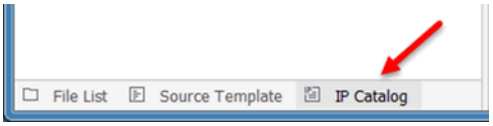
PMI Module Name	IP Catalog Module Name	Lattice Radiant Software User Guide	iCE40UP	LFD2NX, LFCPNX, LFMXO5, LIFCL, UT24C, UT24CP	LAV-AT-E
pmi_add	Adder	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_addsub	Adder Subtractor	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_complex_mult	Complex_Mult	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_counter	Counter	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_dsp	Mult_Accumulate	Arithmetic Modules User Guide	Yes	No	No
pmi_mac	Mult_Accumulate	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_multaddsub	Mult_Add_Sub	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_multaddsubsum	Mult_Add_Sub_Sum	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_mult	Multiplier	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_sub	Subtractor	Arithmetic Modules User Guide	Yes	Yes	Yes
pmi_fifo	FIFO	Memory Modules User Guide	Yes	Yes	Yes
pmi_fifo_dc	FIFO_DC	Memory Modules User Guide	Yes	Yes	Yes
pmi_distributed_shift_reg	Shift_Register	Memory Modules User Guide	No	Yes	Yes
pmi_distributed_dpram	Distributed_DPRAM	Memory Modules User Guide	No	Yes	Yes
pmi_distributed_rom	Distributed_ROM	Memory Modules User Guide	No	Yes	Yes
pmi_distributed_spram	Distributed_SPRAM	Memory Modules User Guide	No	Yes	Yes
pmi_ram_dp	RAM_DP	Memory Modules User Guide	Yes	Yes	Yes
pmi_ram_dp_be	RAM_DP	Memory Modules User Guide	Yes	Yes	Yes
pmi_ram_dp_true	RAM_DP_True	Memory Modules User Guide	No	Yes	Yes
pmi_ram_dq	RAM_DQ	Memory Modules User Guide	Yes	Yes	Yes
pmi_ram_dq_be	RAM_DQ	Memory Modules User Guide	Yes	Yes	Yes
pmi_rom	ROM	Memory Modules User Guide	Yes	Yes	Yes

Creating IP Catalog Components

IP Catalog is an easy way to use a collection of functional blocks from the Radiant software. There are two types of components available through IP Catalog: modules and IP. IP Catalog enables you to extensively customize these components. They can be created as part of a specific project or as a library for multiple projects.

Below are the basic steps of using IP Catalog IP. For details of performing these steps, see the following topics.

1. Open IP Catalog. IP Catalog is accessed via a tab at the lower left of the Radiant software. Click the tab, or click in the tool bar, to view the list of available components.



2. Customize the component. These IP can be extensively customized for your design. The options range from setting the width of a data bus to selecting features in a communications protocol.
3. Generate the component and bring its .ipx file into your project. After generating the component, select the option “Insert to project.” This will automatically bring the .ipx file into your project after the generation step completes. If you do not select this option, add the .ipx file to your project as you would with any other source file (such as a Verilog or VHDL file) after the generation is complete.
4. Instantiate the component into the project’s design. An HDL instance template is generated during the generation step to simplify this step.
5. IP Catalog IP can be further modified or updated later. After the .ipx file has been added to the Radiant software project, it is visible in the project’s file list. Double-clicking the .ipx file brings up the component’s configuration dialog box where changes can be made and the generation process repeated.

Downloading IP from IP on Server

You can use the IP on Server tab of IP Catalog to download and install the latest IP available from Lattice Semiconductor. Before you can download any IP, you need to set up an Internet connection. If you haven’t already, choose **Tools > Options > General > Network Settings** and fill in the dialog box.

The website also has links to other vendors of IP. To see all that’s available and to learn about licensing and other vendors of IP, go to the Lattice website: www.latticesemi.com/ip.

