Lattice Radiant Software Guide for Lattice Diamond Users

February 12, 2018

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 Guide for Lattice Diamond Users

A primary feature of the Lattice Radiant™ software is to enable Lattice Diamond users to import their Field Programmable Gate Array (FPGA) designs easily and seamlessly. Lattice Diamond users will quickly recognize a familiar design environment, graphical user interface (GUI) layout and related tools such as Power Calculator and Reveal. However, the Radiant software includes many significant feature enhancements from Lattice Diamond, including:

- Upgraded database to provide fast and efficient software.
- Constraints utilizing industry standard Synopsys Design Constraints (SDC) format.
- Efficient and easy-to-use GUI.
- Unified timing analysis engine for faster design timing closure.

The objective of this guide is to provide guidelines for Lattice Diamond users so that they can quickly grasp the concepts of the new features in the Radiant software.

Table 1 lists major functions in a typical FPGA design flow, and compares Lattice Diamond to the Radiant software.

Table 1: Comparison of Lattice Diamond to Radiant Software

Lattice Diamond Design Function	Radiant Software Design Functions
Design Entry	
▶ Supports Hardware Design Languages (HDL) such as Verilog and VHDL.	▶ Supports HDL such as Verilog and VHDL. In addition, Radiant software provides the Template Editor tool which helps users implement Verilog or VHDL language constructs, device primitives and PMI templates. Refer to "Design Entry" on page 10.
▶ Supports primitives instantiation.	▶ Supports primitives instantiation. Refer to "Using Radiant Software Primitives" on page 10.
▶ Supports soft IP and modules with IP Express/Clarity Designer tools.	▶ Supports soft IP and modules with Radiant software IP Catalog. Refer to "Using Modules and Soft IP in Radiant Software" on page 10.
▶ Supports Parameterized Module Instantiation (PMI).	▶ Supports PMI. Refer to "Using Parameterized Module Instantiation" on page 13
Design Constraints	
▶ All timing and physical constraints defined in preferences.	▶ All timing and physical constraints defined in standard SDC. Refer to "Radiant Software Constraint Tools" on page 15.
▶ Supports HDL attributes.	▶ Supports HDL attributes. Refer to "Lattice Diamond Attributes Compared Radiant Software Attributes" on page 20.
▶ Supports preferences flow.	▶ Supports constraints flow. Refer to "Lattice Diamond Attributes Compared Radiant Software Attributes" on page 20.
Design Implementation	
▶ Supports standard FPGA design flow including synthesis, map, place-and-route (PAR), and bitgen.	▶ Supports standard FPGA design flow including synthesis, map, PAR, and bitgen. Refer to "Lattice Diamond Software and Radiant Software Process Flow" on page 25.
▶ Process flow supported.	▶ Process flow is implemented in Process Toolbar. Refer to "Lattice Diamond Software and Radiant Software Process Flow" on page 25.
▶ Supports strategies, which are options related to implementation tools.	▶ Supports strategies. Refer to "Design Analysis and Debug" on page 27.
▶ Project file extension is .ldf.	▶ Project file extension is .rdf. Refer to "Using the Wizard to Import a Lattice Diamond Project into Radiant Software" on page 22.

Table 1: Comparison of Lattice Diamond to Radiant Software (Continued)

Lattice Diamond Design Function	Radiant Software Design Functions
Design Analysis and Debug	
▶ Includes Reveal Inserter and Reveal Analyzer debug and analysis tools.	▶ Includes Reveal Inserter and Reveal Analyzer debug and analysis tools with features specific to iCE40 UltraPlus. Refer to "Design Analysis and Debug" on page 27.
▶ Includes Timing Analysis View.	▶ Includes new Timing Analysis View with new Constraints flow including Setup/Hold Endpoint Analysis. Refer to "Design Analysis and Debug" on page 27.
▶ Includes Power Calculator.	▶ Includes Power Calculator. Refer to "Power Calculator" on page 27.
▶ Includes Simulation Wizard.	▶ Includes Simulation Wizard. Refer to "Simulation Wizard" on page 27.
Tools	
▶ Includes a full suite of Lattice Diamond tools.	▶ Includes a full suite of Radiant software tools. Some tools have been combined for more efficiency. Refer to "Radiant Software Tools" on page 33

Design Entry

Hardware description languages (HDLs) such as Verilog and VHDL are fully supported in both Lattice Diamond and Radiant software. This section describes various design HDL methodologies used in both Lattice Diamond and Radiant software, and highlights similarities and differences.

Using HDL Languages in Radiant Software

Users can re-use their Lattice Diamond HDL languages such as Verilog and VHDL source files when migrating designs from Lattice Diamond to Radiant software.

Using Radiant Software Primitives

If your design contains Lattice Diamond primitives, new Radiant software primitives are available. For more information, refer to the *FPGA Libraries Reference Guide* in the Radiant software online Help.

Using Modules and Soft IP in Radiant Software

The IPexpress/Clarity tools in Lattice Diamond support two types of functional blocks: modules and intellectual property (IP). In the Radiant software, modules and IP can be created as part of a specific project or as a library for multiple projects.

- Modules are basic, configurable blocks that provide a variety of functions including I/O, arithmetic, memory, and more.
- Soft IP are more complex, configurable blocks that must first be downloaded and installed as a separate step before they can be accessed.

The major difference between Lattice Diamond and Radiant software is the soft IP flow. Figure 1 depicts the Lattice Diamond IPexpress flow. Figure 2 depicts the Radiant software IP Catalog flow.

Figure 1: Lattice Diamond IPexpress/Clarity Flow

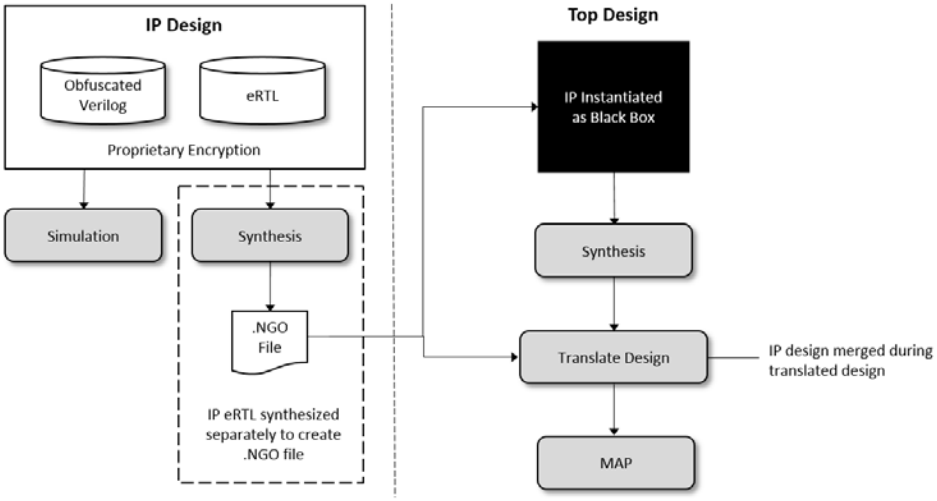
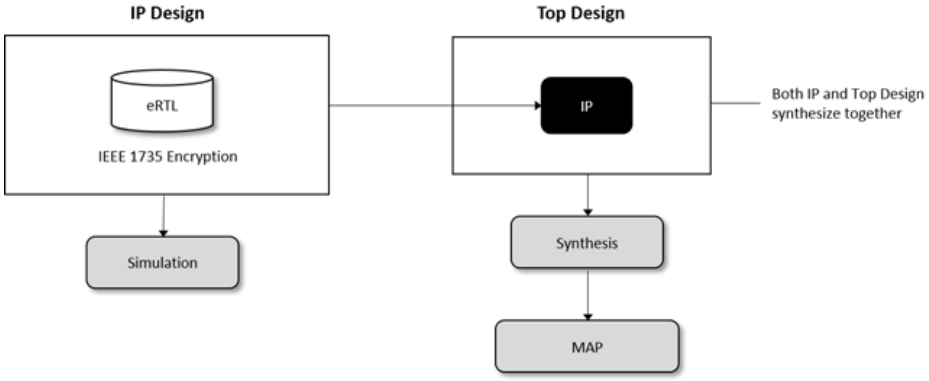


Figure 2: Radiant Software Soft IP Flow



Note

The Radiant software v1.0 IP Catalog currently only supports module functional blocks. Soft IP blocks will be available for Radiant software in a future release.

