Licensing User Guide for Linux

November 22, 2023
Copyright
Copyright © 2023 Lattice Semiconductor Corporation. All rights reserved. This document may not, in whole or part, be reproduced, modified, distributed, or publicly displayed without prior written consent from Lattice Semiconductor Corporation (“Lattice”).

Trademarks
All Lattice trademarks are as listed at www.latticesemi.com/legal. Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. Modelsim and Questa are trademarks or registered trademarks of Siemens Industry Software Inc. or its subsidiaries in the United States or other countries. All other trademarks are the property of their respective owners.

Disclaimers
NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL LATTICE OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF LATTICE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Lattice may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. Lattice makes no commitment to update this documentation. Lattice reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. Lattice recommends its customers obtain the latest version of the relevant information to establish that the information being relied upon is current and before ordering any products.
## Type Conventions Used in This Document

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

Chapter 1  Lattice Development Tools  6
  Device Support and Licensing  6
  Lattice Radiant Software  6
  Lattice Diamond Software  7
  ispLEVER Classic Software  7
  Lattice Propel Design Environment  8
  iCEcube2 Design Software  9

Chapter 2  Licensing Overview  10
  Introduction to Lattice Design Tools Licensing  10
  How to use this Guide  12

Chapter 3  Licensing Basics  13
  License Types  13
    Node-locked License  13
    Floating License  15
    FEATURE vs. INCREMENT in a Floating License  16
    VERSION vs. EXPIRATION DATE in a License  17
  How to Obtain a License  17
    Free Licenses  17
    Subscription Licenses  18
    How to determine the MAC address/NIC ID  19
    How to Fill in the Licensing Forms  19

Chapter 4  License Installation  22
  Node-locked License Installation  22
  Floating License Installation  23
    Server  23
    Client  25
Chapter 5  IP (Intellectual Property) Licensing  27
  Introduction to IP Cores  27
  Introduction to IP Licensing  34
    Types of IP Licenses  35
    Free IP Licenses  35
    Evaluation Mode in IP  36

Chapter 6  Frequently Asked Questions  37

Revision History  44
Lattice Development Tools

Device Support and Licensing

Lattice design tools are built to help you keep innovating. Whether you are designing high-volume mobile handsets or leading-edge telecom infrastructure, our easy-to-use tools will help you bring your ideas to market faster - ahead of your competition.

Lattice Radiant Software

Full-featured FPGA design suite offering best-in-class tools for small form factor FPGA applications. Powerful yet intuitive tools provide fast design starts and precise implementation with intelligent planning and accurate analysis.

Supported FPGA

- Avant-E
- MachXO5-NX
- CertusPro-NX
- Certus-NX
- CrossLink-NX
- CertusPro-NX-RT
- Certus-NX-RT
- iCE40 UltraPlus
Lattice Diamond Software
Leading edge design software for Lattice FPGA families. Upgrade your design process with an easy-to-use interface, superior design exploration, optimized design flow, Tcl scripting, and more.

Supported FPGA
- ECP5UM
- ECP5UM5G
- LatticeECP3
- LatticeECP2M/S
- LatticeECP2/S
- LatticeSC
- LatticeSCM
- Crosslink
- CrosslinkPlus
- ECP5U
- MachXO3D
- LatticeECP2
- Mach-NX
- MachXO2
- LatticeXP2

ispLEVER Classic Software
ispLEVER Classic is the design environment for Lattice CPLDs and mature programmable products. It can be used to take a Lattice device design completely through the design process, from concept to device JEDEC or Bitstream programming file output.
Supported FPGA

- ispMACH 4000
- ispMACH 4A3
- ispMACH 4A5
- ispMach 5000VG
- ispGDX
- ispGDX2
- ispLSI 1K
- ispLSI 2K
- ispLSI 5000VG
- ispXPGA-E

Lattice Propel Design Environment

Lattice Propel is a complete set of graphical and command-line tools to create, analyze, compile, and debug both FPGA-based processor system hardware and software design.
iCEcube2 Design Software

Easy-to-use design tools to help you hit your cost, power, and time-to-market targets. iCEcube2 design software supports the iCE40 family of ultra-low-density FPGAs.

Supported FPGA

- iCE40 UltraPlus
- iCE40 LP/HX/LM
- iCE40 Ultra/UltraLite
Chapter 2

Licensing Overview

Introduction to Lattice Design Tools Licensing

Lattice Design Tools require a license to utilize the software. This comes into two categories: Free and Subscription Licenses.

**Free Tools licenses** permit access to certain devices with the full bitstream. With Radiant, you can still generate bitstream with a free license using Evaluation Mode for certain devices.

