Lattice Radiant Software Design Flow Overview for Xilinx Vivado Users

User Guide

FPGA-UG-02165-1.0

July 2022

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults and associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Contents

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Figures

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Tables

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CDT	C/C++ Development Tools
CPLD	Complex Programmable Logic Device
ECO	Engineering Change Order
EDA	Electronic Design Automation
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HDL	Hardware Description Language
IDE	Integrated Development Environment
IES	Incisive Enterprise Simulator
LSE	Lattice Synthesis Engine
OS	Operating System
RTL	Register Transfer Level
SDK	Software Development Kit
SoC	System on-chip
VCS	Verilog Compiler Simulator

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1. Introduction

The Lattice Semiconductor Field Programmable Gate Array (FPGA)/Complex Programmable Logic Device (CPLD) design flow is similar in conception and implementation to Xilinx®[1] FPGA design flow. At its core, a hardware description language (HDL) code of the register transfer level (RTL) can be imported into one of Lattice's design software and then configured to one of Lattice's FPGA.

This document guides Xilinx FPGA designers familiar with Xilinx® Vivado®[1] software in migrating existing designs to Lattice Radiant™ software. It also highlights some of the differences and similarities between the design flows of Xilinx Vivado and Lattice Radiant software.

This user guide starts with an overview and comparison of the design software and a mapping of the tools and file extensions between the two. The next section compares the design flows of the design software. The succeeding sections provides a step-by-step walkthrough about the following:

- Project creation and management,
- Design entry
- Compilation flow
- Simulation
- Programming/Configuration

For more details on Lattice Radiant software design flow, refer to Lattice Radiant software video tutorials listed in Table 1.1.

Table 1.1. Lattice Radiant Software Video Tutorial Links

Introduction to Lattice Radiant Software	Lattice Radiant software introductory video featuring key easy design experience features.
Lattice Radiant Software Tool Flow – Part 1	Lattice Radiant software Tool Flow training series with step-by-step procedure on full design flow and key features.
Lattice Radiant Software Tool Flow – Part 2	
Lattice Radiant Software Tool Flow – Part 3	
Setting up a Floating License Tutorial	Instructional video on how to setup a floating license on a server and how to setup environment variables on a client machine to access the server license.

[1] *Xilinx, the Xilinx logo, and Vivado are trademarks of Xilinx Corporation or its subsidiaries.*
Xilinx Vivado software is used to show the differences and similarities between the design flows of Vivado and Lattice Radiant software.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2. Design Software Overview

A design software is required to develop and implement FPGA/CPLD designs. The Lattice Radiant software features the following characteristics to reduce the design's time-to-market:

- Full-featured – Able to offer all necessary tools for design development
- High-performance – Able to perform powerful optimizations and analyses
- Intuitive user interface – Able to provide the best user experience with a graphical user interface that is both modular and wizard driven

The Lattice Radiant software is a solution for low-end to mid-range FPGA designs. It features a leading-edge software design environment for cost-effective, low-power Lattice FPGA architectures. The Lattice Radiant software integrated tool environment provides a modern, comprehensive user interface for controlling the Lattice Semiconductor FPGA implementation process.

The Lattice Radiant software uses an expanded project-based design flow and integrated tool views so that design alternatives and what-if scenarios can be created and analyzed. The Implementations and Strategies concepts provide a convenient way to try alternate design structures and manage multiple tool settings. System-level information—including process flow, hierarchy, and file lists—is available, along with integrated HDL code checking and consolidated reporting features. A fast Timing Analysis loop and Programmer provide capabilities in the integrated framework. The cross-probing feature and the shared memory architecture ensure fast performance and better memory utilization. The Lattice Radiant software is highly customizable and provides TCL scripting capabilities from either its built-in console or from an external shell.

Most of the capabilities available in the Vivado software are available as well in Lattice Radiant software. However, the terminology of the individual tools, features, and file formats may differ from one software to the other. Table 2.1 lists some of the tools, features, and file formats in Vivado and its corresponding name in Lattice Radiant software.

Table 2.1. Mapping of Vivado and Lattice Radiant Software Tools, Features, and File Extensions

Description	Vivado	Lattice Radiant Software
Optimization or implementation settings	Run strategies	Design strategies
Design entry	HDL file IP Catalog IP Integrator	HDL file IP catalog
Design constraints	Vivado IDE Constraints Manager Timing Constraints Wizard Timing Constraints Window	Device/Physical Constraint Editor Timing Constraint Editor
Synthesis tools	Vivado Design Suite Synthesis	Lattice Synthesis Engine (LSE) Synplify Pro®
Schematic tools	Schematic	Netlist Analyzer HDL Analyst on Synplify Pro
Place and Route tools	Vivado Implementation	Place and Route
Supported simulators	Vivado simulator Mentor Graphics Questa Advanced Simulator Mentor Graphics ModelSim Simulator Cadence® Incisive® Enterprise Simulator (IES) Synopsys® Verilog Compiler Simulator® (VCS) Aldec Rivera-PRO™ Simulator Aldec Active-HDL™ Cadence Xcelium Parallel Simulator	Cadence Xcelium Synopsys VCS Siemens Questasim Siemens ModelSim
Simulation levels	Behavioral Simulation Post-Synthesis Simulation (Functional and Timing) Post-Implementation Simulation (Functional and Timing)	RTL Simulation Post-Synthesis Simulation Post-Route Gate-Level Simulation Post-Route Gate-Level+Timing Simulation

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Description	Vivado	Lattice Radiant Software
Power analysis	Xilinx Power Estimation Report Power	Power Calculator
On-chip debug tool	Vivado logic analyzer (Hardware Manager)	Reveal™
Hardware programming tool	Vivado Programmer (Hardware Manager)	Programmer
Project file extension	.xpr	.rdf
IP configuration file extension	.xci	.ipx
Soft processor development and generator	Vitis™ Core Development Kit Xilinx Software Development Kit (XSDK)	Lattice Propel™ Builder
Soft processor	Zynq MicroBlaze™	RISC-V

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. FPGA Design Flow Overview

When creating designs for Field Programmable Gate Arrays (FPGAs), the Lattice and Xilinx software tools have similarities in terms of concepts, approach, and functionality. Lattice Radiant software framework technology uses the typical FPGA design flow (see Figure 3.1) that adheres to a sequence of steps which initially requires setting up the design environment and ends with the generation of programming files that will be used to program the hardware. In cases where Xilinx designers who are familiar with Xilinx Vivado would like to convert the existing Xilinx Vivado designs to Lattice Radiant software environment, they can simply import the HDL (hardware description language) files to Lattice Radiant and begin implementing the project-based methodology (see Figure 3.2).

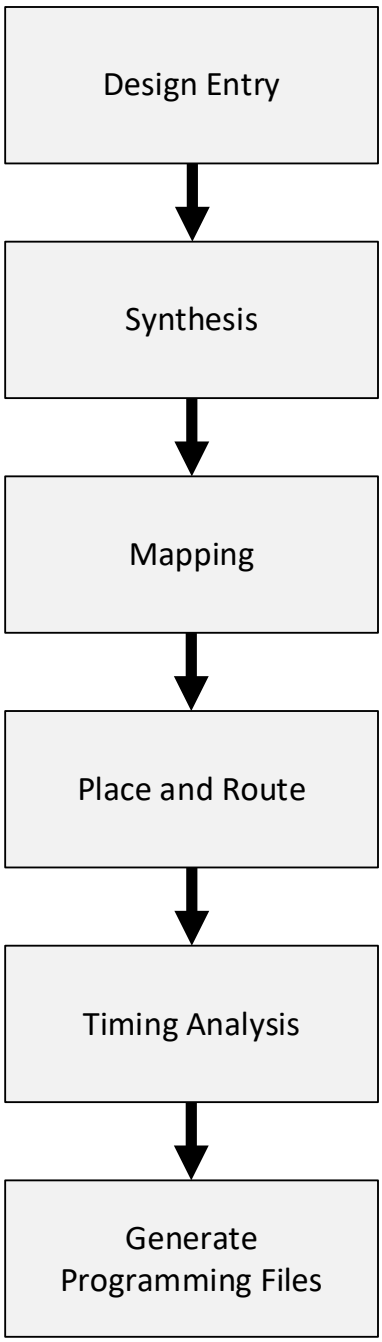


Figure 3.1. Typical FPGA Design Flow Used by Xilinx Vivado and Lattice Radiant Software