Table 2 lists differences between Lattice Diamond and Radiant software soft IP and module flow.

Table 2: Differences Between Lattice Diamond and Radiant Software Soft IP and Module Flow

Lattice Diamond	Radiant Software
A simulation testbench file (.tb) is included when module or IP is generated.	A simulation testbench file (.tb) is included when IP is generated (in a future Radiant software version).
Uses Blowfish encryption.	Uses IEEE 1735 encryption.

Table 2: Differences Between Lattice Diamond and Radiant Software Soft IP and Module Flow

Lattice Diamond	Radiant Software
▶ Encrypted eRTL file - used for synthesis. ▶ Obfuscated Verilog - used for simulation.	Encrypted modules/IP can be synthesized and simulated.
Many IP output files are generated.	Fewer IP output files are generated. Flow simplified in Radiant software.

Radiant Software IP Catalog Output Files for Modules IP Catalog creates the output files shown in Table 3 for modules under the specified Project Path. The <*file_name*> comes from the File Name specified in the Configuration tab.

Table 3: Radiant Software Soft IP Output Files

File Type	Definition	Function
<*file_name*>_bb.v	Verilog IP black box file	Provides the Verilog synthesis black box for the IP core and defines the port list.
<*file_name*>.cfg	IP parameter configuration file	Used for re-creating or modifying the core in the IP Catalog tool.
<*ifile_name*>.svg	Scalable Vector Graphics file	A graphic file used to display module/IP schematic and ports.
<*file_name*>_tmpl.v	Verilog template file	A template for instantiating the generated module. This file can be copied into a user Verilog file.
<*file_name*>_tmpl.vhd	VHDL module template file	A template for instantiating the generated module. This file can be copied into a user VHDL file.
<*ifile_name*>.v	Verilog module file	Verilog netlist generated by IP Catalog to match the user configuration of the module.

For more information on Radiant software soft IP and modules, refer to the following documents:

- Arithmetic Modules User Guide
- Memory Modules User Guide
- iCE40 UltraPlus sysCLOCK PLL Design and Usage Guide - Radiant
- iCE40 UltraPlus I2C and SPI Hardened IP Usage Guide - Radiant

Using Parameterized Module Instantiation

Parameterized Module Instantiation (PMI) is an alternate way to use some of the modules in Lattice Diamond and Radiant software. PMI allows you to directly instantiate a module into your HDL and customize it by setting parameters in the HDL.

Table 4 lists the differences in Lattice Diamond PMI parameters versus Radiant software PMI parameters.

Table 4: Lattice Diamond PMI Compared to Radiant Software PMI

Category	Module	Lattice Diamond	Radiant Software
Arithmetic	pmi_add	Yes	Yes
	pmi_sub	Yes	Yes
	pmi_mult	Yes	Yes
	pmi_mac	Yes	Yes
	pmi_complex_mult	Yes	Yes
	pmi_multaddsub	Yes	Yes
DSP	pmi_dsp	No	Yes
Memory	pmi_ram_dp	Yes	Yes
	pmi_ram_dq	Yes	Yes

For more information on Radiant software soft IP and modules, refer to the following documents:

- Arithmetic Modules User Guide
- Memory Modules User Guide
- iCE40 UltraPlus sysCLOCK PLL Design and Usage Guide - Radiant
- iCE40 UltraPlus I2C and SPI Hardened IP Usage Guide - Radiant

Design Constraints

One of the more prominent differences between Lattice Diamond software and Radiant software is the process of constraining a design. Using preferences is an important concept a Lattice Diamond software user needed to know in order to constrain a design. A Logical Preference File (.lpf) was used to constrain a design by tuning all aspects of logical, timing and physical constraints to improve performance.

In Radiant software, preferences have been replaced by the industry standard Synopsys Design Constraints for ease of use and improved compatibility between other third party vendor tools such as Synopsys Synplify Pro synthesis and Aldec Active-HDL simulation tools.

Differences of Lattice Diamond LPF Flow Compared to Radiant Software LDC/PDC Flow

In Lattice Diamond, there is no concept of a unified database and hence any changes to preferences either pre/post synthesis, MAP and all the through Place & Route for back annotation of constraints would require the whole flow to re-run from synthesis. See Figure 3.

Figure 3: Lattice Diamond Constraints Flow

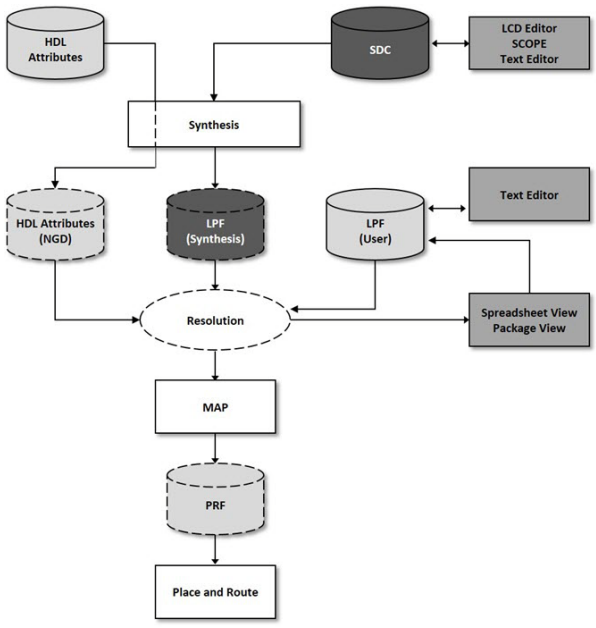
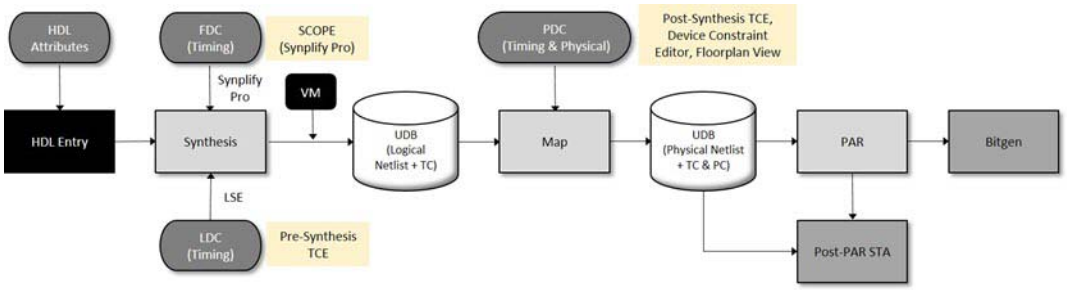


Figure 4 shows how the pre-and post-synthesis constraints are stored in the .ldc file and then kept in the Unified Database (UDB). For other stages in the flow, the PDC holds both post synthesis timing and physical constraints information. The major difference between this flow and Lattice Diamond is

that since all the constraints information is held in the UDB, if further timing/ physical constraints are entered in the Post-Synthesis Constraint Editor, synthesis need not be re-run and the newly entered constraints would be processed from MAP saving runtime and efficiency.

