Release Notes for Lattice Diamond 3.8

Welcome to Lattice Diamond®, the complete design environment for Lattice Semiconductor FPGAs. Lattice Diamond design software offers leading-edge design and implementation tools optimized for cost-sensitive, low-power Lattice FPGA architectures.

Diamond is available for both the Windows and Linux operating systems. For details, see “System Requirements” on page 7.

This version of Diamond adds a variety of enhancements to make designing for Lattice Semiconductor programmable devices easier than ever. The design tools also include support for the latest Lattice Semiconductor devices. See “What’s New in Diamond 3.8” on page 2.

Lattice Semiconductor offers a rich variety of information sources, including the Help system, PDF manuals, tutorials, and online discussions. The easiest way to reach them all is through the online Help. The first topic in the Help provides links to all the other sources of information.

You can also find extensive information about Diamond and its capabilities, tools, and workflow on the Lattice Semiconductor website under:

www.latticesemi.com/latticediamond

What's New in Diamond 3.8

This release of Diamond provides a variety of new features.

MachXO2 Family Support MachXO2 640/256 QFN 48 package is generally available.

ECP5UM5G Family Support ECP5UM5G 381/554/756 CABGA and 285 CSFBGA package is generally available.

ECP5U Family Support ECP5U 12K device support has been added.

CrossLink (LIFMD) Family Support LIFMD 6000 CSFBGA/CTFBGA/WLCSP/UCFBGA package is generally available.

MachXO3L/LF Family Support Package 9400E CSFBGA, 9400C 256/400/484 CABGA package is generally available.

Soft Error Injection (SEI) Editor SEI Editor now supports MachXO2 and MachXO3L/LF devices is generally available.

Supported Devices

Lattice Diamond can be used with either a free license or a subscription license. The two licenses provide access to different device families.

Device Family	Free License	Subscription License
ASC	◀	◀
ECP5U	◀	◀
ECP5UM		◀
ECP5UM5G		◀
LatticeEC™	◀	◀
LatticeECP™	◀	◀
LatticeECP2™	◀	◀
LatticeECP2M™		◀
LatticeECP2S		◀
LatticeECP2MS		◀
LatticeECP3™		◀
LatticeSC™		◀
LatticeSCM™		◀
LatticeXP™	◀	◀

Device Family	Free License	Subscription License
LatticeXP2™	◀	◀
LIFMD (CrossLink)	◀	◀
MachXO™	◀	◀
MachXO2™	◀	◀
MachXO3L	◀	◀
MachXO3LF	◀	◀
Platform Manager™	◀	◀
Platform Manager 2	◀	◀

Updating Projects from an Earlier Version

If you want to work on a design project created with an earlier version of Diamond, start with the following procedures. These procedures adapt the project for the changes in Diamond.

Find out which version of Diamond your project was created with. Then work through the changes for that and every later version, starting with the earliest and going to the most recent. For example, if your project was created with Diamond 1.1, you would start with the changes for 1.1. After completing those changes, you would work on the changes for 1.2, then 1.4, and so on.

When you open a project from Diamond 1.2 or earlier, Diamond opens a dialog box warning that Diamond will automatically move all SDC files to the Synthesis Constraint Files folder in File List view and remove the “Input SDC Constraint File” options from the strategies. If the project is using LSE, the file names will be changed to use an .ldc extension.

Once saved, the project will not be compatible with earlier Diamond versions.

2.2 Projects

ECP5 does not use the CIN port of the CCU2C Carry Chain primitive. This port should not be connected to anything. If the port is connected, the Design Map stage will fail with an error message. If you see such a failure, correct the design in one of the following ways:

- Rerun synthesis. This should correct the problem if the CCU2C primitive is part of an IPexpress module.
- If the CCU2C primitive was added to your HDL manually, edit the code to remove the connection. See the following examples:

  In Verilog:

```
CCU2C addsub_0 (.A0(scuba_vlo), .A1(DataA[0]),
  .B0(scuba_vlo), .B1(DataB[0]), .C0(scuba_vhi),
```

```
  .C1(scuba_vhi), .D0(scuba_vhi), .D1(scuba_vhi),
  .CIN(), .S0(), .S1(Result[0]), .COUT(co0));
```

In VHDL:

```
signal tmp: std_logic := 'X';
cnt_cia: CCU2C
  generic map (INJECT1_1=> "NO", INJECT1_0=> "NO",
  INIT1=> X"0000", INIT0=> X"0000")
  port map (A0=>scuba_vhi, A1=>scuba_vhi, B0=>scuba_vhi,
  B1=>scuba_vhi, C0=>scuba_vhi, C1=>scuba_vhi,
  D0=>scuba_vhi, D1=>scuba_vhi,
  CIN=>tmp, S0=>open,S1=>open, COUT=>cnt_ci);
```

2.0.1 Projects

Several strategy options have new default values. If you are using Synplify Pro in integrated mode (running synthesis automatically in Diamond), check that the following settings are still as you want them. Also, check the setting of the Auto Hold-Time Correction option under Place & Route Design. Its default changed to On for all devices. This is still current in Synplify Pro versions.