To download Lattice IP on Server:

1. In IP Catalog, click the **IP on Server** tab, located at the upper-left of the tool. The software connects to the Lattice Semiconductor website.
2. Click the refresh icon to update the list.
3. Expand the folder tree and select the IP you want to download.

   Information about the IP appears in the right pane including links for additional information.

   IP that are compatible with your selected device have icons highlighted in dark blue. IP that are not compatible with your version of Radiant software are have blue icons with a yellow triangle. Look for device support information in the right pane.
4. To download the IP, click the IP and choose to the right of the IP name in the list.
5. The downloaded IP is now added to the IP list in the **IP on Local** tab.

Note

If Catalog detects a new version of an IP on Server which has been installed in local, the IP icon in IP on Server will alert you with a red dot next to the IP icon.

Using IP Catalog Search Features

IP Catalog has search features that allow you to search the list of IP and modules by keywords, and also search by by type.

To use the IP Catalog search feature:

1. IP Catalog, click the icon in the upper right.
2. Search by bus type by clicking one or more of the ABH-Lite, APB, or AS14-Stream boxes. The list of modules and IP will display modules that support these bus types.
3. Search by keyword to display module names or IP that contain the keyword in the name. For example, typing “adder” will display all modules that have adder in the module name.

Regenerating IPs

The regenerate function is an easy way to update your IP without needing to invoke the IP Catalog tool.

If your design was created in an older version of Radiant software, or if your design was created for a different Lattice device, your IP may need to be regenerated. Regenerating IP will update the IP to the current version.

You can regenerate one or or regenerate all IPs to update device information or new new version to existing .ipx file.

To regenerate as single IP:

1. In the Radiant software file list, right-click the IP file you wish to regenerate.
2. In the dropdown menu, choose **Regenerate**.

To regenerate all IP in your design:

1. In the Radiant software file list, right-click anywhere in the File List.
2. In the dropdown menu, choose **Regenerate All**. A dialog box will pop up to allow you to select one or more IP to regenerate.

Generating a Component with IP Catalog

Use IP Catalog to generate a customized functional block from any component or installed IP.

To generate a component:

1. In the Module/IP tree, select the component that you want to generate.
2. To get more information about the component, in either the IP on Local tab, or IP on Server tab, click on the module or IP, and then click on the blue question mark **.** The IP Information page will appear on the right.
3. Double-click the Module/IP or click the Generate Module/IP Instance button, located at the upper-left of the tool the In the Module/IP Block Wizard. Specify general project information and the base file name for the component.
   - Component Name is the base name for the component's files (that is, with no extension).
   - Create In is the location for the customized component's files.
4. Click **Next.**

   The component's dialog box opens.
5. In the dialog box, select your desired options.When generating a PLL, click **Calculate** to calculate divider settings and actual output frequencies based on CLKI and desired frequencies.
6. Click **Generate**.
7. To automatically import the .ipx file into your project when the component is generated, select **Insert to project** in the Check Generating Result dialog box.
8. Click **Finish**.

IP Catalog Output Files for Components

IP Catalog creates the following output files for components under the specified Project Path. The <*file_name*> comes from the File Name specified in the Module/IP Block Wizard.

IP Catalog creates some different files for IP. These are documented in the IP's associated user guide.

Table 2: Output Files

File Name	Description
<*file_name*>.ipx	Manifest file. This file is loaded into the design project so that modifications can be made to the component.
<*file_name*>_tmpl.v	Instantiation template for Verilog netlist.
file_name>_tmpl.vhd	Instantiation template for VHDL netlist
<*file_name*>.v	Verilog HDL file for both synthesis and simulation. Verilog output files declare implicit wire types.
tb_top.v	Testbench for associated component.

Importing an IP Catalog Component into a Project

After generating component source files using IP Catalog, you can import the component by importing the IP Catalog manifest file (.ipx). Components have several files of different types, but you only need to import the .ipx. The .ipx file identifies the components needed to make up the component.