Figure 4: Radiant Software Constraints Flow



Radiant Software Constraint Tools

In Lattice Diamond, timing and physical preferences are applied using Spreadsheet View, Package View, Netlist View and Device View.

In the Radiant software:

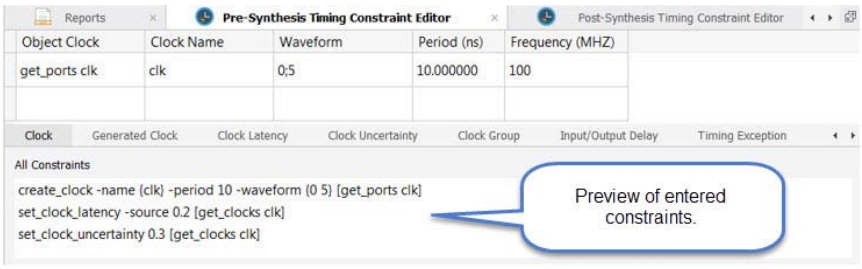
- Timing constraints are applied using the Timing Constraint Editors (Pre- and Post-Synthesis).
- Physical constraints are applied using the Device Constraint Editor (DCE).

Timing Constraints As shown in Figure 4, timing constraints are managed in SDC format in an .fdc file (for Synopsys Synplify Pro synthesis), or in an .ldc file for LSE.

The new Radiant software tools for pre/post synthesis timing constraints are:

- Pre-Synthesis Timing Constraint Editor (Figure 5) - Reads the HDL designs and produces timing constraints, saved in the .ldc file, based on HDL signal, port, and object names.

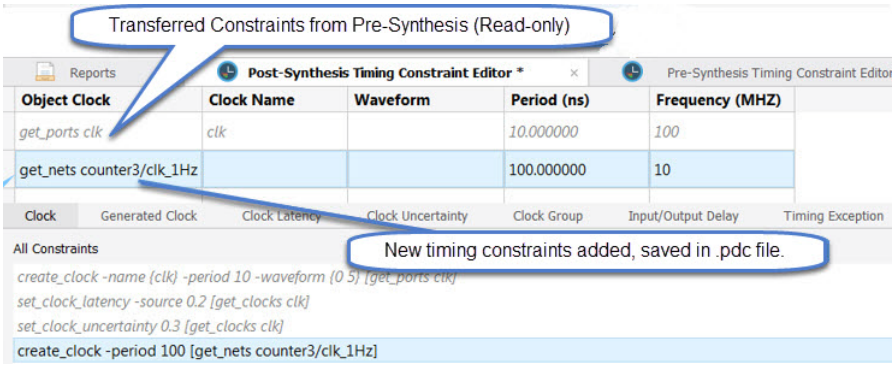
Figure 5: Pre-Synthesis Timing Constraint Editor



- Post-Synthesis Timing Constraint Editor (Figure 6) - Reads the post-synthesis netlists and produces timing constraints (saved in a .pdc file)

based on post-synthesis netlist signal, port and object names. This flow allows further tuning of timing constraints for the Place & Route process.

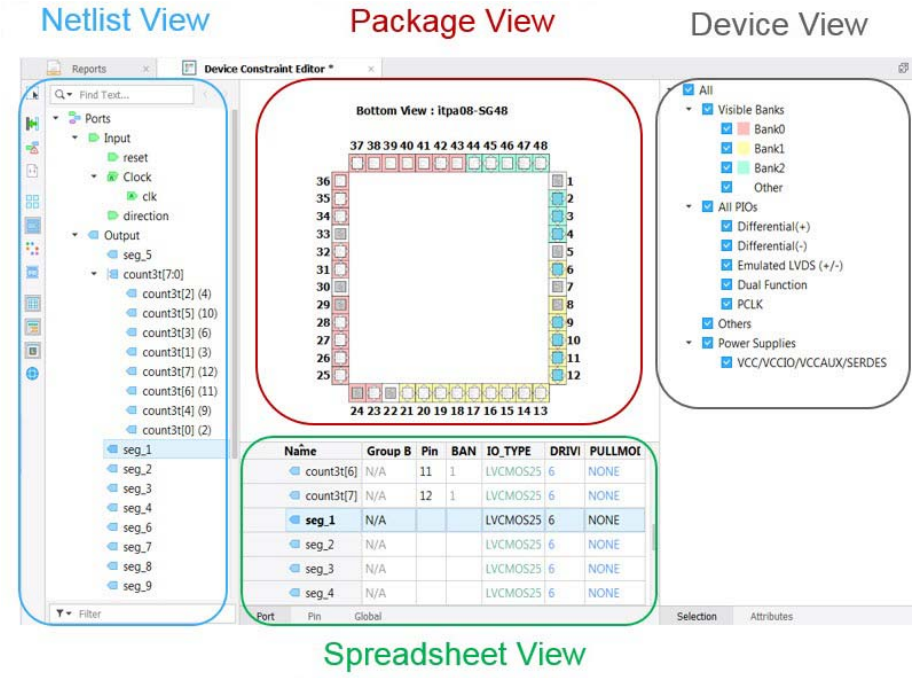
Figure 6: Post-Synthesis Timing Constraint Editor



The new post-synthesis timing constraints entered will override pre-synthesis constraints. Physical constraints are also entered in the .pdc file.

Physical Constraints The Radiant software Device Constraint Editor tool now combines Netlist, Package, Device and Spreadsheet views into one GUI for the primary purpose of entering physical (.pdc) constraints. This makes it easier to manage multiple tools and perform such actions as cross probing between multiple views. See Figure 7.

Figure 7: Device Constraint Editor



Each of the views in Figure 7 are used to apply constraints such as prohibiting pins and assigning IO_TYPEs.

Note the Floorplan View is used for creating GROUPs and REGIONs. For more information on Device Constraint Editor, refer to the Radiant software online Help topic **User Guides > Applying Design Constraints > Using Radiant Software Tools > Device Constraint Editor.**

Lattice Diamond Physical Preferences Compared to Radiant Software Physical Constraints

Table 5 shows the most commonly used Lattice Diamond software physical preference keywords and the equivalent Radiant software SDC commands to create a physical constraint.

Table 5: Commonly Used Lattice Diamond Preference Keywords Compared to Radiant Software SDC Commands

Lattice Diamond Physical Preference	Radiant software Physical Constraint	Description
Global, Net and Clock Attributes	ldc_set_attribute	Sets global attributes or specific attributes to the selected object.
GRP	ldc_create_group	Defines a user group.
IOBUF	ldc_set_port	Sets constraint attributes to the selected port
LOCATE	ldc_set_location	Places an object on a site or a user group into a region.
LOCATE VREF	ldc_create_vref	Defines a voltage reference site.
PROHIBIT	ldc_prohibit	Prohibits use of a site.
REGION	ldc_create_region	Defines a rectangular region area.
SYSCONFIG	ldc_set_sysconfig	Sets SysConfig attributes.
VCC_NOMINAL/ VCC_DERATE/ VOLTAGE	ldc_set_vcc	Sets a voltage to a bank or derates the core voltage.