Table 1: New Default Values for Synplify Pro for Lattice

Option	Before	Now
Fanout Limit is now Fanout Guide	100	1000
Export Diamond Settings to Synplify Pro GUI (new in 2.2)	Not available	No
Fix Gated Clocks and Fix Generated Clocks combined into new Clock Conversion	3 (converts and reports all sequential elements)	True (converts with no report)
Frequency	200	auto (blank means “auto”)
Number of Critical Paths	3	blank (unspecified)
Number of Start/End Points	0	blank (unspecified)
Output Preference File	False	True
Pipelining and Retiming	False	Pipelining Only
Resolved Mixed Drivers	True	False
Use Clock Period for Unconstrained I/O	True	False

1.4 Projects

For Diamond 1.4 and earlier, there might be some constraints that are not honored because of the Synplify Pro cross-probing feature. This EDIF renaming is usually related to bus names.

If such a problem occurs, you can turn off the renaming feature by placing the following line in the “Command line Options” text box of the Synplify Pro section of the active strategy:

set_option -syn_edif_array_rename 0

1.2 Projects

There were several enhancements for IP and MachXO2.

IP Incompatibilities

SPI4.2 2.7 is not compatible with Diamond 1.3 or later. If you are using this IP, check the Lattice Semiconductor Web site for a more recent version.

MachXO2 Changes

See if your design involves any of the following features:

- For EFB modules with user flash memory (UFM), regenerate the module.
- For IO_TYPE=PCI33 on a MachXO2-1200 or larger device, check if the CLAMP is using the default setting. With Diamond 1.3 the CLAMP default changes from ON to PCI and the I/O will be placed in bank 2. If you were using the default and still want the setting to be ON, you need to set it explicitly.
- For PCI33 MT 6.5 and PCI33 T 6.4 IP, either set the CLAMP to ON explicitly or choose a bigger package (256 or more).

1.1 or 1.0 Projects

There were several enhancements for IP and MachXO2.

IP Incompatibilities

The following IP versions are not compatible with Diamond 1.2 or later. If you are using any of these IP, check the Lattice Semiconductor Web site for a more recent version.

- Convolution Block Encoder 3.6
- DDR1 6.9
- DDR2 7.1
- DDR3 1.2.1
- DDR1_CP 1.1 with MachXO2
- DDR2_CP 1.1 with MachXO2
- Interleaver Deinterleaver 3.5
- PCI_MT_33 6.4
- PCIe RC Lite 1.2
- Tri-Speed MAC 3.4
- Viterbi Block Decoder 4.6

ECP5 Support

For ECP5U and ECP5UM designs, for better EBR and SERDES tuning and DSP/Distributed RAM control for constant process, it is a necessity to recompile your design using the current release of Diamond.

MachXO2 Support

Some aspects of the software support for MachXO2 designs have been improved. See if your design involves any of the following features:

- The 4K/7K design with PLL has a CIB-to-PLL jump change. If you are using this design, recompile it.
- The EFB simulation model has changed. If you are using the EFB module, rerun your simulation tests to see more accurate results.
- In the DDR_GENERIC module of IPexpress, the GDDRX1_RX.Aligned with PLL interface is no longer supported. If you are using such a module, use IPexpress to regenerate it without the PLL option.

Also, MachXO2 has IP evaluation capability and TransFR mode for all I/Os.

Migrating ispLEVER Projects

Diamond uses a different project structure than ispLEVER and cannot directly open an ispLEVER project. However, design projects created in ispLEVER can easily be imported into Diamond. The process is automatic except for the ispLEVER process properties, which are similar to the Diamond strategy settings, and some modules and IPs. All of your ispLEVER project source will be automatically handled.

Projects created using ispLEVER can be imported into Lattice Diamond through two different paths:

- On the Start Page, click **Import ispLEVER Project** (in the upper-left corner).
- From the File menu, choose **Open > Import ispLEVER Project**.

Follow the directions in the dialog box that opens to convert your ispLEVER project into a Lattice Diamond project.