Importing the files may not be necessary if the "Import IPX to project" option was selected when the component was generated.

To import a component:

1. In the File List view, right-click the implementation folder ( ) and choose **Add > Existing File**.
2. Browse for the customized component's .ipx file, <*file_name*>**.ipx**, and select it.
3. Click **Add**.

   The .ipx file is added to the File List view.
4. Check the Output Panel for error messages. If the component is not targeted for the current device, try double-clicking the file to regenerate the component.

After importing the component, you can further customize it for your design project by changing options specific to the component. You can also update older components or IP to the latest version.

Instantiating an IP Catalog Component

IP Catalog components are instantiated the same way other components are in your HDL. When you generate components in IP Catalog, the tool also generates Verilog and VHDL reference templates. You can instantiate the reference templates for implementation or simulation flow of any design, as needed.

The instantiation file is located in the folder in which the component was created. The file name is based on the component's name: <*component_name*>_tmpl.v or <*component_name*>_tmpl.vhd.

You can instantiate the IP Catalog component in one of the following ways:

- If you selected the option of not inserting the generated IP into your project, browse to the <*component_name*>_tmpl.v or <*component_name*>_tmpl.vhd, open the file in a text editor, copy the text, and paste into your source file. Then, add an instance and signal names to the component ports.
- If you selected the option of inserting the generated IP into your project, the IP component configuration file (.ipx) appears in your design implementation.

  In the File List, right-click the .ipx file, and choose the instantiation command:

  - For Verilog, choose **Copy Verilog Instantiation**, and paste it into your source file. Then, add an instance and signal names to the component ports.
  - For VHDL, choose **Copy VHDL Component**, and paste it into your source file; then choose **Copy VHDL Instantiation**, and paste it into your source file. Then, add an instance and signal names to the component ports.

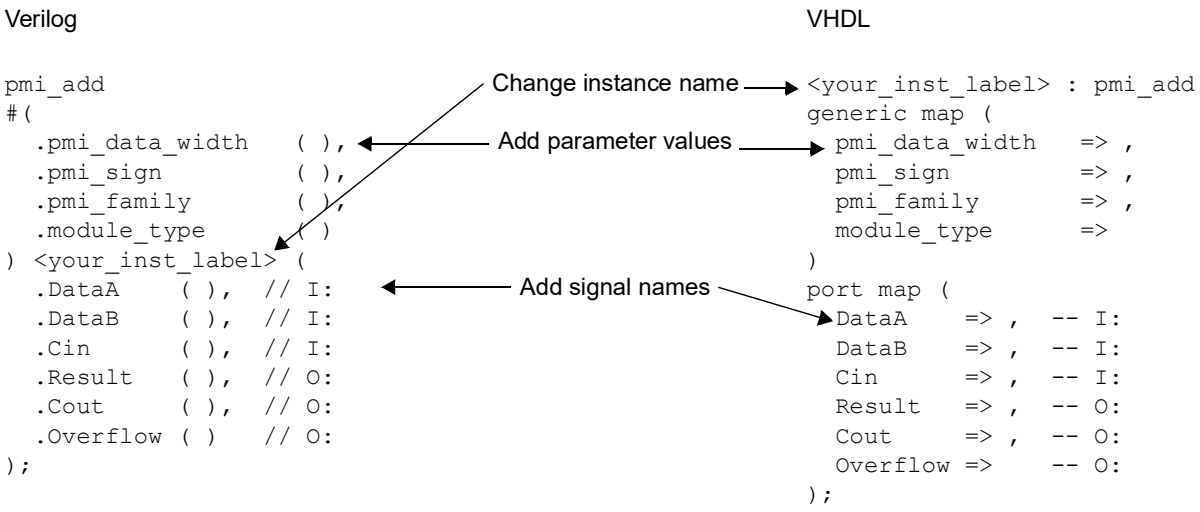
Instantiating a PMI Component

PMI components are instantiated the same way other components are in your HDL. The Radiant software provides a template for the Verilog or VHDL instantiation command that specifies the customized component's ports and parameters. Customize the component by changing its parameters.