All physical constraints and post synthesis timing constraints are stored in the .pdc file.

Lattice Diamond Timing Preferences Compared with Radiant Software Timing Constraints

Table 6 shows the most commonly used Lattice Diamond timing preference keywords and the equivalent Radiant software SDC timing constraints.

Table 6: Lattice Diamond Timing Preference Keywords Compared to Radiant Software SDC Timing Constraints

Lattice Diamond .lpf	Radiant software .sdc	Description
BLOCK INTERCLOCK DOMAIN/ BLOCK CLKNET/ CLKSKEWDISABLE	set_clock_groups	Defines different types of clocking schemes.
BLOCK PATH	set_false_path	Define false path cycles.
CLKSKEWDIFF	set_clock_latency	Defines arrival/departure times.
CLOCK_TO_OUT	set_output_delay	Defines output delay relative to a clock.
FREQUENCY/PERIOD	create_clock	Defines the design clocks.
FREQUENCY/PERIOD	create_generated_clock	Defines generated clocks.
INPUT_SETUP	set_input_delay	Defines input delay relative to a clock.
MAX_DELAY	set_max_delay	Defines maximum delay for timing paths.
MAX_DELAY MIN	set_min_delay	Define minimum delay for timing paths.
MULTICYCLE	set_multicycle_path	Defines multicycle clock cycles.
SYSTEM_JITTER/ CLOCK_JITTER (option)	set_clock_uncertainty	Defines uncertainty delays.

Note

When an INPUT_SETUP or CLOCK_TO_OUT is set in Lattice Diamond without first specifying a clock, Lattice Diamond will automatically create a virtual clock constraint and honor the delay. In the Radiant software, a user should first define a clock (create_clock), whether real or virtual, before using the set_input_delay constraint.

Radiant software constraints listed in Table 7 require Tcl commands to access object names such as cell, pin, net, port, or clock in a design.

Table 7: Tcl Commands

Object Access Types	Description
get_cells	Access cells in a design.
get_pins	Access pins in a design.
get_nets	Access nets in a design.
get_ports	Access ports in a design.
get_clocks	Access clocks in a design.
get_inputs	Access input signals in a design.
get_outputs	Access output signals in a design.
all_clocks	Access all clocks in a design.

Table 8 lists examples commonly used in Lattice Diamond preferences and equivalent Radiant software constraints in the SDC format.

Table 8: Examples of Timing Preferences in SDC Format

.LPF Example	.SDC Example
BLOCK PATH FROM PORT “abc” TO CELL “reg1/”;	set_false_path -from [get_ports abc] -to [get_cells reg1]
CLKSKEWDIFF CLKPORT “clk1” CLKPORT “clk2” 2 NS;	set_clock _latency 2 -source [get_clocks clk1]
CLKSKEWDISABLE CLKNET “clk1” CLKNET “clk2”;	set_false_path -from [get_clocks clk1] -to [get_clocks clk2]
CLOCK_TO_OUT PORT “out1” 8 ns CLKPORT=”clk2”;	set_output_delay (x-8) -clock [get_clocks clk2] [get_ports out1] *,**
FREQUENCY (PERIOD) NET “clk1” 100Mhz;	create_clock -period 10 -name clk1 [get_nets clk1]
INPUT_SETUP PORT “in_a” 4 ns CLKNET-”clk1”;	set_input_delay (x-4) -clock [get_clocks clk1] [get_ports_in_a] *, **
MAXDELAY FROM CELL “reg1” TO CELL “reg2” 5 NS	set_max_delay 5 -from [get_cells reg1] -to [get_cells reg2]
MULTICYCLE FROM CLKNET “clk1” TO CLKNET “clk2” 2 X;	set_multicycle_path 2 -from [get_clocks_clk1] -to [get_clocks clk2] **
SYSTEM_JITTER 1.0 NS	set_clock_uncertainty 1 [get_clocks *]

* : x = clock period

**: need to set create_clock first

For more information on the details of SDC constraints, refer the Radiant software online Help topic **Reference Guides > Constraints Reference Guide > Lattice Synthesis Engine Constraints > Synopsys Design Constraints**.

Lattice Diamond Attributes Compared Radiant Software Attributes

Synthesis attributes are the same in both Lattice Diamond and the Radiant software. The only difference is in the current device support in Radiant software only supports a subset of the attributes. Also, many attributes don't apply because they formerly apply to the Preference method of constraining where Radiant software is using standard .sdc format.

Note that any attributes not in Table 9 that were in Lattice Diamond are obsoleted and not used in Radiant software.

Table 9: Lattice Diamond Attributes Compared to Radiant Software Attributes

Lattice Diamond Software	Radiant Software
BBOX	BBOX
CLAMP	Target for future version.
DIFFRESISTOR	Target for future version.
DRIVE	DRIVE
GLITCHFILTER	Target for future version.
GSR	Target for future version.
HGROUP	Replaced by GRP
HYSTERSIS	HYSTERSIS
INBUF	INBUF
INIT	INIT
IO_TYPE	IO_TYPE
LOC	LOC
NOCLIP	NOCLIP
NOMERGE(SAVE)	NOMERGE
OPENDRAIN	Target for future version.
PULLMODE	PULLMODE
RBBOX	RBBOX
REGION	REGION
SLEWRATE	Target for future version.

Table 9: Lattice Diamond Attributes Compared to Radiant Software Attributes (Continued)

Lattice Diamond Software	Radiant Software
TERMINATION	Target for future version.
UGROUP	Replaced by GRP
VREF	Target for future version.

Design Implementation

This section describes how to import a Lattice Diamond project into the Radiant software. This section also describes similarities and differences between Lattice Diamond and the Radiant software.

- Lattice Diamond project files are not compatible with Radiant software project files. Lattice Diamond project files use .ldf extension. The Radiant software project files use .rdf extension.
- The Radiant software provides a wizard that allows you to import a Lattice Diamond project into the Radiant software.

Importing a Lattice Diamond Project into Radiant Software

The Radiant software allows you to import Lattice Diamond projects using an Import Diamond Project Wizard. Many features and setting from Lattice Diamond can be directly imported into the Radiant software, allowing you to avoid creating a new project from scratch.

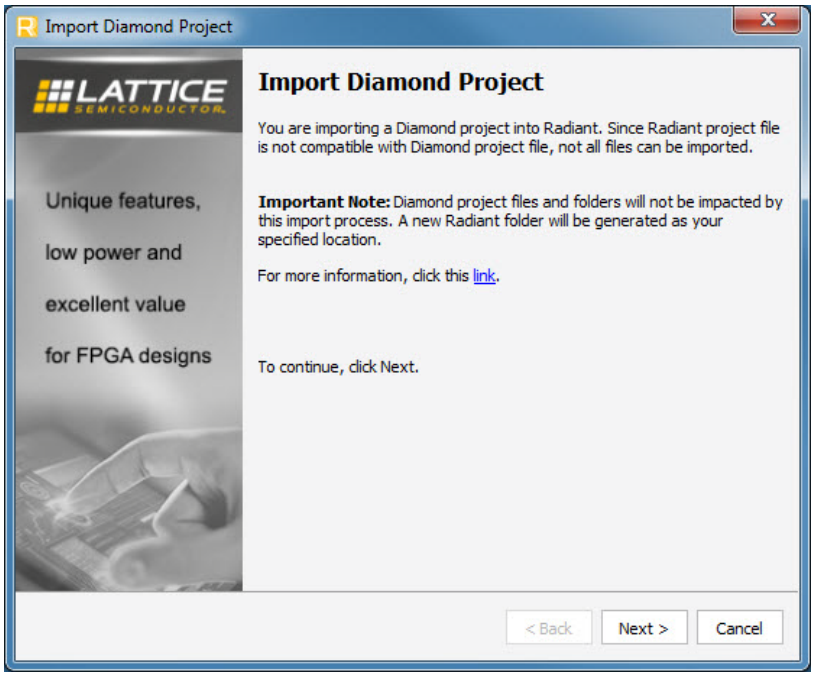
Using the Wizard to Import a Lattice Diamond Project into Radiant Software

Design projects created in Lattice Diamond can be imported into the Radiant software using the Import Diamond Project wizard, shown in Figure 8. Imported Lattice Diamond projects are targeted to devices supported in the Radiant software. Lattice Diamond design preferences will be converted into the Radiant software design constraints.