### Table 1: Device Support Per Design Flow

<table>
<thead>
<tr>
<th>Device Support</th>
<th>Radiant Free License</th>
<th>Diamond Free License</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Synthesize</td>
<td>Map</td>
</tr>
<tr>
<td>Avant-AT-E</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Avant-AT-G</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Avant-AT-X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CrossLinkU-NX</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MachXO5-NX</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CertusPro-NX</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Certus-NX</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CrossLink-NX</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CertusPro-NX-RT</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Certus-NX-RT</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>iCE40 UltraPlus</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Support</th>
<th>Synthesize</th>
<th>Map</th>
<th>Place &amp; Route</th>
<th>Bitstream</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECP5UM, ECP5UM5G</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Not Available*</td>
</tr>
<tr>
<td>LatticeECP3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Not Available*</td>
</tr>
</tbody>
</table>
### Table 1: Device Support Per Design Flow (Continued)

<table>
<thead>
<tr>
<th>Device Support</th>
<th>Synthesize</th>
<th>Map</th>
<th>Place &amp; Route</th>
<th>Bitstream</th>
</tr>
</thead>
<tbody>
<tr>
<td>LatticeECP2M/S, LatticeECP2/S</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Not Available*</td>
</tr>
<tr>
<td>LatticeSC, LatticeSCM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Not Available*</td>
</tr>
<tr>
<td>Crosslink, CrosslinkPlus</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ECP5U</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LatticeECP2, LatticeEC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MachXO3D, MachXO3L/LF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MachXO2, MachXO</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LatticeXP2, LatticeXP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Platform Manager 2, Platform Manager</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mach-NX</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Support</th>
<th>Diamond Free License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesize</td>
<td>Map</td>
</tr>
<tr>
<td>ispMACH 4000, ispMACH 4A3, ispMACH 4A5, ispMach 5000VG</td>
<td>✓</td>
</tr>
<tr>
<td>ispGDX, ispGDX2</td>
<td>✓</td>
</tr>
<tr>
<td>ispLSI 1K, ispLSI 2K, ispLSI 5000VG</td>
<td>✓</td>
</tr>
<tr>
<td>ispXPGA-E</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Note:**

*You need a subscription license to enable full bitstream capability.

**Enables a 4-hour hardware timer. You need a subscription license to remove this and have full access to the bitstream.

Please visit the Lattice online store.
How to use this Guide
This installation guide is authored for Client Machines and License Administrators.

Floating License Installation
- For License Administrators who set up company servers and client licenses, please refer to the Server section.
- For Clients who access the server license, please refer to the Client section to set up the local client license.
Chapter 3

Licensing Basics

License Types

There are two license categories for Lattice Tools: **Node-locked and Floating**.

**Node-locked License**

A node-locked license is confined to use on one specific machine only. The license is uncounted, which means that if software is operating on a particular machine, there is no limit on the number of instances permitted to run.

**Node-locked License: 4 Machines = 4 Licenses**

Software *limited* to single machine
LICENSING BASICS  :  License Types

Free Node-locked License

Subscription Node-locked License
Floating License

A floating license allows multiple clients to check out individual features concurrently from a shared license server. Floating licenses require the correct license server information and daemon location. Floating licenses are limited to the number of features seats specified in the license.

**Floating License: No of seats — No of machines with access**

Software *available* to entire network

Free Floating License
Subscription Floating License

**FEATURE vs. INCREMENT in a Floating License**

**INCREMENT**
Increment are additive lines in which a series of lines result in the sum of all the seats.

Increment lines: 1+4 = Total of 5 licenses

```
INCREMENT latticeminin mgclld 2023.09 25-mar-2024 13F9655C51783315ADF7B
VENDOR STRING=LIC_RADIANTR user_info=Trisha.Radiantlatticeminin.com TI_OK
INCREMENT latticeminin mgclld 2023.09 25-mar-2024 EF9655C51783315ADF7B
VENDOR STRING=LIC_RADIANTR user_info=Trisha.Radiantlatticeminin.com TI_OK
```
FEATURE
Feature lines are not additive in which only the first feature line will be taken into consideration in a license file.

Feature lines: Move the 4-seat license FEATURE line to the top of the license if 4 concurrent users are needed

```
FEATURE LSC_RADIANT lattice 2024.03 25-mar-2024 1 191937CDF4B1 \
  VENDOR_STRING=LSC_RADIANT

FEATURE LSC_RADIANT lattice 2024.03 25-mar-2024 4 191937CDF4B1 \
  VENDOR_STRING=LSC_RADIANT
```

VERSION vs. EXPIRATION DATE in a License

VERSION
The Maintenance Version covers the latest valid tool version. It needs to be greater than the software version to run the tools.

```
INCREMENT latticemsim mgcl9 2023.09 25-mar-2024 4 3f96f5c91783315ADF7B \
  VENDOR_STRING=7D06295E ISSUE="ModelSIM Lattice" SN=285454249 \
```

EXPIRATION DATE
The License Expiration Date specifies the last day that the license is valid.