To instantiate a PMI component:

1. Click **Tools > Source Template > Verilog/VHDL > PMI Templates**.
2. With Source Editor, open the source file that will receive the component.
3. Drag and drop the text into your source file.
4. Add an instance name, set parameter values, and add signal names to the corresponding component ports in the instantiation command. Refer to "PMI Instantiation Command Example" on page 13.

Figure 1: PMI Instantiation Command Example



5. If using stand-alone synthesis and simulation tools, add the PMI soft IP from the <*install_path*>/ip/pmi/ directory.
6. Save and close the source file.

Encrypting IPs

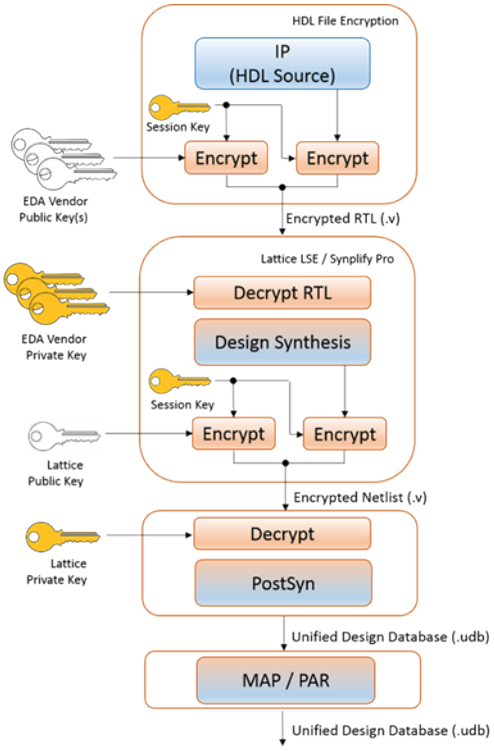
The Radiant software provides the IP security based on the Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP) of the IEEE 1735-2014 version 1 standard.

The implementation enables securing IP while ensuring the design flow and the interoperability across different EDA tools. Upon entering the design into the Radiant software, you select the desired level of security by encrypting the HDL source files. The set security level propagates through the design and impacts the visibility of secured objects in various Radiant tools. The current Radiant software supports encryption of VHDL and Verilog HDL files. The encrypted IP can be processed by other third-party EDA tools including Aldec Active-HDL, Cadence NCSim, Mentor Graphics ModelSim, Synopsys Synplify Pro, and Synopsys VCS simulator.

IP Encryption Flow

IP encryption flow enables you to protect your IP design. Following the industry standard, the Radiant software, through the IP encryption flow, allows the partnership between the IP Vendor, supported EDA vendor, and Lattice.

Figure 2: IP Encryption Flow



The encryption flow uses symmetric and asymmetric encryption methods to maximize the design security. The symmetric method only involves a single symmetric key for both, encryption and decryption. The asymmetric method involves the public-private key pair. The public key is published by a vendor and is used by the Radiant software. The private key is never revealed to the public.

The Radiant software supports these cryptographic algorithms:

- AES-128/AES-256: symmetric algorithm used to encrypt the content of the HDL source file.
- RSA-2048: asymmetric algorithm used to obfuscate a key used in HDL file encryption. The RSA is defined by the public-private key pair. You must be familiar with both keys in order to perform RSA decryption.