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

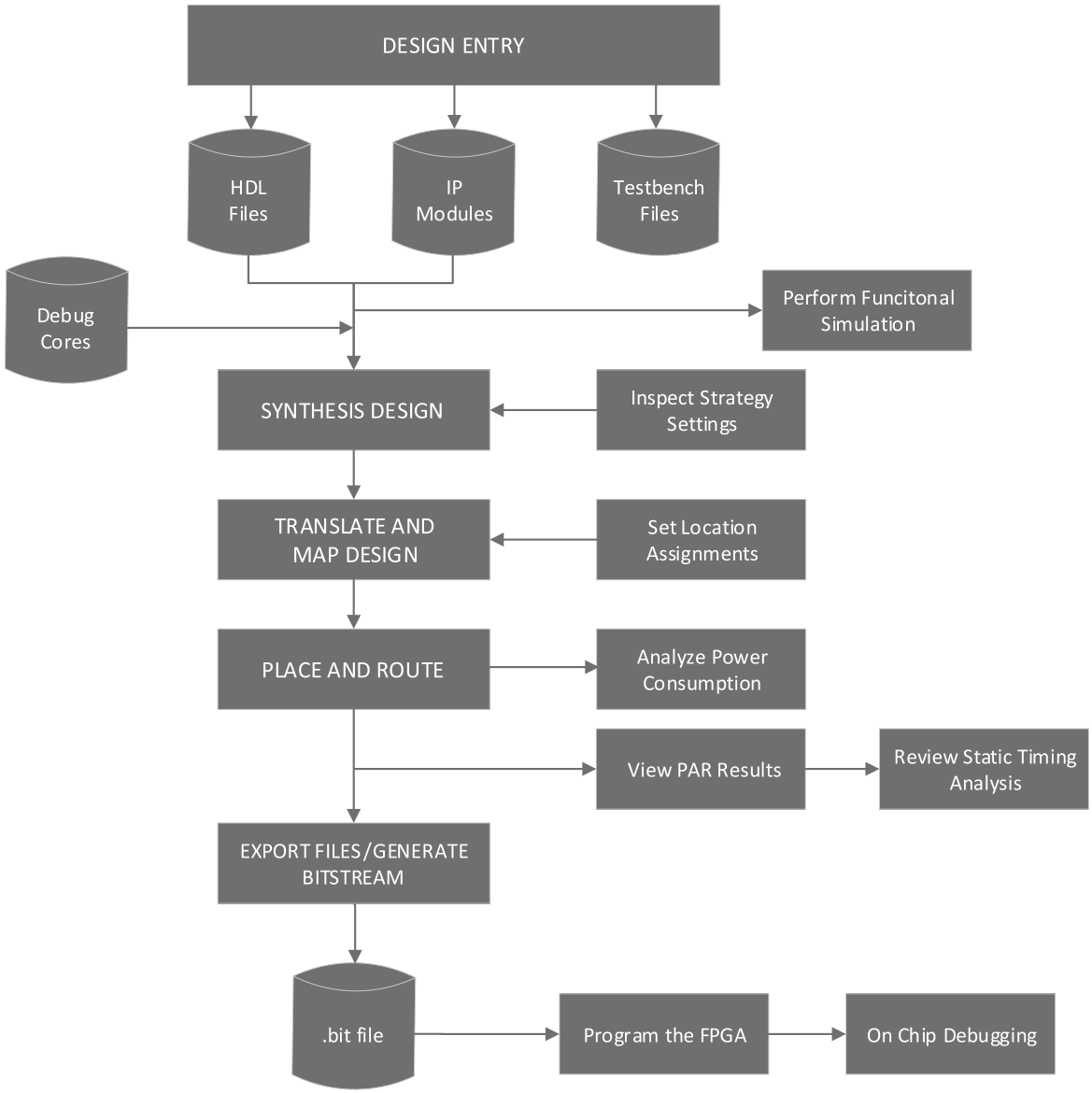


Figure 3.2. Lattice Radiant Software Detailed Design Flow

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4. FPGA Design Flow Tools and Views

This section shows how Lattice Radiant and Xilinx Vivado software handle the FPGA design flow steps with various tools and views.

4.1. Project Creation and Management

In setting up the initial design environment, both Xilinx Vivado and Lattice Radiant software provides two ways of creating a new design – batch mode or GUI mode. To start the software in each mode, see Table 4.1:

Table 4.1. Batch Mode and GUI Mode Comparison

Mode	Xilinx Vivado		Lattice Radiant	
	Windows OS	Linux OS	Windows OS	Linux OS
Batch Mode/TCL Scripting	In the command line, use the command: *vivado -mode batch -source <user_TCL_script>*	In the terminal, use the command: *vivado -mode batch -source <user_TCL_script>*	Use the Radiant TCL console tool included in Radiant Software Package or Setup the Environment to Run the software in Command Line	Radiant software provides a similar standalone TCL Console window (radiantc) that sets the environment or Setup the Environment to Run the software in Terminal
GUI	Launch the software through software icon/executable file or run command in command line: *Vivado or vivado -mod gui*	Run command in terminal: *Vivado or vivado -mod gui*	Launch the software through software icon/executable file	Run command in terminal: *<radiant install path>/radiant/<radiant version>/bin/lin64/radiant*

Note: Supported Windows and Linux OS version depends on each software version, which can be found in software release notes.

Upon launching the GUI mode of each software,similar to Xilinx Vivado's Quick Start function, Lattice Radiant provides a new project wizard functionality (Figure 4.1) that guides the users through the steps of project creation. The Lattice Radiant new project wizard allows the user to specify the project name and location (Figure 4.2), add existing source files (Figure 4.3), select the target device for the design (Figure 4.4) and choose the preferred Synthesis tool to be used, Lattice Synthesis Engine (LSE) or Synplify Pro (Figure 4.5).

Note: All project settings that the user selected are changeable at later stages in the design process.