After March 25, 2024, this FEATURE line is no longer valid

```
FEATURE LSC_RADIANT_SUBSCRIPTION lattice 2024.03 25-mar-2024 1 \
  B0F016E4F29A VENDOR_STRING=LSC_RADIANT_SUBSCRIPTION
```
How to Obtain a License

Free Licenses
Lattice offers free licenses which enable you to design and evaluate the performance of the supported devices per Software tool. To request free software licenses, see the links below. For other Software Tools, please go to the Software Licensing page.

Radiant Free License
The free license enables full design and implementation functionality for Radiant-supported devices.
- Request Node-locked License
- Request Floating License

Diamond Free License
The free license enables you to design and evaluate the performance of non-SERDES-based Diamond-supported devices.
- Request Node-locked License
- Request Floating License

Subscription Licenses
Subscription licenses enable you to design and optimize solutions for supported devices in each Software tool. To purchase or renew a Software license, please go to the Online Store or contact a local sales representative or distributor.

If you have purchased a Software license and received a Software Serial Number, please go to our Subscription licensing form to generate the required license.

Subscription License 30-Day Extension
If you need a temporary license extension while license renewal is being processed, we have a solution for this scenario.

This license is only for a temporary extension to a subscription license that will expire.
To request an extension, you will need the following:

- Your subscription license serial number
- Click **Request an extension** and go to **My Licenses** to edit the record and click **Extend 30 days**.
- If you have not previously connected to our support portal, you will need to verify your contact information.

**How to determine the MAC address/NIC ID**

The Network Interface Card ID uniquely identifies your workstation on the network. There are two ways to determine your MAC Address:

1. In a terminal window, type `ifconfig -a` and press return.
   - In the list of interfaces, find `eth0`.
   - Locate the number next to `HWAddr`.

2. Obtain the host ID of your license server with the following command:
   ```bash
   % <install_path>/ispfpfga/bin/lin64/lmutil lmhostid
   ```

**How to Fill in the Licensing Forms**

**Free Web Licenses**

You may generate free web licenses for certain devices that are supported by Lattice Software Tools.

To fill in the **Software Licensing Request Form**

1. Ensure that you enter your NIC ID without any separator or spaces.
2. Proceed to check the tick box verifying that you are not an employee of Cadence Design Systems, Mentor Graphics Corporation, or Synopsys, Inc.
3. You may choose to include some Free IPs listed and click generate.
Free License Software Licensing Form

Subscription Licenses
If you have purchased a Software license and received a Software Serial Number, you will be directed to our Subscription licensing form where you will need:

- A latticesemi.com account.
  
  It is recommended to create an account if you do not already have one yet.
- The given serial number.

Note:
Click Check SN to check if the serial number is available to license.

- The NIC/Physical address of the computer you wish to license.
- Company Name

Subscription Licensing Page
Subscription Licensing Form

Evaluation Licenses
Lattice offers evaluation license to enable free subscription license for end-users up to 60 days. You may raise a ticket to the Submit Support Ticket page.

Academic License Program
We offer a 1-year Lattice Design tools license free for colleges and universities who meet our academic license program requirements. If you are a professor, and you are interested in applying, please login to your Lattice account and fill out the application form here. If you are a student, please coordinate with your professor to join this program.
Chapter 4

License Installation

To fully utilize the Lattice Software Design Tools, Software Licenses must be installed properly. The process depends on the type of license you are to use.

Node-locked License Installation

1. You will receive a license email once the license has been generated. Save the attached file (license.dat) to the installation folder of your software product (e.g. <sw_install>/license/license.dat).

2. Set up the license as follows:
   
   If your license.dat file is not under <install_path>/2022.1/license, you must set the LM_LICENSE_FILE variable to the location of your license file. For example:
   
   If you are using CSH, set the variable to:
   
   ```
   % setenv LM_LICENSE_FILE $LM_LICENSE_FILE (/<license_directory>/license.dat)
   ```
   
   If you are using BASH, set the variable to:
   
   ```
   % export LM_LICENSE_FILE=$LM_LICENSE_FILE:/<license_directory>/license.dat
   ```

3. Run the Radiant software executable file in the command line as follows:

   ```
   % <install_path>/bin/lin64/radiant &
   ```
   
   With the Radiant software script, you can also run the following tools in stand-alone mode.
   
   To invoke stand-alone Reveal Analyzer, run:

   ```
   % revealrva
   ```
Floating License Installation

Server

To enable a floating license, you must have a license server set up on a Windows or Linux server to monitor your Radiant software license. Each client PC must have the LM_LICENSE_FILE variable set to point to the license file on the server.

To edit the License File:

After obtaining a floating license from Lattice Semiconductor, you must edit the license file to specify the server’s name and the paths to the Lattice daemon.

1. Edit the SERVER line by replacing the nodename with the host name and the port ID (1710).

   The port ID (1710) in this example must be assigned a TCP/IP port number that is not already in use on the server, so you might need to change it.