The Radiant software project file extension is .rdf, while the Lattice Diamond project file extension is .ldf.

For more information on using the Import Lattice Diamond Project Wizard, in the Radiant software online Help topic **User Guides > Managing Projects > Importing Lattice Diamond Projects**.

Figure 8: Import Diamond Project Wizard



Importing Lattice Diamond Strategies into Radiant Software

A strategy provides a unified view of all the options related to implementation tools such as synthesis, map, and place and route. Strategy options are listed in the Strategy dialog box in both Lattice Diamond and Radiant software.

Lattice Diamond strategies are imported into Radiant software if the strategies are unchanged. If Radiant software does not support Lattice Diamond strategies, those Lattice Diamond strategies will be ignored. Users should check all strategies after they have been imported from Lattice Diamond into Radiant software.

Compatible Settings and Files

The following Lattice Diamond settings and source can be imported into the Radiant software:

HDL Source Files and Properties HDL source files such Verilog (.v) and VHDL (.vhd) can be imported into the Radiant software.

Synplify Pro SDC Files Synplify Pro Timing Constraint Files (SDC) can be imported into the Radiant software.

Lattice Synthesis Engine Timing Constraint File LSE timing constraints can be imported into the Radiant software. Some records with bus do not use standard Tcl syntax and will be discarded when loaded into the Radiant software database.

Implementations All implementations and their settings, such as top-level unit and synthesis tool selection, are imported into Radiant software.

Strategies All supported strategies are imported into the Radiant software.

Reveal Inserter Settings Reveal Inserter debug files (.rvl) are imported from Lattice Diamond into the Radiant software. However, due to differences between devices, not all options or features may be available in the Radiant software Reveal Inserter.

Note

The Radiant software v1.0 Reveal Inserter supports only one Reveal debug module (core) with one clock domain per design. The Lattice Diamond Reveal Inserter supports multiple Reveal debug modules (cores) with multiple clock domains per design.

When importing Lattice Diamond designs into the Radiant software, you must remove multiple Reveal modules (cores) and use only one Reveal module (core) in the Radiant software v1.0.

Simulation Wizard All Simulation Wizard scripts are imported into the Radiant software.

Incompatible Settings and Files

Some settings and files from Lattice Diamond are not compatible with the Radiant software. The following Lattice Diamond settings and files cannot be imported:

Target Device Currently no devices supported by Lattice Diamond are supported in the Radiant software. Therefore, it will be necessary to select a Radiant software-supported device.

Note

For the Radiant software v1.0, iCE40 UltraPlus is the only supported device.

Lattice Diamond Preference (.lpf) Files Lattice Diamond preference files (.lpf) are not imported into the Radiant software. The Radiant software does not support Lattice Diamond preferences. All Lattice Diamond preferences must be manually converted into the Radiant software constraints.

Soft IP and Module (.ipx) Files Soft IP and Module package instance files (.ipx) are not imported because the IP flow in the Radiant software differs from Lattice Diamond. Refer to "Using Modules and Soft IP in Radiant Software" on page 10 for a description of the differences.

Reveal Analyzer File Because of differences between devices, Lattice Diamond Reveal Analyzer (.rva) files imported into the Radiant software v1.0 may not be useful. New waveforms generated with the Radiant software v1.0 Reveal Analyzer will overwrite .rva files imported from Lattice Diamond.

Power Calculator File Because of differences between devices, Lattice Diamond Power Calculator (.pcf) files cannot be imported into the Radiant software v1.0.

Programmer Project File Because of differences between devices, Programmer Project Files (.xcf) files cannot be imported into the Radiant software v1.0.

Unsupported Design Source in Radiant Software

The following design source files are not supported in the Radiant software and cannot be imported:

EDIF Design Entry Files Electronic Design Interchange Format (EDIF) is not supported in the Radiant software. Therefore, such files as .edf and .edn cannot be imported into the Radiant software.

Schematic files The Radiant software does not support schematic entry. Therefore, schematic files cannot be imported into the Radiant software.

Clarity Design Files The Radiant software does not support ECP5 devices or Clarity (.sbx) files. Therefore, .sbx files cannot be imported into the Radiant software.

Platform Designer Files The Radiant software does not support Platform Manager or Platform Manager 2 devices. Therefore, Platform Designer (.ptm) files cannot be imported into the Radiant software.

Lattice Diamond Software and Radiant Software Process Flow

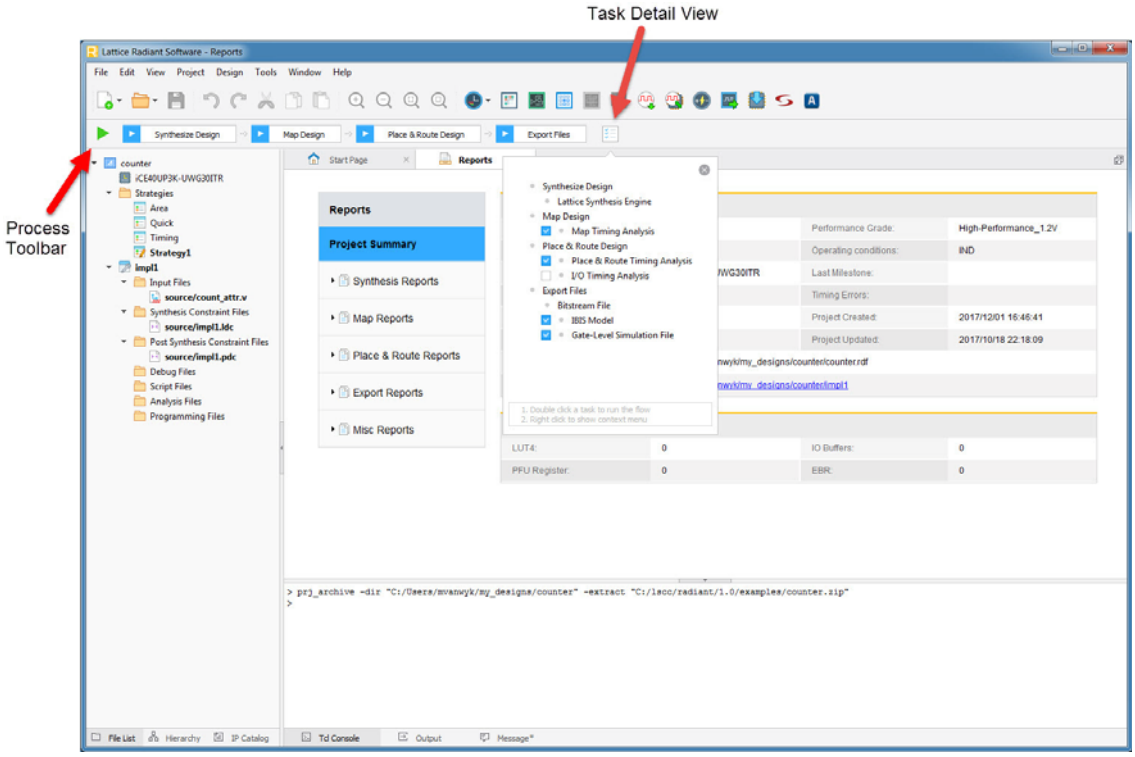
The Radiant software graphical user interface has changed from the Lattice Diamond. The Radiant software main window is shown in Figure 9.