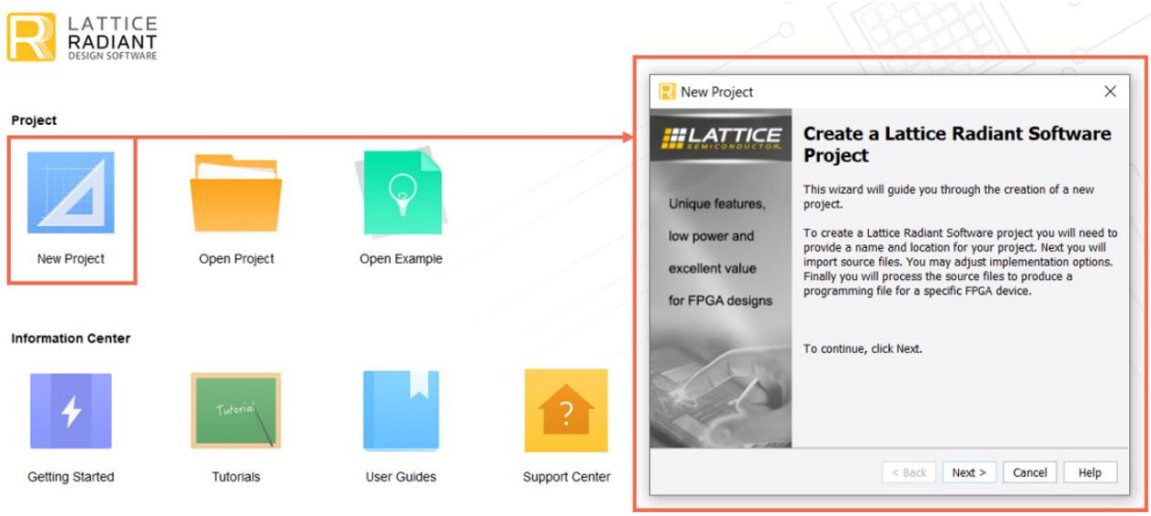


Figure 4.1. New Project Wizard

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

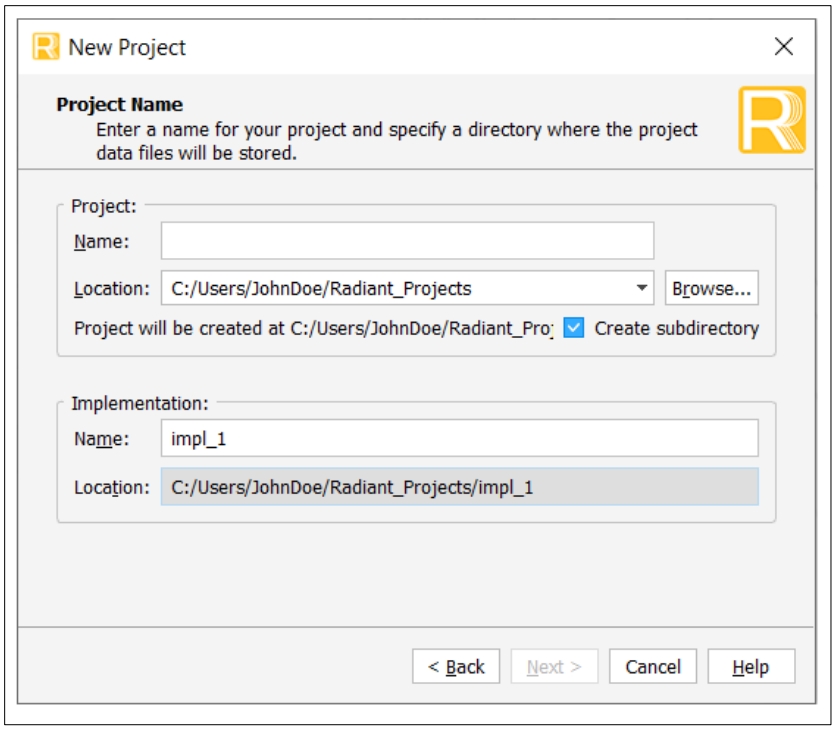


Figure 4.2. Project Settings

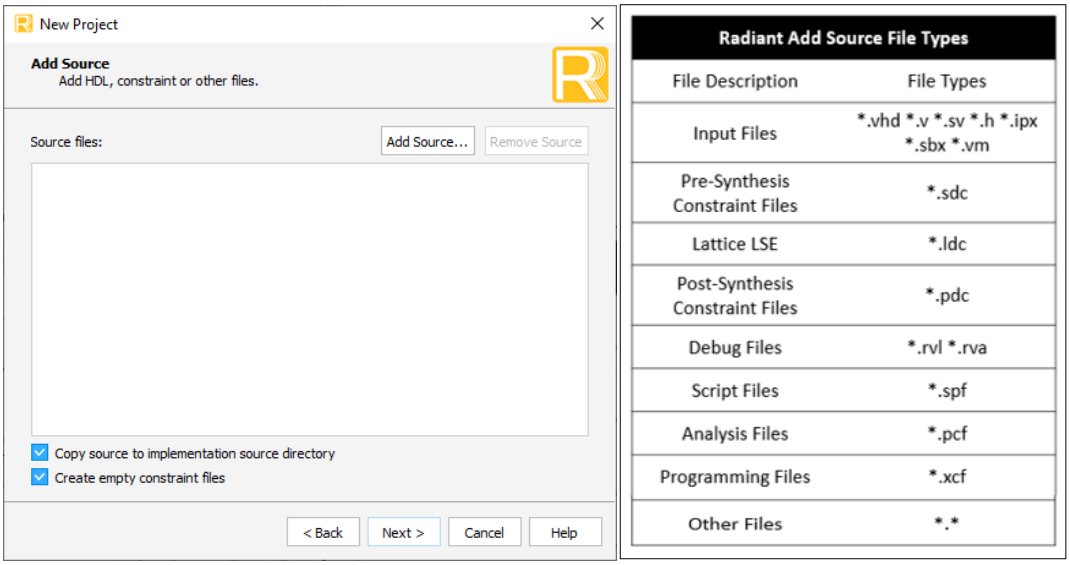
In the Project Name page, the following items are available.

Under the Project group:

- Name – Name of project
- Location – File path where the project is saved
- Create subdirectory – Enables the software to create a subfolder where the project is saved

The following items are available under the Implementation group:

- Name – Implementation name
- Location – File path where the implementation is saved



Radiant Add Source File Types	
File Description	File Types
Input Files	*.vhd *.v *.sv *.h *.ipx *.sbx *.vm
Pre-Synthesis Constraint Files	*.sdc
Lattice LSE	*.ldc
Post-Synthesis Constraint Files	*.pdc
Debug Files	*.rvl *.rva
Script Files	*.spf
Analysis Files	*.pcf
Programming Files	*.xcf
Other Files	*.*

Figure 4.3. Adding Existing Source

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

In the Add Source page, the following items are available.

- Add Source – Adds HDL files, constraint files or testbench in the current project.
- Copy source to implementation source directory – Copies the source files added in the current project directory.
- Create empty constraint files – Automatically generates a blank Pre Synthesis (.sdc) and Post Synthesis (.pdc) constraint files.

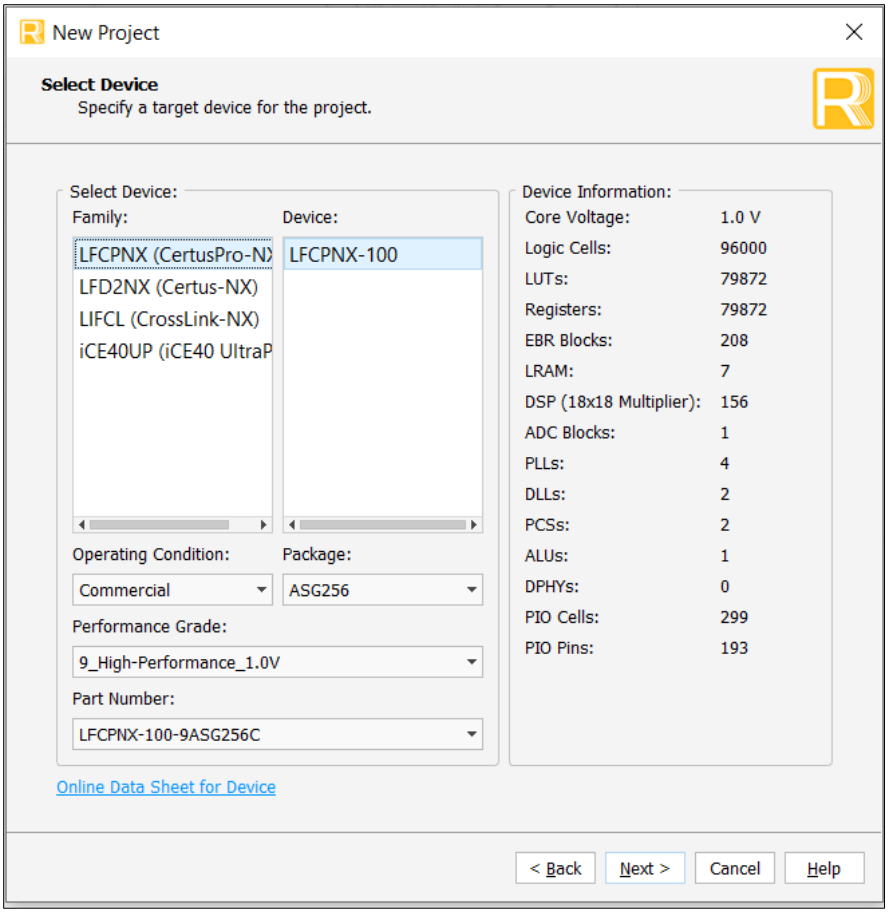


Figure 4.4. Target Device Selection

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

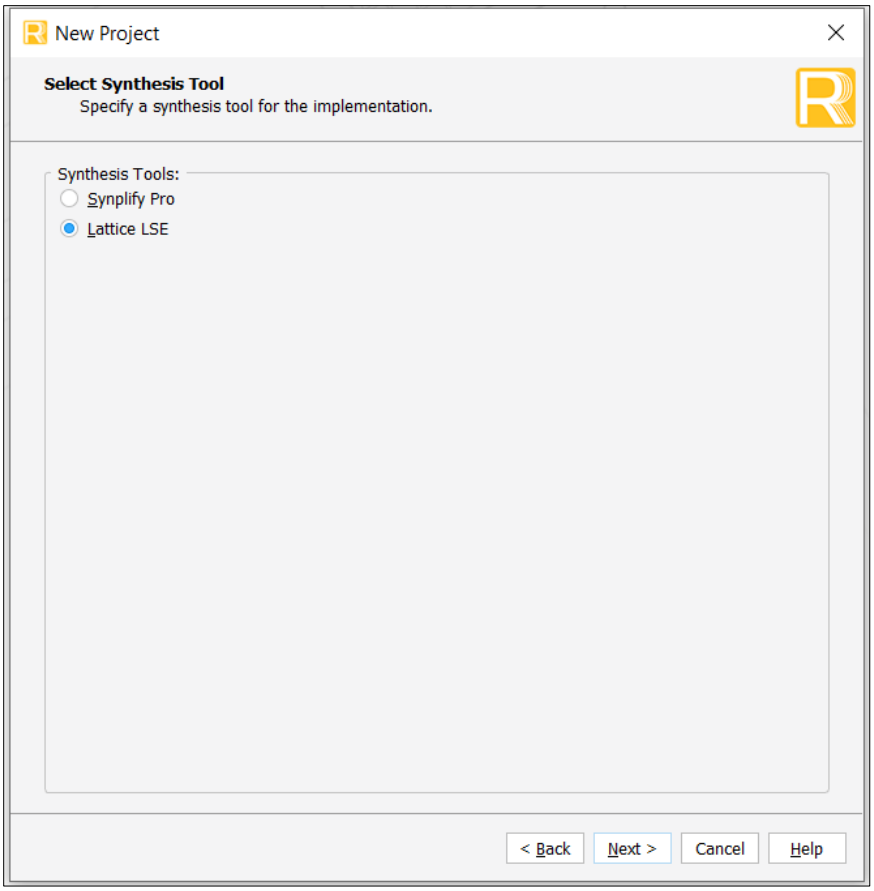


Figure 4.5. Synthesis Tool Selection

Once the project has been created, choosing the logical and consistent way of organizing each source file allows the users to easily locate and modify them as needed. Both Vivado and the Lattice Radiant supports this practice by combining design sources into different categories and list them in a Files/Files List View. Similar to Xilinx Vivado's Project Manager, the Lattice Radiant software helps users to keep the source files classified and listed under below category folders:

- Strategies
- Input Files
- Synthesis Constraint Files
- Debug Files
- Script Files
- Analysis Files
- Programming Files

Refer to Table 4.2 for the file types and extensions.