2. Edit the Lattice DAEMON line, replacing daemon_path with the path to Radiant software. For lmgrd V11, the path is:

   ```
   % <install_path>/ispfpga/bin/lin64
   ```

3. Edit the ModelSim DAEMON line, replacing daemon_path with the path to ModelSim Lattice Edition. For ModelSim, the path is:

   ```
   For Radiant: %<install_path>/radiant/2022.1/modeltech/linuxloem/mgcld
   For Diamond: % <install_path>/diamond/3.12/modeltech/win32loem/mgcld
   ```

To start the License Manager:

- Type the following command on one line to start the license manager daemon:

  ```
  % <install_path>/ispfpga/bin/lin64/lmgrd -l <install_path>/license/license.log -c <install_path>/license/license.dat
  ```

  Redirecting output to a log file is helpful when you debug licensing problems. The -l switch tells the license manager to send its output to a log file (license.log), and -c tells it which license to serve.

  The log file contains information on the status of the server and the daemon and TCP port in use. It also shows which users have checked out the license and the checkout time.

To stop the License Manager:

If it is necessary to stop the FLEXlm license manager, you may follow this procedure:
1. Confirm that the daemon is running by typing the following command:
   % ps -ef | grep lmgrd

2. If lmgrd.exe is running, type the following command on one line to stop the daemon:
   
   For 64-bit systems:
   % <install_path>/ispfpga/bin/lin64/lmutil lmdown -c <install_path>/license/license.dat

   The following prompt appears:
   Shutting down FLEXlm on nodes: <hostname>
   Are you sure? [y/n]:

3. Type Y and press Enter to shut down the license daemon.

To install and run the License Manager on a Remote Server:

You can install and run the License Manager from a location other than the default directory.

1. Copy the files lattice, lmgrd, and lmutil from the following directory:
   % <install_path>/ispfpga/bin/lin64/

2. You can also install it to your desired location. For example:
   % <remote_server_install_path>/my_machine/lattice_license/

To start the License Manager from a Remote Server:

Type the following command on one line to start the license manager daemon:

   % <remote_server_install_path>/lmgrd
   -l <install_path>/license/license.log
   -c <install_path>/license/license.dat

To stop the License Manager on a Remote Server:

If it is necessary to stop the FLEXlm license manager running on a remote server, you may follow this procedure:

1. Confirm that the daemon is running by typing the following command:
   % ps -ef | grep lmgrd

2. If lmgrd.exe is running, type the following command on one line to stop the daemon:
   % <remote_server_install_path>/lmutil lmdown -c <install_path>/license/license.dat

   The following prompt appears:
   Shutting down FLEXlm on nodes: <hostname>
   Are you sure? [y/n]:

3. Type Y and press Enter to shut down the license daemon.
Client

In this configuration, the software tool is installed on your license server (for license manager utilities and daemons) and on each client that uses the software tool. This configuration gives the best run-time performance. After you receive your floating license and ensure that the license manager is running, install the software locally on each client that will use the floating license.

To gain access to the licenses on the remote license server, you need to set the environment variable LM_LICENSE_FILE value to:
license_port_number@linux/ubuntu_host_name.

If you are using CSH, set the environment variable to:

% setenv LM_LICENSE_FILE $LM_LICENSE_FILE (/<license_port_number@linux/ubuntu_host_name )

If you are using BASH, set the environment variable to:

% export LM_LICENSE_FILE=$LM_LICENSE_FILE:/<license_port_number@linux/ubuntu_host_name

Installing System Library Packages to Set up Floating License Server

If the dynamic linker/loader ld-lsb-x86-64 cannot be found, it is an indication that LSB packages are missing.

Installing System Library Package on Red Hat 64-bit Operating System Manually

If you want to manually install the system library package, use the following command:

sudo yum install redhat-lsb

Installing System Library package on Ubuntu 64-bit Operating System Manually

If you want to manually install the system library package, use the following command:

sudo apt-get install lsb-core

Checking the License File with Lmutil

Use the Lmutil tool to troubleshoot the status of your license file. The Lmutil tool is located at:

<install_path>/ispfpga/bin/lin64
The `lmstat` command determines the features of your license file.

- The `-a` argument displays all information.
- The `-c` argument uses the specified license files.

The following example shows the usage of the `lmutil lmstat` command to check the license file status:

```
% ./lmutil lmstat -a -c <license_port>@<license_server>
```

The license status is returned, including feature lines, number of licenses issued, and licenses in use.

---

**Running Radiant Software from the Command Line**

There are two ways to run the Radiant software from the command line:

- Through Radiant Tcl Console.
- By running executable files directly.

---

**Running Stand-Alone Radiant Tcl Console**

Radiant software development environment includes Tcl Console, which allows you to run scripts for automating common tasks. Tcl Console is also available outside of the user interface in order to run custom scripts.

To launch the stand-alone Tcl Console, enter the following to the command line:

```
% <install_path>/bin/lin64/radiant
```

These commands configure the environment, allowing all of the underlying design tools to be run. Refer to the Radiant Help for more information about the command line.