Processes are controlled using the Process toolbar. All process tasks can be viewed and selected using the Task Detail view.

- The Radiant software also supports Lattice Diamond project features such as Implementations and Strategies.
- A major difference between Lattice Diamond and Radiant is the constraint language and flow, as discussed in “Design Constraints” on page 14.

- Report View in the Radiant software is similar to Report View in Lattice Diamond, but the Radiant software uses different report formats.

Figure 9: Radiant Software Main Window



Design Analysis and Debug

With the exception of the Timing Analysis tool, the debug and analysis tools within the Radiant software have remained unchanged. For more information on Timing Analysis, refer to "Timing Analysis" on page 29.

Simulation Wizard

The Simulation Wizard remains the same. A user is guided through the simulation setup process as before.

Power Calculator

Power Calculator is unchanged from Lattice Diamond.

Reveal

The Reveal Inserter and Analyzer GUI also remain unchanged. Only device specific features for iCE40 UltraPlus are supported. For example, only soft JTAG core is supported and only one debug core can be inserted due to resource limitations.

Timing Analysis Differences Between Lattice Diamond and Radiant Software

The following table lists differences between Lattice Diamond and Radiant software.

Lattice Diamond	Radiant Software
Timing is modeled as register-to-register within the FPGA boundary. Constraints are modeled in four methods: ▶ Input to output paths (MAXDELAY) ▶ Input to register paths (INPUT_SETUP) ▶ Register to register paths (FREQUENCY/ PERIOD) ▶ Register to output paths (CLOCK_TO_OUT)	Timing is modeled register-to-register from outside of the FPGA boundary. ▶ Register to register paths (create_clock, set_input/output_delay, set_clock_latency)
.html links to timing paths.	.html based Table of Contents reporting for easy navigation.

Lattice Diamond	Radiant Software
Critical Endpoints: Critical paths listed for each single path analyzed. (Need to analyze every path to determine critical endpoints.)	Lists top critical endpoints irrespective of the constraint or path.
Embedded within some of the timing reports.	Unconstrained clocks / paths clearly listed.

Static Timing Analysis Concepts in Lattice Diamond and Radiant Software

Lattice Diamond modeled timing paths from within the FPGA boundary. This explains why an assortment of timing preferences were available to model timing paths that began and ended within the FPGA. Preference constraints such as CLOCK_TO_OUT, INPUT_SETUP and MAXDELAY were used to constrain these paths with the assumptions that any timing information for arriving exterior logic is known.

Figure 10: Lattice Diamond Constraint Timing Modeling

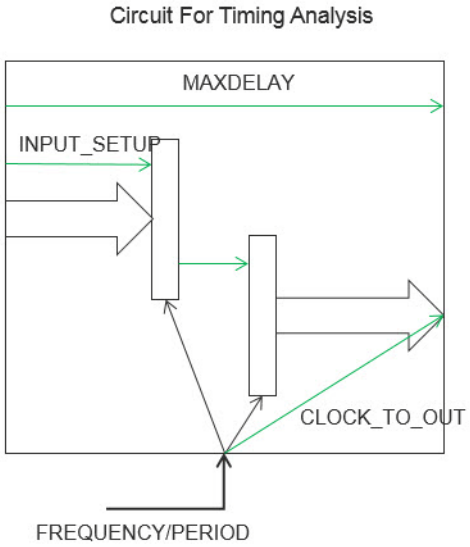
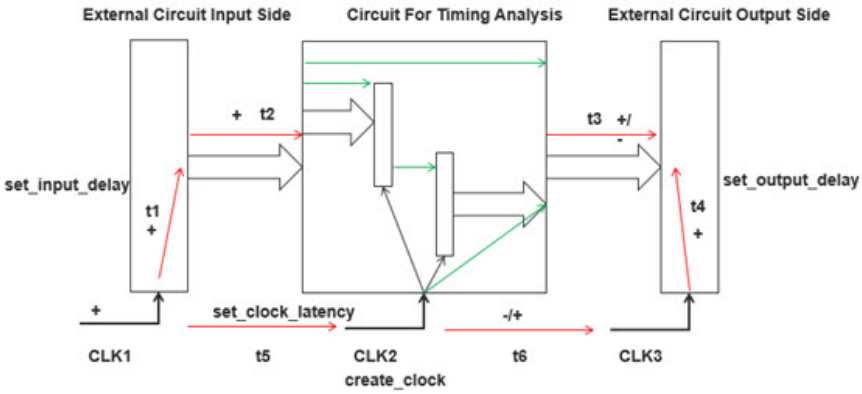


Figure 11 shows an overview of how a circuit is modeled for timing within the Radiant software. As mentioned earlier, the key difference with Radiant software is that the path is modeled from register-to-register across the entire path which can begin and end outside of the FPGA boundary. As a result, standard .sdc constraints are used to model the input and output delays (i.e. set_input/output_delay) from external to the FPGA boundary, and appropriate latency (ldc_set_latency) can be specified from external to internal of the chip boundary.

Critical endpoints list the paths for clocking constraints that have no departure or arrival element.

Unconstrained endpoints show paths that have no constraints defined even though the path is clocked, indicating that there is no departure, arrival primitive or port.

Figure 11: Sample Circuit Analysis



For more information on Device Constraint Editor, refer to the Radiant software online Help topic **User Guides > Analyzing Static Timing > Using Timing Analysis View > Timing Analysis View Feature > Timing Analysis View Main Window Tabs**.

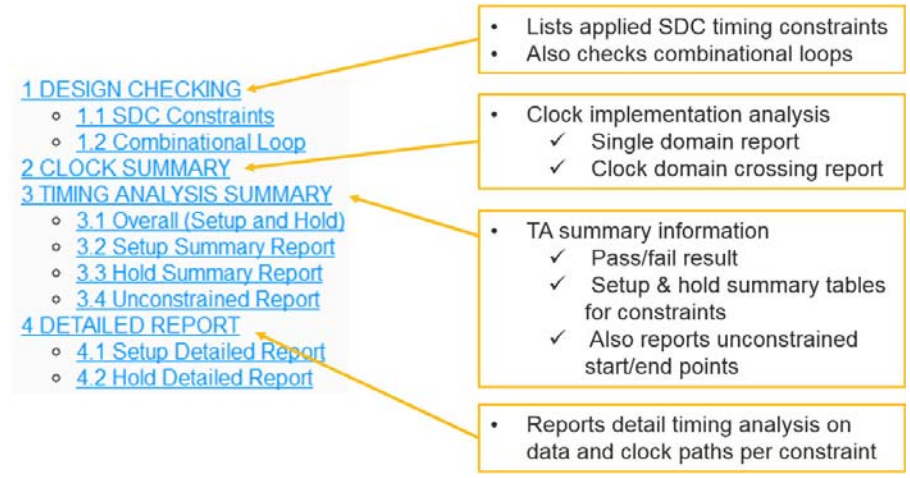
Timing Analysis

The Radiant software makes it easier to analyze and navigate design constraints. In addition to Preference Reports being replaced by .sdc constraints, Timing Reports are now organized in a table of contents with click-able .html links that takes you to a specific section for analysis.