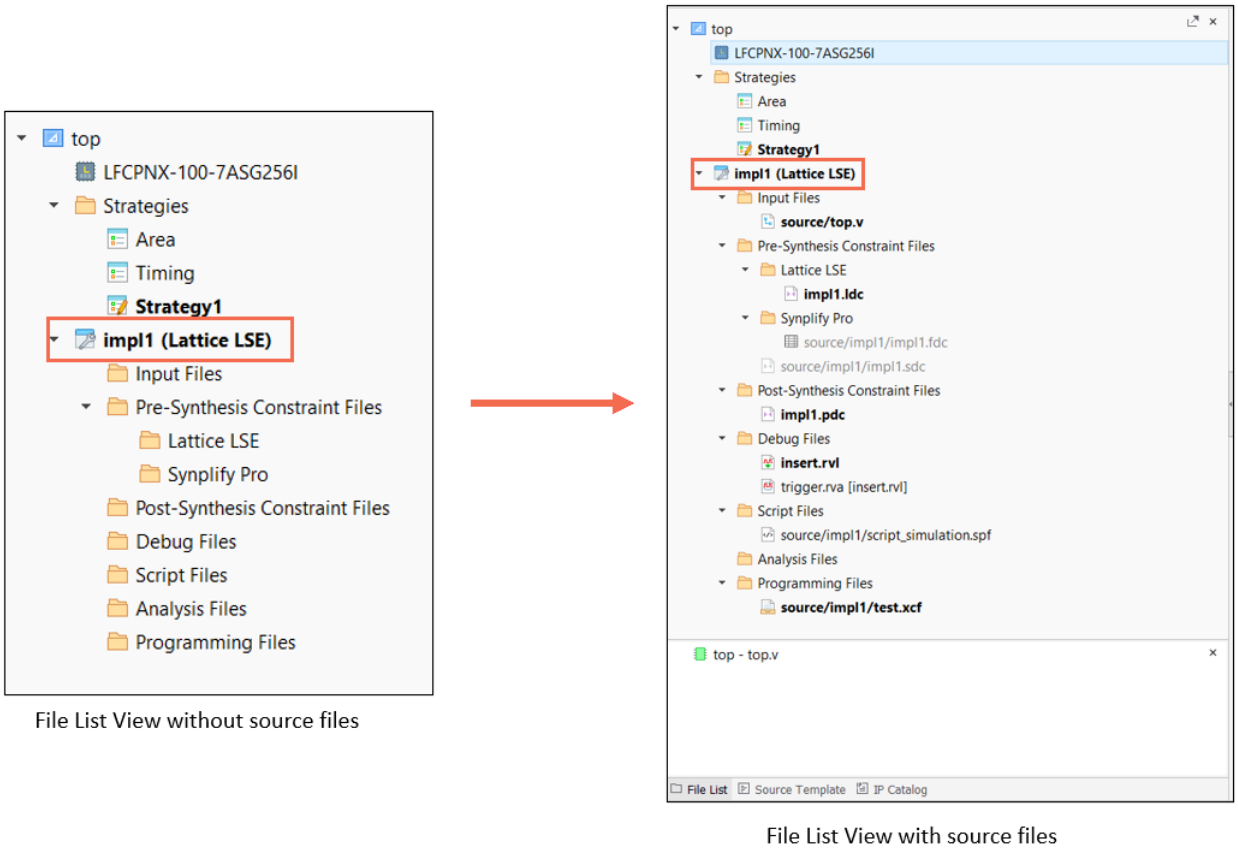
Table 4.2. Files Listed in Lattice Radiant Software File List View

File Type	File Extension
Project Title	None
Target Device	None
Predefined Strategy (Area, I/O Assistant, Quick, and Timing)	.sty
Strategy1 (that can be customized)	.sty
Implementation	None
Verilog Files	.v, .veri, .ver, .vo, .h
VHDL files	.vhd, .vhdl
Structural Verilog	.vm

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

File Type	File Extension
SystemVerilog	.sv
IP Module Config Files	.ipx
Undefined or incorrect	Any source reference
Synthesis Constraint Files	.sdc, .fdc, .ldc
Reveal Project File	.rvl
Simulation Project File	.spf
Reveal Analyzer Files	.rva
Power Calculator Files	.pcf
Programmer Project File	.xcf

User can also see implementations listed in the File List View, as shown in Figure 4.6.



File List View without source files

File List View with source files

Figure 4.6. Lattice Radiant Software File List View

The Lattice Radiant software also offers different ways of involving a source file in the design. If the user right-clicks on each source file, the user is given different options depending on the file type.

These options are:

- Excluding the file from implementation
- Setting the design file as top-level
- Removing the file for either or both synthesis and/or simulation
- Checking for properties

For implementation and strategy, user can create a different version or a clone of each, edit the properties, select the synthesis tool, and select the top level unit.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4.2. Design Strategy

In Xilinx Vivado, user can modify, optimize the design or resolve synthesis and place-and-route challenges using a set of pre-configured command line options. This is called a run strategy. Xilinx Vivado IDE provides several commonly used strategies that are tested against internal benchmarks.

In the Lattice Radiant software, this is equivalent to the design strategy. The Lattice Radiant software allows the user to choose or create its own custom strategy for implementation. A strategy is a collection of implementation-related tool settings or recipes for how the design is implemented.

The Lattice Radiant software provides two pre-defined strategies:

- Area – This strategy is used for area optimization. Its purpose is to minimize the total logic gates used while enabling the tight packing option available in Map.
- Timing– This strategy is used for timing optimization. Its purpose is to achieve timing closure.

It also enables the user to create customized strategies, which can be edited, cloned, and removed. All strategies are available to all of the implementations, and any strategy can be set as the active one for an implementation.

The user can view the strategies on the File List View. Double-click on the strategy to tool settings for each of the steps of the design flow.

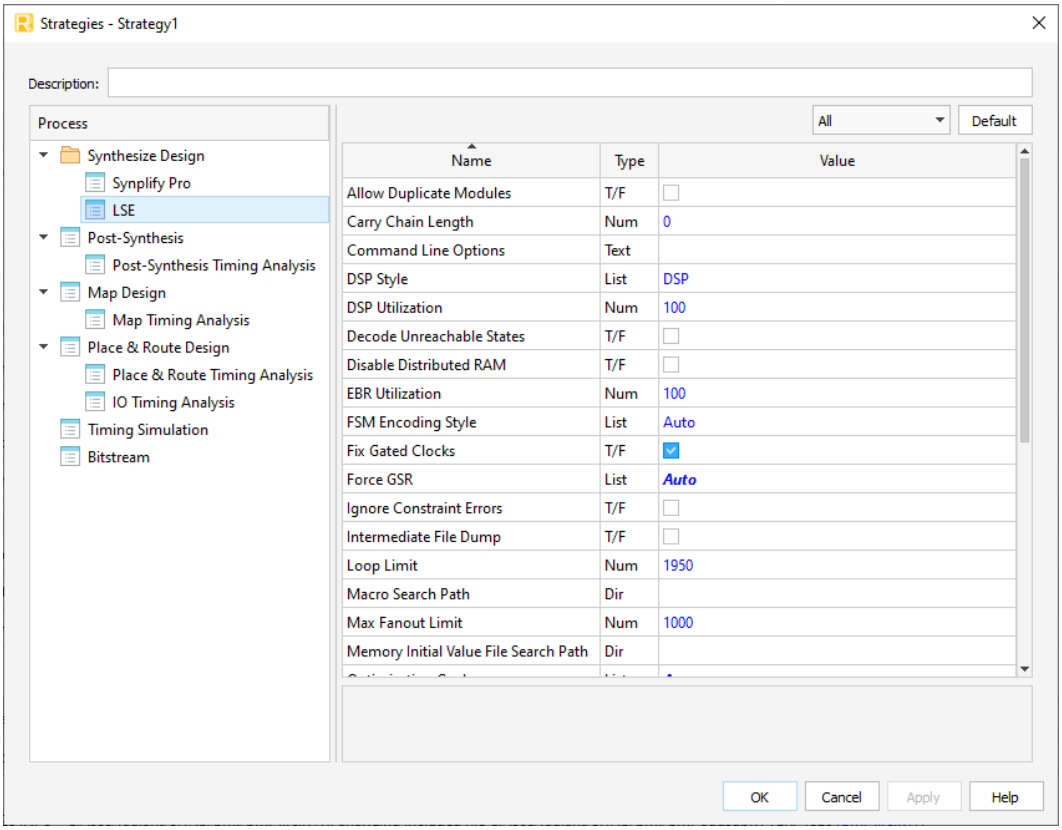


Figure 4.7. Area Pre-defined Strategy Settings

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4.3. Design Entry

The Lattice Radiant software supports HDL source files and IP catalog as design entry methods.