---

**Running Commands Using CSH or BASH Interpreters**

- You must run the following commands if you are using BASH:

  ```bash
  % export bindir=<install_path>/bin/lin64
  % source $bindir/radiant_env
  ```

  Next, you can run the executable files directly. For example, you can invoke the Radiant software GUI by using:

  ```bash
  % pnmain &
  ```

  Or, you can run Power Calculator by running:

  ```bash
  % pwcmain &
  ```
Chapter 5

IP (Intellectual Property) Licensing

Introduction to IP Cores

Lattice IP Cores are pre-tested, reusable functions that allow you to focus on their unique system architectures. These IP cores provide industry-standard functions such as PCI Express, DDR, Ethernet, CPRI, and embedded microprocessors. In addition, several independent IP providers have teamed with Lattice to offer additional high quality, reusable IP cores. For a complete listing of IP cores from Lattice and its 3rd party partners, please go to the Lattice IP page.

Table 1: Radiant Devices

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>Avant</th>
<th>CertusPro-NX</th>
<th>Certus-NX</th>
<th>CrossLink-NX</th>
<th>Mach-NX</th>
<th>MachXO5</th>
<th>Cross Link</th>
<th>CrossLink Plus</th>
<th>iCE40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications</td>
<td>10 Gb Ethernet Mac</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SGMII and Gb Ethernet PCS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Triple Speed 10/100/1G Ethernet MAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connectivity</td>
<td>10 Gb Ethernet PCS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPIO</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x1 Endpoint</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x2 Endpoint</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x4 Endpoint</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express Root Complex Lite x1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1: Radiant Devices

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>Avant</th>
<th>CertusPro-NX</th>
<th>Certus-NX</th>
<th>CrossLink-NX</th>
<th>Mach-NX</th>
<th>MachXO5</th>
<th>Cross Link</th>
<th>CrossLink Plus</th>
<th>iCE40</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gigabit SGMII Ethernet PHY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>PCI Express Root Complex Lite x4</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Digital Signal Processing</td>
<td>CORDIC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Divider</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>FFT Compiler</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>FIR Filter Generator</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Processor, Controller &amp; Peripheral</td>
<td>DDR3 SDRAM Controller</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>DDR3 SDRAM PHY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>I2C Master</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>I2C Slave</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>I3C Master</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>I3C Slave</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>LPDDR2 SDRAM Controller Lite</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>LPDDR4 SDRAM Controller</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Multi-Port Arbiter for DDR3 Memory</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Controller</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Scatter Gather DMA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>SPI Master</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>SPI Slave</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>UART 16550</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Neural Network Accelerators</td>
<td>CNN Plus Acceleruator</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Compact CNN Acceleriator</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
### Lattice Propel

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>Avant</th>
<th>CertusPro-NX</th>
<th>Certus-NX</th>
<th>CrossLink-NX</th>
<th>Mach-NX</th>
<th>MachXO5</th>
<th>Cross Link</th>
<th>CrossLink Plus</th>
<th>iCE40</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHB Lite Interconnect Module</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AHB Lite to APB Bridge Module</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APB Interconnect Module</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2C Master</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RISC-V MC CPU IP</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RISC-V SM CPU IP</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGMII and Gb Ethernet PCS</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Memory Module</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART IP Core</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXI4 to AHB-Lite Bridge</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXI4 to APB Bridge</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXI4 Interconnect</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Video & Imaging

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>Avant</th>
<th>CertusPro-NX</th>
<th>Certus-NX</th>
<th>CrossLink-NX</th>
<th>Mach-NX</th>
<th>MachXO5</th>
<th>Cross Link</th>
<th>CrossLink Plus</th>
<th>iCE40</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Scaler</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:1 MPI CSI-2 Bridge</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Byte to Pixel Converter</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color Space Converter</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMOS to MIPI D-PHY Interface Bridge</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:2 and 1:1 MIPI CSI-2 to CSI-2 Camera Interface Bridge</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIPI CSI-2 Bridge</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSI-2/DSI D-PHY Receiver</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1: Radiant Devices

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>Avant</th>
<th>CertusPro-NX</th>
<th>Certus-NX</th>
<th>CrossLink-NX</th>
<th>Mach-NX</th>
<th>MachXO5</th>
<th>Cross Link</th>
<th>CrossLink Plus</th>
<th>iCE40</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSI-2/DSI D-PHY Transmitter</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deinterlacer</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSI to DSI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPD-LINK Receiver</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPD-LINK Transmitter</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gamma Corrector</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIPI D-PHY to CMOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
</tr>
<tr>
<td>MIPI DSI Bandwidth Reducer Display Interface Bridge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIPI DSI to OpenLDI/FPD-Link/LVDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pixel to Byte Converter</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLVS-EC Receiver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubLVDS Image Sensor Receiver</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubLVDS to MIPI CSI-2 Image Sensor Interface Bridge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video Frame Buffer</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color Correction Matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
</tr>
<tr>
<td>2D Image Noise Reduction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table 2: Diamond Devices