Once static timing analysis is performed, .twr and .html report files are created. The report structure of the .sdc constraints is also different than in Lattice Diamond.

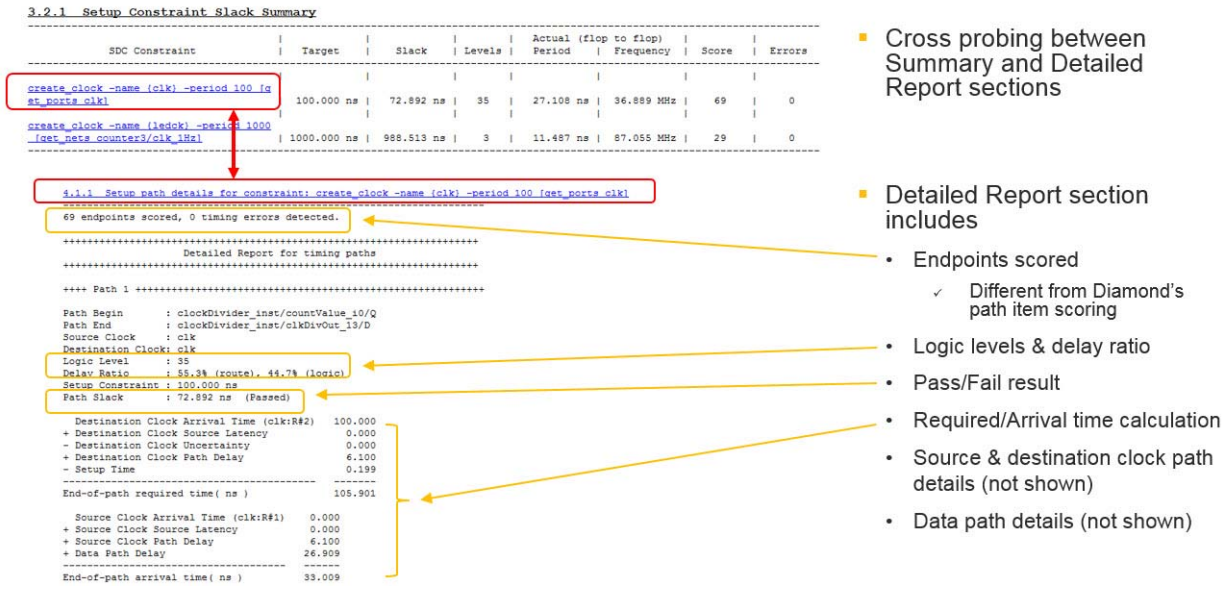
The Radiant software Timing Report is grouped into multiple hyper-linked categories, as shown in Figure 12.

Figure 12: Radiant Software Timing Report



Upon clicking a link, a comprehensive report is shown detailing all calculations, as well as indicating whether a positive or negative slack has occurred. Figure 13 shows an example of this.

Figure 13: Detailed Timing Report



A timing report consists of three primary sections:

- Paths for Various Constraint - Setup and Hold constraints based on SDC constraints applied.
- Critical Endpoint Summary - Detailed path summary with total slack, delay and clock information calculations based on endpoint element.
- Unconstrained Endpoints summary - A calculation resulting from no paths to an end point or because the end point was not properly constrained.

For more information on the details of Analyzing Static Timing, refer to the Radiant software online Help topic **User Guides > Analyze Static Timing > Running Timing Analysis > Timing Analysis Report File > Timing Analysis Report**.

Timing Analysis View

The redesigned Timing Analysis View features a detailed analysis of all constraints, as shown in Figure 14. To view path, path detail, and path calculations:

1. Click on a constraint in the Constraint box in the upper left.
2. Click on a path in the Path Summary box in the lower left.

The constraint's path, path detail and path calculations are shown on the right.

Figure 14: Timing Analysis View for Radiant Software

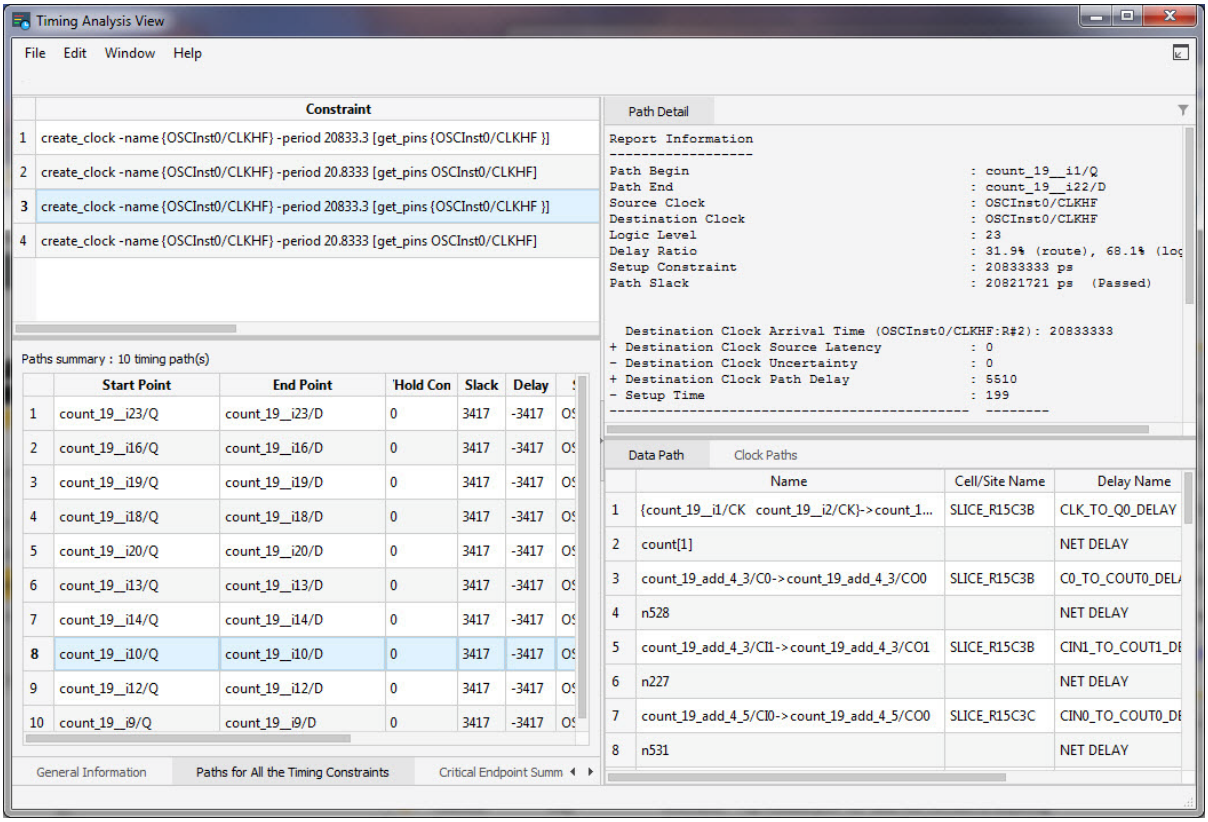
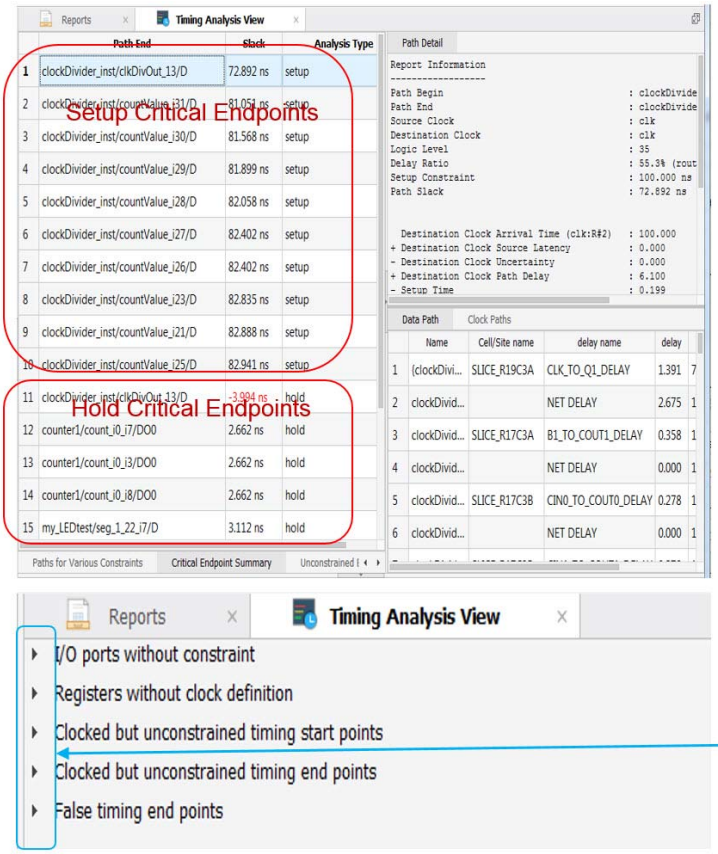