4.3.1. HDL Source Files

In Xilinx Vivado, the user creates a new HDL source file by clicking on Add Source under Project Manager. An Add Source window opens from which the user can choose to Add or create design sources. Similarly, in the Lattice Radiant software, clicking File > New opens the New File dialog box where user can choose which HDL file type to add to the design, as shown in Figure 4.8.

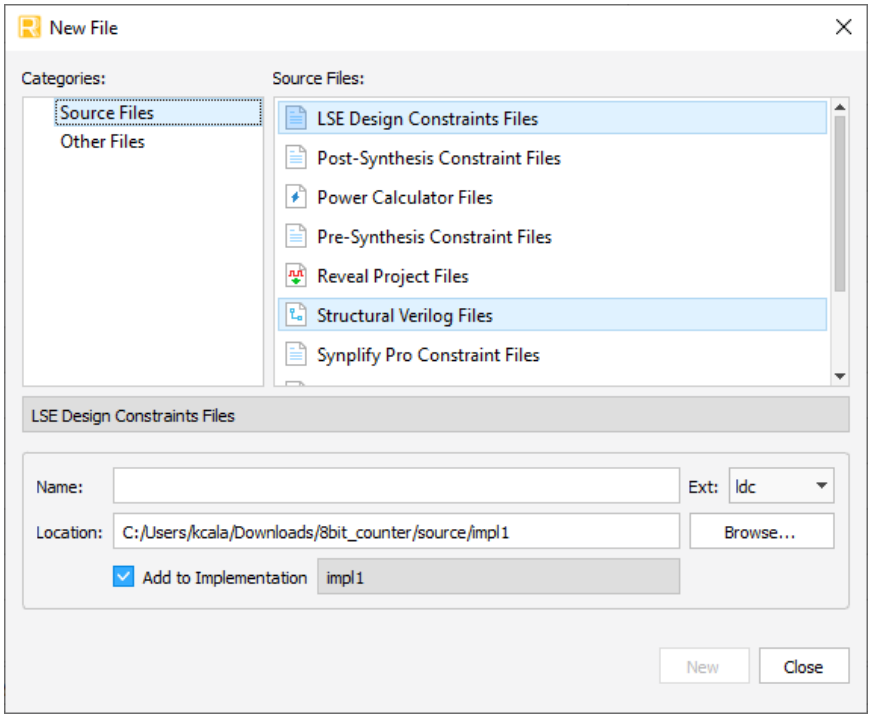


Figure 4.8. New File Dialog Box

Xilinx Vivado has templates that user can insert on the design by clicking on Language Templates under Project Manager. These templates are sample Verilog or VHDL templates, primitive instantiations, and timing and physical constraints templates. In the Lattice Radiant software, user can access these templates by clicking the Source Template tab.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

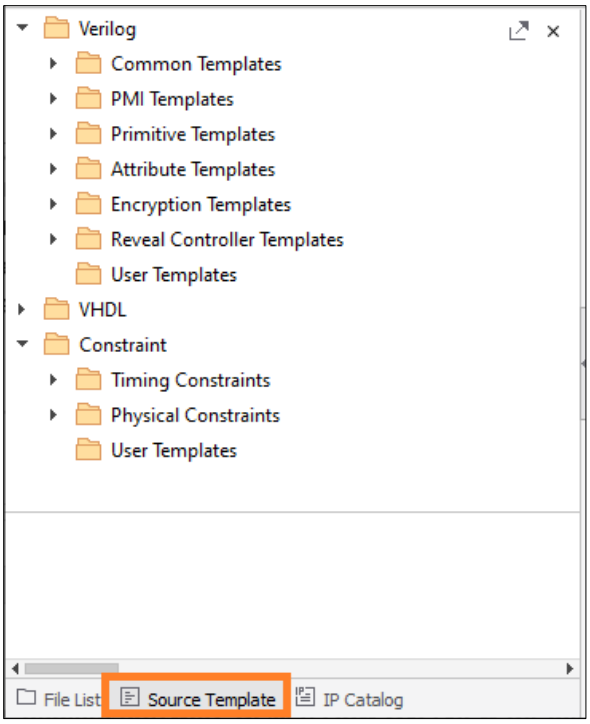


Figure 4.9. Source Template Tab

4.3.2. IP Catalog

In Xilinx Vivado, user can open the IP catalog by clicking on the IP Catalog under Project Manager. The IP catalog pane opens from where user can select which IP cores to customize and add to the design. In Lattice Radiant software, the IP catalog is accessed by clicking either the IP Catalog icon on the toolbar or the IP Catalog tab.

On the IP Catalog tab, user can select installed IPs on local folder or download and install additional IPs from the IP Server.

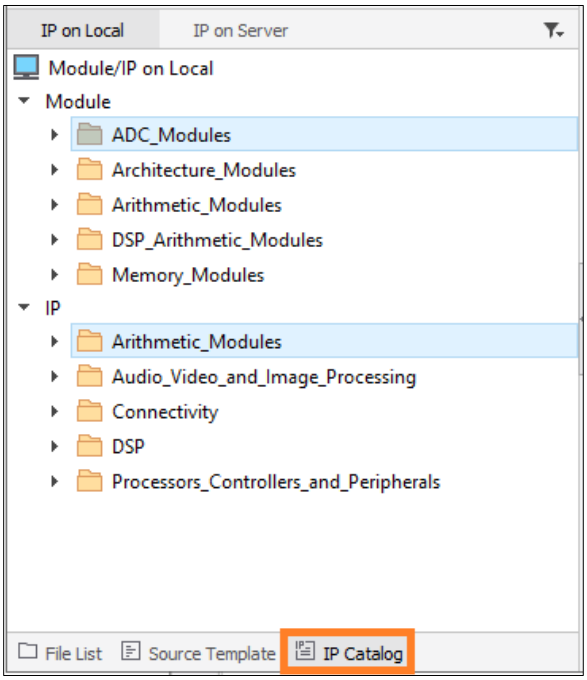


Figure 4.10. IP Catalog Tab

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

For more information, refer to the Lattice Radiant Help or the following documents:

- Lattice Radiant Software IP User Guide
- Lattice Radiant Software 3.1 User Guide sections:
  - IP Catalog
  - Packaging IP Using IP Packager
  - Running Radiant IP Packager and Viewing Documentation
  - Installing IP Created with IP Packager into IP Catalog

4.4. Design Constraints

Design constraints are used to specify the performance requirements desired for the FPGA design. Various tools helps the designers to meet those conditions. Constraints are instructions applied to the design elements that guide the design toward desired results and performance goals. They are critical to achieving timing closure or managing reusable intellectual property (IP). The most common constraints are those for timing and pin assignments, but constraints are also available for placement, routing, and many other functions.

Table 4.3. Design Tools Comparison

Xilinx Vivado Design	Lattice Radiant
Device, Physical and Timing Constraints window	Device, Physical and Timing Constraint Editor

The Lattice Radiant software provides tools similar to Xilinx Vivado with GUI for assigning constraints: Device/Physical/Timing Constraint Editor.

Xilinx Vivado stores all the design constraints and attributes in one single file, (.xdc) Xilinx Design Constraint, which includes all timing and device constraints.
In the Lattice Radiant software, constraints can be defined in different constraint files (.ldc/.pdc/.sdc/.fdc) or HDL attributes (for physical pin locking) depending on the synthesis tool being used and constraints entry points.
.ldc and .sdc are used for the synthesis tool and specify design goals. Synthesis, map, and place-and-route work to meet these goals. Post-synthesis constraints (.pdc) can also be specified. The flow combines these based on different entry points within the Radiant software design flow. The timing analysis tool reports whether or not the goals were met.