HDL File Encryption Flow

The overall HDL file encryption flow is summarized in these steps:

- The source file of the IP design is AES encrypted using a symmetric session key. The AES encryption uses the CBC-128 or CBC-256 algorithm. In the source files, this section is referred to as a data block.
- The session key is RSA encrypted using the vendor's public key. In the source files, this section is referred to as a key block. Multiple key blocks may be present in the source file.
- The encrypted session key and the encrypted design are merged to file generally named the Encrypted RTL

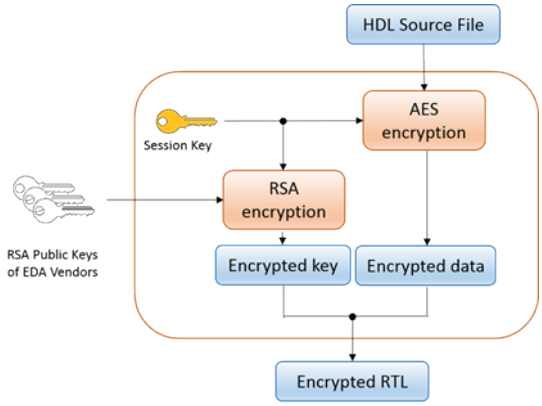
Each encrypted source file contains a single data block and one or more key blocks. The number of key blocks depends on the number of provided public keys.

NOTE

To decrypt an encrypted source file, you must perform the IP encryption flow steps in the reverse order.

During the next step in the design flow, typically synthesis, the Encrypted RTL is decrypted to access the original IP design, as shown in the following figure.

Figure 3: HDL Encryption Flow



By separating the encryption of data and key, you can use public keys from different vendors to encrypt the same HDL file.

For specific steps and information on how to encrypt HDL files in the Radiant software, refer to the following section in the Radiant software online help: **User Guides > Securing the Design**.

Packaging IP Using IP Packager

IP Packager is a tool for you to create an IP package easily. You can edit ports, files, parameters, and memory in the IP Packager, and pack a customized IP directly.

See Also

- “Running IP Packager and Viewing Documentation” on page 16
- “Installing IP Created with IP Packager into IP Catalog” on page 16

Running IP Packager and Viewing Documentation

IP Packager is a stand-alone tool. IP Packager can be run from both Windows and Linux. IP Packager documentation is contained within the IP Packager tool.

To run IP Packager:

- In Windows, go to the Windows Start menu and choose **Programs > Lattice Radiant Software > Accessories > IP Packager**.
- In Linux: from the ./<Radiant Software Install Path>/bin/lin64 directory, enter the following on a command line:

```
./ippack.sh
```

The IP Packager tool opens.

To view IP Packager documentation:

- In the IP Packager main window, choose **Help > Help**.

The IP Packager can also be run from the command line. Refer to the Command Line Reference Guide, located in both the Radiant Help, and in the *Radiant Software User Guide*.

See Also

- “Packaging IP Using IP Packager” on page 15
- “Installing IP Created with IP Packager into IP Catalog” on page 16

Installing IP Created with IP Packager into IP Catalog

You can download and add IP created with IP Packager into IP Catalog.

To download and add a User IP:

1. In the Radiant IP Catalog, click on the **Install a User IP** button.
2. In the **Select user IP package file to install** dialog box, browse to the Radiant Software IP Package (.ipk) file, and click **Open**.

- The Soft IP will be installed into a folder in the user’s personal directory. For example: C:/Users/<*username*>/RadiantIPLocal/<*IP_name*>.
- The Soft IP will be added into the IP Catalog.

See Also

- “Packaging IP Using IP Packager” on page 15
- “Running IP Packager and Viewing Documentation” on page 16

Revision History

The following table gives the revision history for this document.

Date	Version	Description
11/9/2022	2022.1	▶ Added LAV-AT-E content.
03/29/2022	3.1.1 Update	▶ Added MachXO5 content.
12/13/2021	3.1	▶ Added Parameterized Module Instantiation table to “Designing with IP and PMI” section. ▶ Changed “Packaging IP Using Radiant IP Packager” section to “Packaging IP Using IP Packager,” to document new IP Packager tool. ▶ Removed Appendix A, “Metadata (.xml) File Structure.”
05/22/2020	2.1	Updated screen capture to match Radiant 2.1 software.
11/05/2019	2.0	Updates for Radiant 2.0 software.
04/23/2019	1.0	Initial Release.