Figure 15 elaborates on Critical Endpoint Summary and Unconstrained Endpoint Summary, the two primary types of Timing Analysis available, that can assist you in debugging timing issues.

Figure 15: Types of Timing Analysis



- Critical Endpoint Summary
  - Reports worst slack paths on the top from the entire timing analysis
  - Number of paths user configurable (set to 10 shown)
  - Useful for a quick and bigger-picture TA result
- Unconstrained Endpoint Summary
  - Reported with 5 sub-sections
  - I/O, Registers, Clocks
  - False path endpoints are also reported here (Beta 5)
  - Expand each section to see the list of unconstrained endpoints

The Timing Analysis View also provides the ability to search elements and filter out specific paths for timing analysis. The following example shows how to analyze a clock domain crossing from *clk_1Hz* to *clk*:

1. Set the source (clk_1Hz).
2. Enter "**/Q**' in the filter box to identify start points.
3. Select all Q pins.
4. Move selection to the "From box >".
5. Enter "**/D**" in the filter box to get destination points.
6. Select all D pins.
7. Move selection to the "To box >".
   - All associated start and end points are listed in which a patch can be selected for analysis.
   - Path detail and calculation data are shown.
   - Apply the appropriate constraints for performance tuning.

Radiant Software Tools

The Radiant software provides an improved user friendly, concise, and efficient tool and design structure over its predecessor Lattice Diamond.

Lattice Diamond Compared with Radiant Software Tools

Table 10 lists the similarities and differences between Lattice Diamond and Radiant software tools.

Table 10: Tools Comparison Between Lattice Diamond and Radiant Software

Lattice Diamond Tools	Radiant Software Tools
▶ Spreadsheet View ▶ Package View ▶ Device View ▶ Netlist View	▶ Device Constraint Editor (DCE) ▶ The Device Constraint Editor tool combines Spreadsheet, Package, Device and Netlist views into one tool for the primary purpose of entering physical (.pdc) constraints. This cross probing between multiple views. Note that some views in this tool can be hidden to maximize work space. For more information on Device Constraint Editor, refer to the Radiant software online Help topic **User Guides > Applying Design Constraints > Using Radiant Software Tools > Device Constraint Editor**.
▶ Netlist Analyzer	▶ Netlist Analyzer: ▶ Features are similar to Lattice Diamond.
▶ NCD View	▶ Not Supported
▶ IPexpress	▶ IP Catalog: ▶ Features are similar to Lattice Diamond, but with major enhancements added, such as IEEE 1735 encryption support for synthesis and simulation.
▶ Reveal Inserter and Analyzer	▶ Reveal Inserter and Analyzer: ▶ Features are similar to Lattice Diamond. The Radiant software uses soft jtag, which limits to one Reveal core and single device. In Radiant software 1.0, the Power-On Reset (POR) Debug feature is disabled.
▶ Floorplan View	▶ Floorplan View: ▶ Major features are similar to Lattice Diamond. Enhancements include a World View feature.

Table 10: Tools Comparison Between Lattice Diamond and Radiant Software

Lattice Diamond Tools	Radiant Software Tools
▶ Physical View	▶ Physical View: ▶ Allows you to easily locate target objects using logic hierarchy. Supports Netlist Widget, Property Widget, and World View. ▶ For IOLOGIC & I/O pads, the ASIC View replaces Logic Block View and Non-schematic table format. ▶ The Radiant software Physical view supports cross probing between Floorplan View and Physical View and from DCE to Floorplan View. ▶ The Radiant software Physical View also supports one-way probing for ports. ▶ The Radiant software Physical View does not support switchbox and wire level display.
▶ Timing Analysis View	▶ Timing Analysis View: ▶ This tool has significant changes. For more information, refer to "Timing Analysis View" on page 31.
▶ Power Calculator	▶ Power Calculator: ▶ Features are similar to Lattice Diamond.
▶ Lattice Design Constraint (.LDC) Editor	▶ Timing Constraint Editor, includes: ▶ Pre-Synthesis Timing Constraint Editor tool used for generating or editing pre-synthesis timing constraints which are stored in .ldc file. ▶ Post Synthesis Timing Constraint Editor Tool used for generating or editing post-synthesis timing constraints, stored in .pdc file. These constraints may override the pre-synthesis timing constraints. For more information, refer to "Radiant Software Constraint Tools" on page 15.
▶ Power Calculator	▶ Power Calculator: ▶ Features are similar to Lattice Diamond.
▶ ECO Editor	▶ Will be supported in a future version.
▶ Programmer	▶ Programmer: ▶ Features are similar to Lattice Diamond. ▶ Includes Programmer, Deployment Tool, Download Debugger, and Programming File Utility. ▶ Model 300 Programmer is not supported.
▶ Partition Manager	▶ Not supported.
▶ Synplify Pro for Lattice	▶ Synplify Pro for Lattice: ▶ Features are similar to Lattice Diamond. For more information on Synplify Pro for Lattice, from the Radiant software start page, click: **User Guides > Synopsys Synplify Pro for Lattice User Guide** and **User Guides > Synopsys Synplify Pro for Lattice Reference Manual**.

Table 10: Tools Comparison Between Lattice Diamond and Radiant Software

Lattice Diamond Tools	Radiant Software Tools
▶ Active-HDL Lattice Edition	▶ Active-HDL Lattice Edition (Windows only): ▶ Features are similar to Lattice Diamond. For more information about Active-HDL Lattice Edition, from the Radiant software start page, click: **User Guides > Active-HDL On-line Documentation (Windows only)**.
▶ Run Manager	▶ Will be supported in a future version of Radiant software.
▶ Simulation Wizard	▶ Simulation Wizard: ▶ Features are similar to Lattice Diamond.

Revision History

The following table gives the revision history for this document.

Date	Version	Description
02/12/2018	1.0	Initial Release.