The pre-synthesis constraint creation flow depends on synthesis tool selection:

- .ldc (through LSE)
- .sdc (through Synplify Pro and LSE)
- .fdc (through Synplify Pro)

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

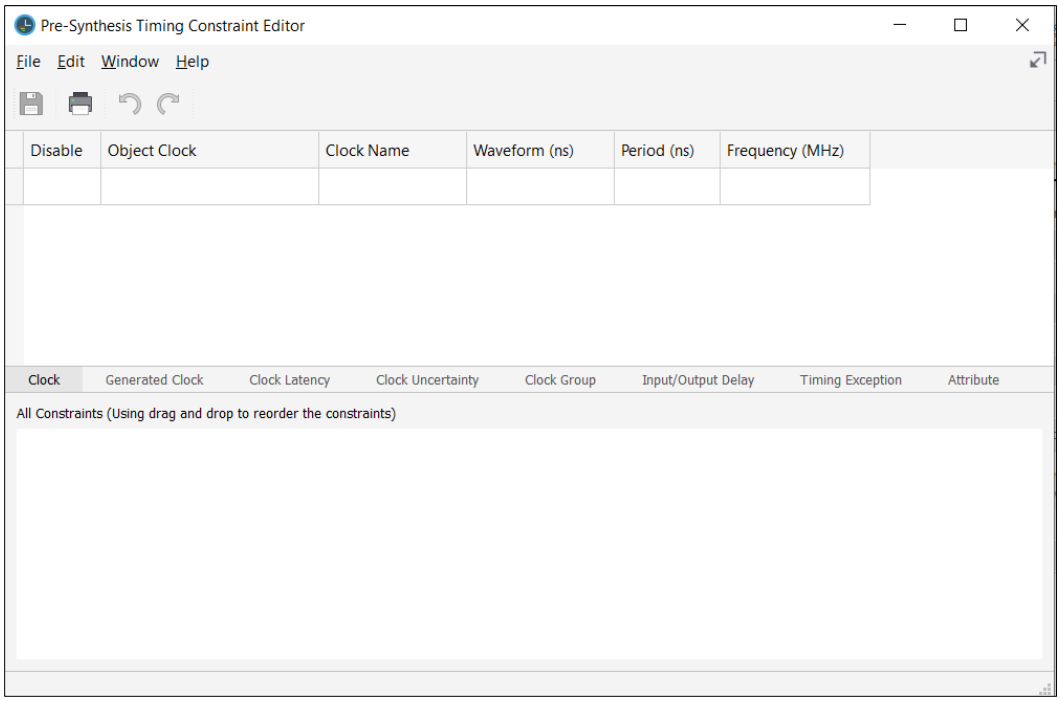


Figure 4.11. Pre-Synthesis Timing Constraint Editor Window

The post-synthesis constraint flow is the same for all projects, and should be done before MAP and PAR. Post-synthesis constraints are saved in the .pdc file.

General Timing Constraint Flow can be seen in Figure 4.13.

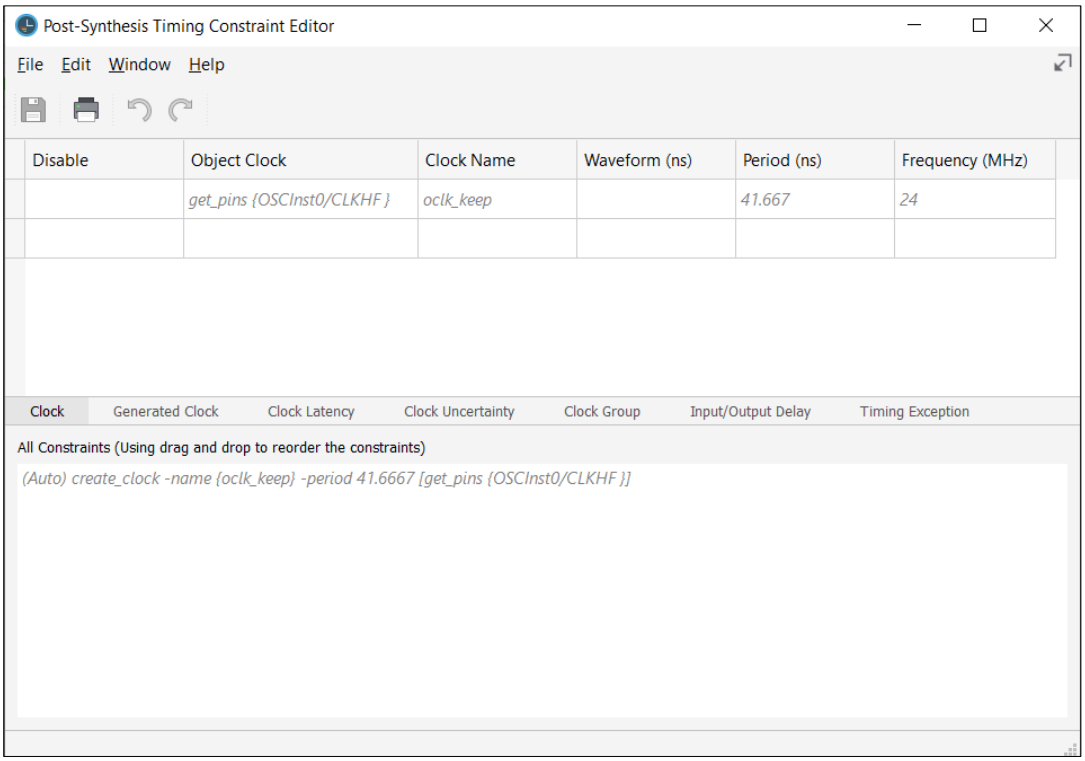


Figure 4.12. Post-Synthesis Timing Constraint Editor Window

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

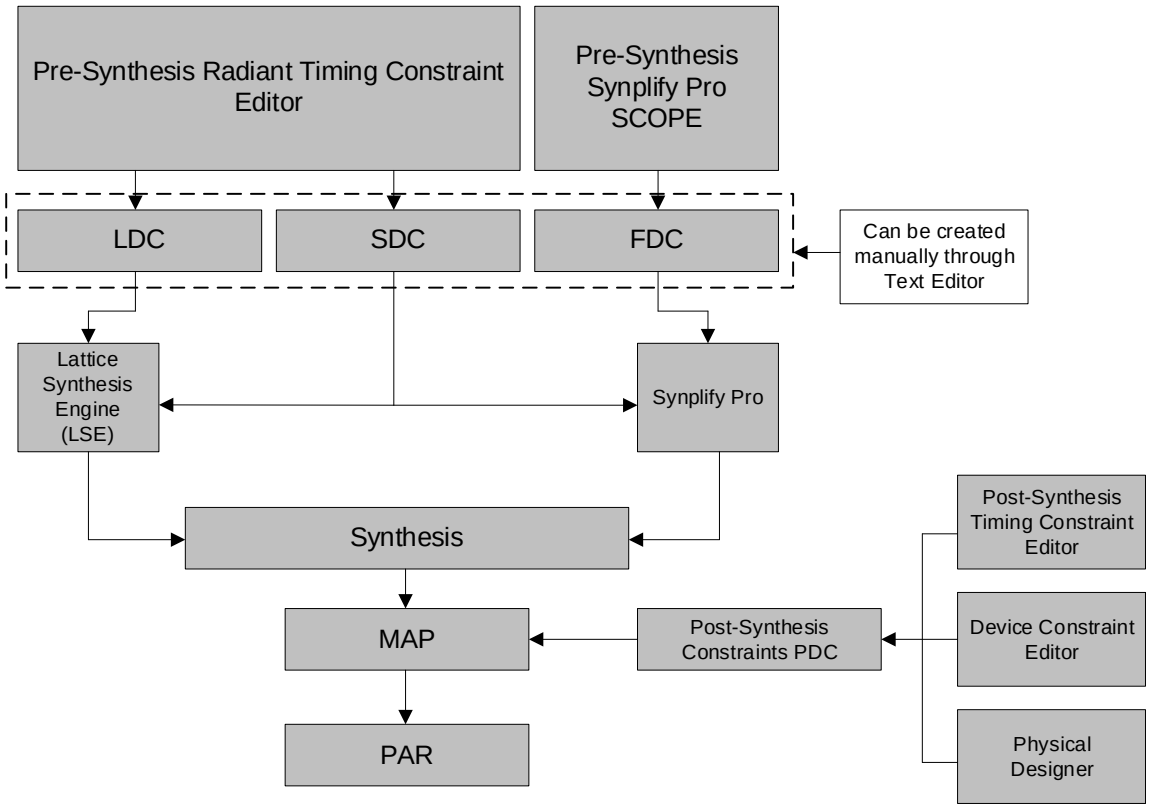


Figure 4.13. General Constraint Flow

Xilinx Vivado provides different windows for instance-specific settings and constraints like I/O Ports Window, Physical, and Timing Window, while the Lattice Radiant Device Editor showcases all the pin layout of the device and displays the assignments of signals to device pins. This view allows the user to edit these assignments, reserve sites on the layout to exclude from placement and routing. It is also the entry mechanism for physical constraints which are saved in the .pdc file. The Device Constraint Editor views enables the user to develop constraints that shortens the turn-around time and achieve a design that conforms to critical circuit performance requirements.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