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>ECP5/ECP5-5G</th>
<th>ECP3</th>
<th>ECP2M</th>
<th>ECP2</th>
<th>MachXO2</th>
<th>MachXO3D</th>
<th>XP2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communications</strong></td>
<td>10 Gb Ethernet MAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.5 Gb Ethernet MAC</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.5 Gb Ethernet PCS</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPRI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPRI 5G</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPI4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SGMII and Gb Ethernet PCS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Triple Speed 10/100/1G Ethernet MAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>XAUI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connectivity</strong></td>
<td>JESD204A</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JESD204B</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JESD207</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x1 Endpoint</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x2 Endpoint</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x4 Endpoint</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express Root Complex Lite x1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express Root Complex Lite x4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x1 Endpoint - Optimized for ECP5UM5G</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI Express x2 Endpoint - Optimized for ECP5UM5G</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2: Diamond Devices

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>ECP5/ ECP5-5G</th>
<th>ECP3</th>
<th>ECP2M</th>
<th>ECP2</th>
<th>MachXO2</th>
<th>MachXO3D</th>
<th>XP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIPE</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI Master/ Target 33</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>PCI Target 33</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>PCI Target 66</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Serial RapidIO</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tri-Rate Serial Digital Interface (SDI) PHY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>Digital Signal Processing</strong></td>
<td><strong>Block Convolutional Encoder</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Block Viterbi Decoder</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cascaded Integrator-Comb (CIC) Filter</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CORDIC</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed Arithmetic (DA) FIR Filter</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divider</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Block Reed-Solomon Decoder</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFT Compiler</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIR Filter Generator</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interleaver/De-interleaver</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine Learning</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Median Filter</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerically-Controlled Oscillator (NCO)</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>IP Core</td>
<td>ECP5/ ECP5-5G</td>
<td>ECP3</td>
<td>ECP2M</td>
<td>ECP2</td>
<td>MachXO2</td>
<td>MachXO3D</td>
<td>XP2</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------</td>
<td>---------------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>-----</td>
</tr>
<tr>
<td>Processor, Controller &amp; Peripheral</td>
<td>Peak Cancellation Crest Factor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DDR SDRAM Controller Pipelined</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DDR2 SDRAM Controller Pipelined</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DDR3 SDRAM Controller</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DDR3 SDRAM PHY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LPDDR SDRAM Controller</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LPDDR2 SDRAM Controller Lite</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LPDDR3 SDRAM Controller</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Scatter Gather DMA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Neural Network Accelerators</td>
<td>CNN Accelerator</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lattice Propel</td>
<td>AHB Lite Interconnect Module</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AHB Lite to APB Bridge Module</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APB Interconnect Module</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EFB Module</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I2C_Monitor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QSPI_Master_Streamer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QSPI_Monitor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction to IP Licensing

Each Lattice IP core is a set of compiled code that implements a basic function, like a DDR2 memory controller, or a Tri-speed MAC, and is targeted/optimized for a specific Lattice FPGA family. You can first use the IP tools in Diamond or Radiant to configure some parameters in the IP core (for example, with DDR2, you might configure the data word width to a particular length).

The IP tool generates a compiled netlist, which is sometimes called a black box, because while you can connect the inputs and outputs, you cannot see inside this compiled design. Finally, using Diamond or Radiant, you can integrate the IP with your custom RTL to create your final design, which can then be programmed into a Lattice FPGA. This custom code is sometimes called a wrapper as it wraps around the IP core to create the final design.

### Table 2: Diamond Devices

<table>
<thead>
<tr>
<th>Function</th>
<th>IP Core</th>
<th>ECP5/ ECP5-5G</th>
<th>ECP3</th>
<th>ECP2M</th>
<th>ECP2</th>
<th>MachXO2</th>
<th>MachXO3D</th>
<th>XP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RISC-V MC CPU IP</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RISC-V SM CPU IP</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>System Memory Module</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UART IP Core</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Video &amp; Imaging</td>
<td>2D Edge Detector</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2D FIR Filter</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2D Scaler</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Color Space Converter</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deinterlacer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Display Interface Mux</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DVB-ASI</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gamma Corrector</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Median Filter</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video Frame Buffer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
After you have completed your design, you can generate a bitstream file and program a Lattice FPGA.

If you have a copy of Diamond or Radiant, you already have access to all the IP core code via the IPexpress or IP Catalog tool. When you purchase an IP core, you now have a license to fully utilize this code - i.e. you have purchased a key, which comes in the form of a file called license.dat.

To get this file, you need to complete a request using the IP Subscription License Form. After completing the required fields in the form, we will email you a new license.dat file, which works with Diamond or Radiant, and unlocks the IP core.

Types of IP Licenses

There are 2 ways you can purchase a license to use Lattice IP cores. The type of IP license you purchase is reflected by the part number you order; so, there is only one way to license any specific IP core part number. These license types include:

► Single Machine Annual License: Individual IP core that is limited to a single machine installation only. The use period is up to 1 year, so the license will expire at the end of the 12-month term. Upon expiration, you will still be able to use programming bitstream files generated over the previous year but will not be able to generate new bitstream files without the hardware timer. Part numbers for annual licenses will end in a suffix like -US. Extensions to this 1-year period are available.