Limitations to the import/conversion process include:

- NGO files in ispLEVER projects need to be manually copied into the Lattice Diamond project if the NGO files were originally copied into the ispLEVER project. For example, NGO files that were copied from Lattice IP generation.
- The .lpc files are replaced with .ipx files in Lattice Diamond. You need to regenerate your IP by double-clicking on the .lpc file. The resultant wizard will help you generate the new .ipx file, replacing the old .lpc file.

More information on importing ispLEVER projects can be found in the *Lattice Diamond User Guide*, online Help (see **Managing Projects > Importing ispLEVER Projects**), and training videos on the Lattice Web site.

Other Information Resources

Other available information resources for the Diamond software include the following.

- General Information: General information on Lattice Diamond can be found on the Lattice Web site at:

  www.latticesemi.com/latticediamond

- Online Help: Start Lattice Diamond and choose **Help > Lattice Diamond Help**.
- *Lattice Diamond User Guide*: This document can be found from a link on the Start Page view.
- Training Videos: Several short videos are available on different aspects of the Lattice Diamond software. These can be viewed online at:

  www.latticesemi.com/latticediamond

  Click the **Videos** tab.

System Requirements

The basic system requirements for Lattice Diamond are:

- Intel Pentium or Pentium-compatible PC, or AMD Opteron system support (Linux only)
- CPU with the SSE3 instruction set to run the Aldec Active-HDL Lattice Edition simulator
- One of the following operating systems:
  - Windows Vista (32-bit), Windows 7 (32-bit or 64-bit), Windows 8 (32-bit or 64-bit, including Windows 8.1), or Windows 10 (32-bit or 64-bit).
  - Red Hat Enterprise Linux 4, 5, 6 or 7. The host operating system is supported in 64-bit only.

    Version 5.3 of Red Hat Enterprise Linux has some extra installation requirements. See “Configuring Red Hat 5.3” on page 9.
- Approximately 5.75 GB free disk space
- RAM adequate for your FPGA design. For guidelines see “Memory Requirements” on page 8.
- Network adapter and, for a floating license, network connectivity

  A node-locked license is based on the physical (hard-coded) address provided by the network adapter. Network connectivity is not required for a node-locked license. In the absence of a network connection, you can

install the NWLink IPX/SPX protocol to force recognition of your NIC card ID (see the Installation Notice).

A floating license requires access to the license server, so both a network adapter and connectivity are required.

- JavaScript-capable Web browser
- Microsoft Internet Explorer 8 or higher if using the included Aldec Active-HDL Lattice Edition simulator
- Acrobat Reader 5.0 or later

Memory Requirements

Table 2 lists the minimum memory requirements and the recommended memory for the Lattice Semiconductor devices supported by Diamond.

On Windows, designing for the largest FPGAs may require more than the usual 2 GB of memory found in 32-bit computers. For help in extending your memory to 3 GB, see “Extending Memory on Windows” on page 9. Designing for LatticeECP3 with more than 95K LUT on a Windows system requires a 64-bit operating system.

Table 2: Recommended Memory

Device	Size	32-Bit Operating Systems		64-Bit Operating Systems	
		Minimum	Recommended	Minimum	Recommended
ECP5U/UM/UM5G	All	2 GB	3 GB	4 GB	6 GB
LatticeEC, LatticeECP	Up to 20K LUT	512 MB	768 MB	1 GB	1.5 GB
	Up to 50K LUT	768 MB	1 GB	1.5 GB	2 GB
LatticeECP2/M	Up to 20K LUT	768 MB	1 GB	1.5 GB	2 GB
	Up to 50K LUT	1 GB	1.5 GB	2 GB	3 GB
	Up to 100K LUT	1 GB	2 GB	2 GB	4 GB
LatticeECP3	Up to 95K LUT	2 GB	3 GB	4 GB	6 GB
	Up to 150K LUT	3 GB	4 GB	6 GB	8 GB
LatticeSC/M	Up to 40K LUT	768 MB	1 GB	1.5 GB	2 GB
	Up to 115K LUT	1 GB	2.5 GB	2 GB	5 GB
LatticeXP, LatticeXP2	Up to 20K LUT	512 MB	768 MB	1 GB	1.5 GB
	Up to 50K LUT	768 MB	1 GB	1.5 GB	2 GB
MachXO, MachXO2, MachXO3L	All	256 MB	512 MB	512 MB	1 GB
LIFMD (CrossLink)	All	256 MB	512 MB	512 MB	1 GB
Platform Manager, Platform Manager 2	All	256 MB	512 MB	512 MB	1 GB

Extending Memory on Windows

Designing for LatticeECP3 or ECP5 may require more than the 2 GB normally available with 32-bit Windows systems. But you can configure Windows to use up to 3 GB of memory.