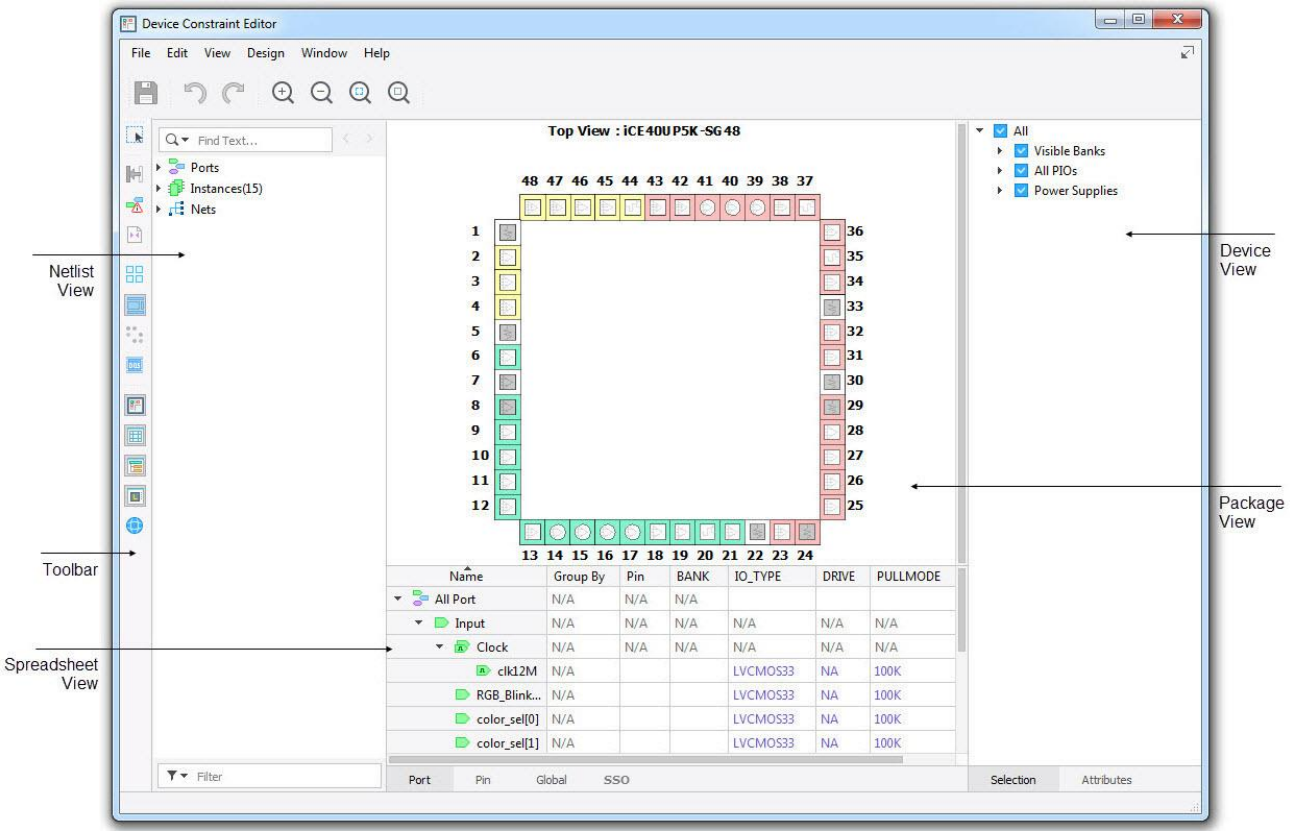


Figure 4.14. Device Constraints Editor

4.5. Synthesis

Xilinx Vivado has the Vivado Design Suite Synthesis to perform synthesis on designs with user logic and Xilinx IP. Lattice Radiant software has a proprietary synthesis tool in Lattice Synthesis Engine (LSE) and supports a third party synthesis tool in Synplify Pro. Both synthesis tools supports synthesis of user logic and Lattice IP.

Table 4.4. Synthesis Tools Comparison

Xilinx Vivado	Lattice Radiant Software
Vivado Design Suite Synthesis	LSE and Synplify Pro

User can select synthesis tools in Vivado by clicking Analysis and Synthesis and selecting Edit Settings > Electronic Design Automation (EDA) Tool Settings. In the Lattice Radiant software, user can do this by right-clicking the current implementation and then choosing Select Synthesis Tool from the drop-down menu.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

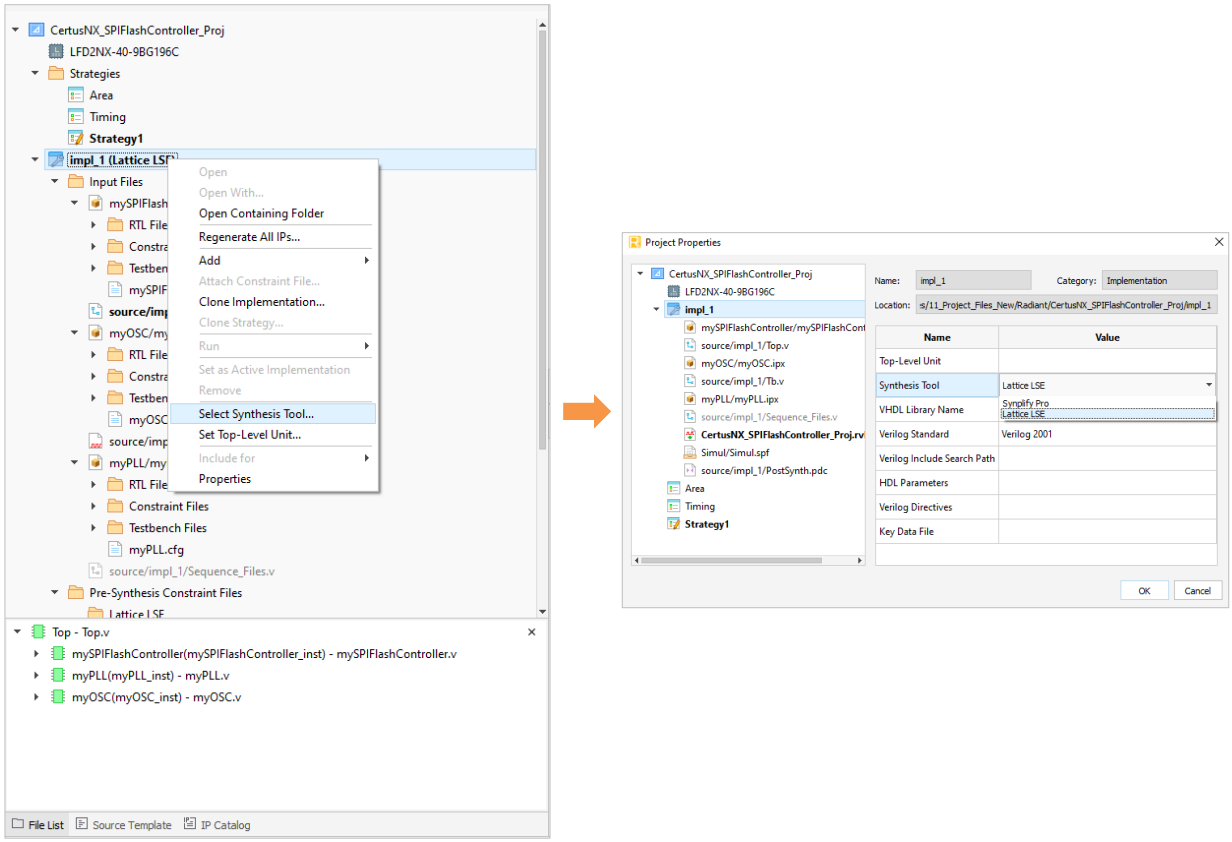


Figure 4.15. Selecting Synthesis Tools

In Xilinx Vivado, synthesis is performed by clicking on Run Synthesis under Synthesis of the Flow Navigator pane. In the Lattice Radiant software, synthesis is performed by clicking on the Synthesis Design icon in the Process toolbar. The Synthesis icon changes to a check icon when Synthesis is successfully performed.

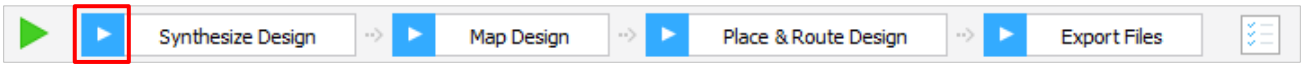


Figure 4.16. Running Synthesis on the Process Toolbar

User can also click on the Run All button to perform the Synthesis up to Export Files. The Run All button changes to a Stop button and clicking this stops the flow.