► Site Perpetual License: This IP license is a floating license installed on a server. There is no expiration date to this license type, but a one-seat license is given for each item you purchase. Part numbers for site licenses will end in a suffix like –UT. These products are specific to a single type of IP and a single target FPGA technology.

For more information and support in IP licensing, please visit the Lattice IP Support page.

Free IP Licenses

Lattice offers free IP that requires licenses which can be generated through the Licensing page.
Evaluation Mode in IP

What are the limitations of using IP in Evaluation Mode?

You can do nearly all your design and prototyping work without an IP license. If you do not have a license to use the IP core, the software will insert a hardware timer into the programming bitstream. This will cause the bitstream to time-out after a set period - usually from 30 minutes to about 4 hours. After the period expires, you can re-program the FPGA to reset the period. Also, without an IP license, full timing simulation is not available.

To turn on Evaluation Mode:
1. In the Project Tab, click **Active Strategy > Bitstream Settings**.

2. Check the **IP Evaluation** tick box.

3. Click **Apply** and **OK** to exit.
1. **How do I manage my subscription licenses?**

   - Log in to our website and go to Support > Licensing.
   - Under Purchase/Renew Radiant License, click Subscription licensing.
   - Select My Licenses to view all licenses generated for your email address.

2. **Why do I get the following warning: "WARNING - Device1 iM4A3-64/32 is a mature device, please contact Lattice for the license to enable this device" when loading a program for my CPLD?**

   Please keep in mind that when programming mature devices, you must use the Diamond Programmer stand-alone version. The Diamond Integrated Programmer (Programmer integrated into the Lattice Diamond software) lists the mature devices in the Device Selection and scans them, but it cannot program mature devices, even with the Mature Device license. Bypass is the only available operation.

   - To program the mature device, visit our licensing page and request a Diamond-free license.

3. **Why does my license fail when using a remote desktop?**

   The error shown below occurs when a Node-locked license is used for remote connections. The resolution to this is having TS_OK on the license file.
4. How to improve Diamond/Radiant/Propel software startup speed caused by invalid license paths?

► In most cases, license paths can be edited using the LM_LICENSE_FILE in Windows Environment Variables (Windows Search Box > Edit Environment Variables).

► However, there are instances during software startup wherein the tool may search for previously used paths that no longer exist (for example, Network License Path). To verify this, go to Help > License Debug and press start. There should be no licensing errors as shown below.
If issues occur because of invalid paths not found in the LM_LICENSE_FILE environment variable, you can manually modify the registry item. You can exit Diamond or Radiant and type REGEDIT in the Windows Search Box.

Open the Registry Editor and go to Edit > Find and search for LATTICE_LICENSE_FILE. This may take a few minutes.

When the search results show the LATTICE_LICENSE_FILE registry entry, double-click it and delete any licensing locations that are no longer needed.
To see if the program startup has improved, press OK and open Lattice Diamond or Lattice Radiant.

5. How to determine the MAC Address for Debian OS?

You need to upgrade first to Debian testing for dependency on a newer library because Debian does not name the ethernet adaptor eth0.

To fix this using systemd:

```
apt install net-tools (for ifconfig -a)
create /etc/systemd/network/80-eth0.link
[Match]
MACAddress=08:62:66:4a:22:e5 (your ethernet adaptor MAC address)
[Link]
Name=eth0
then run update-initramfs -u
then reboot.
```

Note:
This workaround appears to have additional issues with script errors that may have dependencies on RHEL, as well as various requirements on 32-bit libraries before installing iCECube2.

6. How do I use the Serial Number I received from Lattice License Administrator to obtain my Software license?

Log in to our website and navigate to the Licensing Support page or Support > Licensing. Click Diamond Subscription License. When you are in the License Generation page, here are the options you can do.

Requesting a new license

- Select New License or Request a New License.
- Enter your Company Name and NIC (open an MS-DOS window and type "ipconfig /all" and press Enter. The MAC Address is a 12-digit hexadecimal value split into pairs with dashes, like this: 00-01-02-66-1D-E0. For Linux, type ifconfig -a).
- Enter the Serial Number and click check SN. Save the file assuming no errors appear.
- Your license should be delivered to your mailbox within a few minutes.
FREQUENTLY ASKED QUESTIONS

■ Renewal of expired license
  ■ Go to My Licenses.
  ■ Choose the expired license record you want to renew with your new Serial Number and press E (for edit).
  ■ Select Request Renewal, then enter the Company Name, First Name, Last Name, Email, Serial Number and proceed click.
  ■ Select SN and save assuming no errors appear.
  ■ Your license should be delivered to your mailbox within a few minutes.