Note that increasing the amount of memory available to applications decreases the amount available for the file cache, paged pool, and nonpaged pool, which can affect applications with heavy networking or I/O.

Use the `BCDEdit /set increaseuserva 3072` command to set the boot entry option to 3 GB. For details, see Microsoft article “BCDEdit /set”: msdn.microsoft.com/en-us/library/ff542202.aspx

Configuring Red Hat 5.3

Red Hat Enterprise Linux 5.3 has some extra requirements for Diamond:

- In addition to the basic installation of Red Hat 5.3, under Development/ Legacy Software Development, select:

  `1:gtk+-1.2.10-56.el5.i386 - GIMP Toolkit (GTK+) sb:(9 of 9)`

  Under Base System/Legacy Software Support, add the following to the default items:

  `Openmotif22-2.2.3-18.i386 - Open Motif runtime`

  Proper Diamond operation depends upon these libraries being installed.

- When installing the Red Hat Enterprise Linux version, be sure to install the PERL modules XML::Parser, XML::DOM, and XML::RegExp. These PERL modules are available at www.cpan.org.

Issues Fixed

The following known issues are fixed with this release. Their workarounds are no longer needed. For the complete list of known issues, see www.latticesemi.com/view_document?document_id=50676

STAPL file generation for POWR1014A has missing functions

All operations has been migrated from the former isUFW tool into the deployment tool.

Devices affected: All
Fixed_3.8
CR123793

Enhancement to STAPL file for multiple actions

Previously only one action per STP file can be supported. Currently, multiple ACTION per STP file can be embedded together.

i.e. <action name> [“<string>”] = <procname> [OPTIONAL | RECOMMENDED]

{,<procname>[OPTIONAL | RECOMMENDED] }

Devices affected: All
Fixed_3.8
CR124725

Diamond programmer iCE missing CRC and WAKEUP in file Read and Save

The CRC and WAKEUP functions for the iCE40 devices have been added to programmer.

Devices affected: iCE40
Fixed_3.8
CR125433

OTP (One Time Program) feature support for XO3L

OTP (One Time Programmable) operation has been added for this device which was not previously supported.

Devices affected: MachXO3L
Fixed_Release
CR126195

Programmer GUI Enhancements

The data and time as shown next to each device is now updated as soon as the programming file changes. Furthermore, when running an operation, the operation date and time is not updated in the log file.

Devices affected: All
Fixed_3.8
CR121462

Cable Server issue detecting device in Reveal

There was an issue in detecting the FTDI cable for CPLD devices prior to this fix.

Devices affected: MachXO2
Fixed_3.8
CR126611

Synplify Pro can stop working in certain cases

Synplify Pro can stop working when using certain designs with message:

Error Code [nlverify.c5767 Found functional mismatch...]

. As a workaround, add the following to your Verilog design file:

```
/* synthesis syn_preserve = 1 */;

at the module level.
```

Devices affected: All
Fixed_3.8
CR126287

Known Issues

Following are known issues with this release and workarounds for them. For the complete list, see:
www.latticesemi.com/view_document?document_id=50676

Programmer can take a long time to detect USB cable when using Linux RedHat 7 OS

If using Linux RedHat 7 operating system, if you scan a device or perform a programming operation and then attempt to detect a cable, it can take a minute or more for Diamond Programmer to detect the cable.

Versions affected: Diamond 3.8
Devices affected: All
CR126154

Programmer unable to program using I2C for USB 3.0 Pre Windows 8

USB 3.0 did not exist prior to Windows 8 and as such Microsoft did not have driver support for UB 3.0 host controllers.

Versions affected: Diamond 3.8
Devices affected: All
CR126896

Lattice Synthesis Engine may have long run-times in certain designs

Certain designs may create complicated FSM and mux-chain structures causing Lattice Synthesis Engine (LSE) to have long run-times. This can be avoided by setting the LSE Strategy Option “Resource Sharing” to False.

Versions affected: Diamond 3.8
Devices affected: All
CR125870

Contacting Technical Support

FAQs The first place to look. The Answer Database on the Lattice Semiconductor Web site provides solutions to questions that many of our customers have already asked. Lattice Applications Engineers are continuously adding to the Database.

Technical Support Assistance Submit a technical support case via www.latticesemi.com/techsupport.

For Local Support Contact your nearest Lattice Sales Office.

Trademarks