■ License Extension
  ■ Go to My Licenses.
  ■ Choose the expired license record you want to renew with your new Serial Number and press E (for edit).
  ■ Scroll down to find the Extend 30 Days button and click Extend.
  ■ Your license should be delivered to your mailbox within a few minutes.

7. How can I switch or transfer my license to a different computer or PC?

To change or transfer a license to new computer or PC, submit a technical support case with the new computer’s MAC address.

The license will be generated shortly after License Admin acknowledges the request.

8. How can I configure a floating license in RHEL7 using only license management files?

1. Edit the server, daemon, and ModelSim daemon lines to license.dat file in /nas/storage/lattice.
   You can change the PORT numbers used that suit your needs, as shown below:

   SERVER lattice-lic-server 0052376A34FC 17700
   DAEMON lattice /nas/storage/lattice/bin/lin64/ PORT=50500
   Replace the daemon_path with the ModelSim Lattice Edition path. For ModelSim, the path is: %<install_path>/diamond/3.12/modeltech/win32loem/mgcl
   Please visit the link for the license daemons.

2. Download the Lattice RPM: diamond_3_12-base-240-2-x86_64-linux.rpm from: https://www.latticesemi.com/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond
   to: /tmp/lattice/ on the Linux server

3. List the contents of the rpm and look for the license binaries lmgrd|lattice
cd /tmp/lattice/

rpm -qlpv diamond_3_12-base-240-2-x86_64-linux.rpm | less

rpm -qlpv diamond_3_12-base-240-2-x86_64-linux.rpm | grep -i lattice

rpm -qlpv diamond_3_12-base-240-2-x86_64-linux.rpm | grep -i license

rpm -qlpv diamond_3_12-base-240-2-x86_64-linux.rpm | grep -i lmgrd

rpm -qlpv diamond_3_12-base-240-2-x86_64-linux.rpm | grep -i 'ispfpga'

drwxr-xr-x 2 root root 0 Dec 2 22:21 /usr/local/diamond/3.12/ispfpga

-rw-r-xr-x 1 root root 778858185 Dec 2 22:01 /usr/local/diamond/3.12/ispfpga/ispfpga.tar.gz

4. Extract the linux license binaries from the rpm file: ispfpga.tar.gz using the rpm2cpio and cpio and tar commands

```bash
cd /tmp/lattice

rpm -qlpv diamond_3_12-base-240-2-x86_64-linux.rpm | grep -i 'ispfpga'

drwxr-xr-x 2 root root 0 Dec 2 22:21 /usr/local/diamond/3.12/ispfpga

-rw-r-xr-x 1 root root 778858185 Dec 2 22:01 /usr/local/diamond/3.12/ispfpga/ispfpga.tar.gz

rpm2cpio diamond_3_12-base-240-2-x86_64-linux.rpm | cpio -idmv ...

./usr/local/diamond/3.12/examples/examples.tar.gz

./usr/local/diamond/3.12/ispfpga

-->>./usr/local/diamond/3.12/ispfpga/ispfpga.tar.gz <<------THIS IS THE TAR FILE YOU WILL NEED

./usr/local/diamond/3.12/license

./usr/local/diamond/3.12/license/license.txt

...
```

5. List the contents of the tarfile ispfpga.tar.gz looking for the lmgrd lmutil and lattice daemons

```bash
tar -tvf ./usr/local/diamond/3.12/ispfpga/ispfpga.tar.gz | grep -i '/lmutil|/lmgrd|/lattice'

-rw-r-xr-x relman/neosoft 1351888 2019-10-30 02:05 bin/lin64/lattice

-rw-r-xr-x relman/neosoft 1170840 2019-07-08 23:06 bin/lin64/lmgrd

-rw-r-xr-x relman/neosoft 1138392 2019-07-08 23:06 bin/lin64/lmutil
```

6. Extract the lmgrd lmutil and lattice daemons from the tarfile

```bash
tar -xvf ./usr/local/diamond/3.12/ispfpga/ispfpga.tar.gz bin/lin64/lattice

tar -xvf ./usr/local/diamond/3.12/ispfpga/ispfpga.tar.gz bin/lin64/lmgrd
```
FREQUENTLY ASKED QUESTIONS

7. On the Linux server, start the license service

    start server: /nas/storage/lattice/bin/lin64/lmgrd -c /nas/storage/lattice/license.dat -l /nas/storage/lattice/licence.log
    stop server: /nas/storage/lattice/bin/lin64/lmutil lmdown -c /nas/storage/lattice/license.dat
    query server: /nas/storage/lattice/bin/lin64/lmutil lmstat -c /nas/storage/lattice/license.dat
    check service: ps fuxwa | egrep -i 'lmgrd|lattice' | egrep -v grep
                   ps fuxwa | egrep -i 'lmgrd|lattice' | egrep -v grep

8. Query the lattice server showing ALL the licenses available and usage:

    /nas/storage/lattice/bin/lin64/lmutil lmstat -a -c /nas/storage/lattice/license.dat
Revision History

The following table gives the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/22/2023</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>