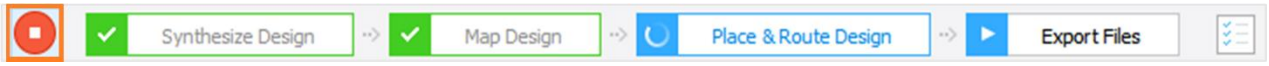


Figure 4.17. Stopping the flow on the Process Toolbar

To modify the optimization and other settings related to synthesis, user can refer to the Design Strategy section.

4.6. Netlist Viewer

A generated netlist can be viewed through one or more schematic views and a browser that shows the lists of modules, instances, ports, and nets. In Xilinx Vivado, this can be performed going to Synthesis > Open Synthesized Design > Schematic on the Flow Navigator pane. While in the Lattice Radiant software, this can be performed using Netlist Analyzer of LSE, or the HDL-Analyst on Synplify Pro.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 4.5. Tools Comparison

Xilinx Vivado	Lattice Radiant Software
RTL Viewer and Technology Map Viewer	Netlist Analyzer HDL-Analyst on Synplify Pro

To open Netlist Analyzer, click the Netlist Analyzer icon on the toolbar.

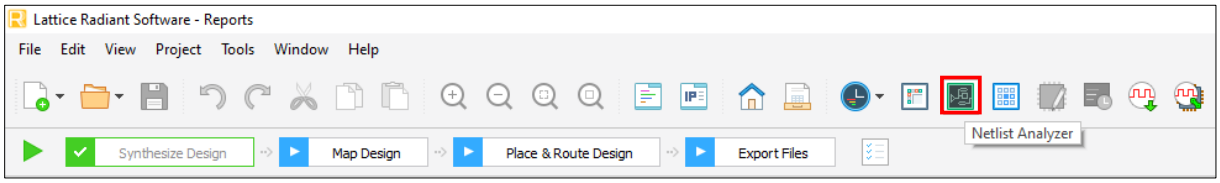


Figure 4.18. Opening Netlist Analyzer

To open HDL-Analyst, open Synplify Pro from Lattice Radiant software and then click on the Technology View icon on the toolbar.

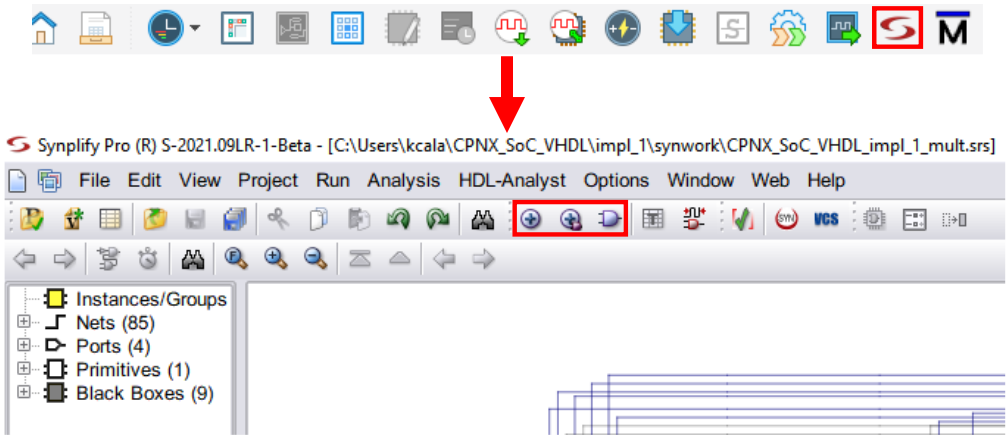


Figure 4.19. Opening HDL-Analyst Views

The RTL view provides a high-level, technology-independent, graphic representation of the design after compilation, using technology-independent components like variable-width adders, registers, large multiplexers, and state machines. While the technology view provides a low-level, technology-specific view of the design after mapping, using components such as look-up tables, cascade and carry chains, multiplexers, and flip-flops.

For more information, refer to Lattice Radiant Software Help or to Lattice Radiant Software 3.1 User Guide Netlist Analyzer section.

4.7. Design Mapping

Design mapping converts the logical design into a network of physical components or configurable logic blocks. In Xilinx Vivado, this process is combined with the Implementation process. In Lattice Radiant software, this is a separate process in the design flow that can be optimized through the design strategy settings.

To map a design in Lattice Radiant, click the Map Design icon.

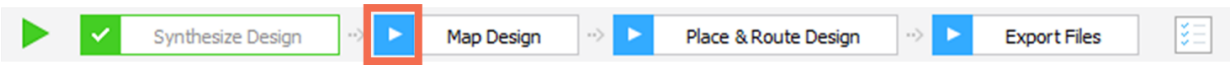


Figure 4.20. Running Map on the Process Toolbar

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4.8. Place and Route

In Xilinx Vivado, the translation to bring the physical design format in Mapping and passed it on the Placement and Routing phase are all in the Implementation stage of the software. In Lattice Radiant, Map and Place and Route are in two separate process.

To run Place and Route, click the Place and Route Design icon.

Figure 4.21. Running Place and Route on the Process Toolbar

Placement is the process of assigning the device-specific components produced by the MAP process to specific locations on the device floorplan. After placement is complete, the route phase establishes physical connections to join components in an electrical network. The place and route process takes a mapped physical design and places and routes the design. Placement and routing of a design can be cost-based or timing driven.

In Xilinx Vivado software, the implementation flow places and routes the netlist onto the FPGA device resources based on the constraints of the design and processing of optimizing, placing, and routing the primitives of a synthesized netlist

Table 4.6. Place and Route Tool Comparison

Xilinx Vivado	Lattice Radiant
Implementation	Place and Route

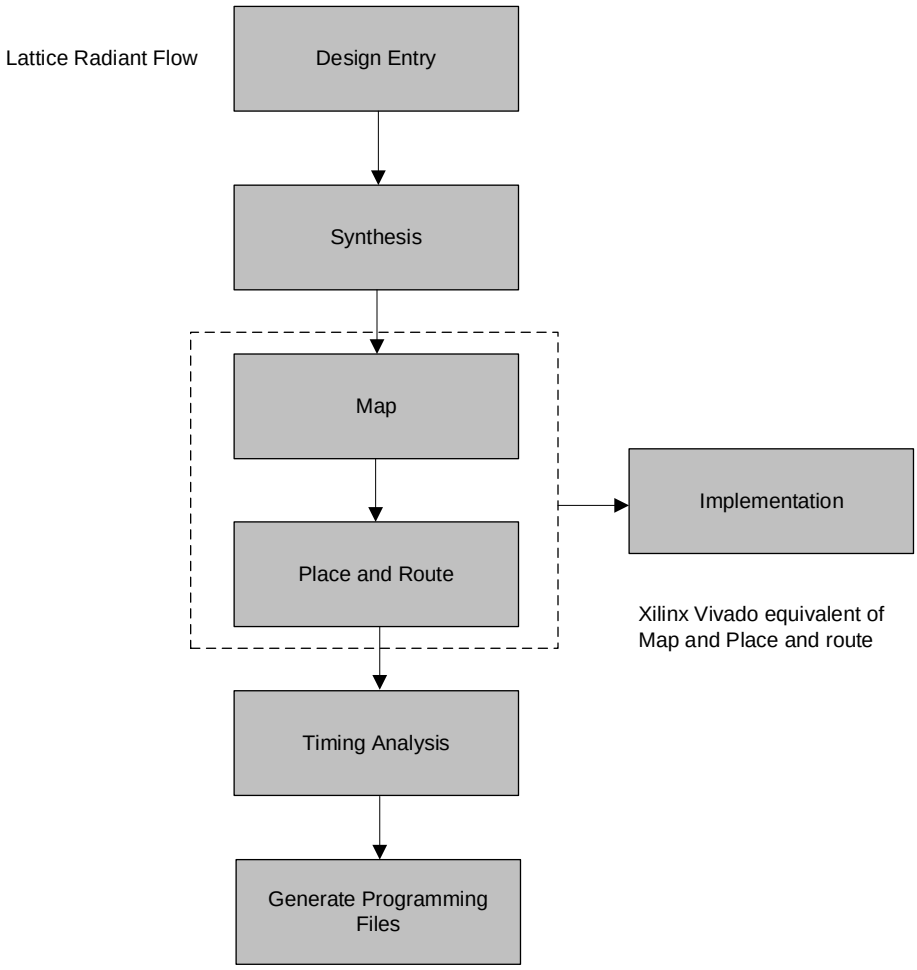


Figure 4.22. Lattice Radiant and Xilinx Vivado Flow Comparison

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Placement and routing options can influence the performance and utilization of the design implementation and ease incremental design changes. Some options affect the way the results are reported. Experimenting with place and route settings in the Strategies dialog box can help improve the placement and routing results.

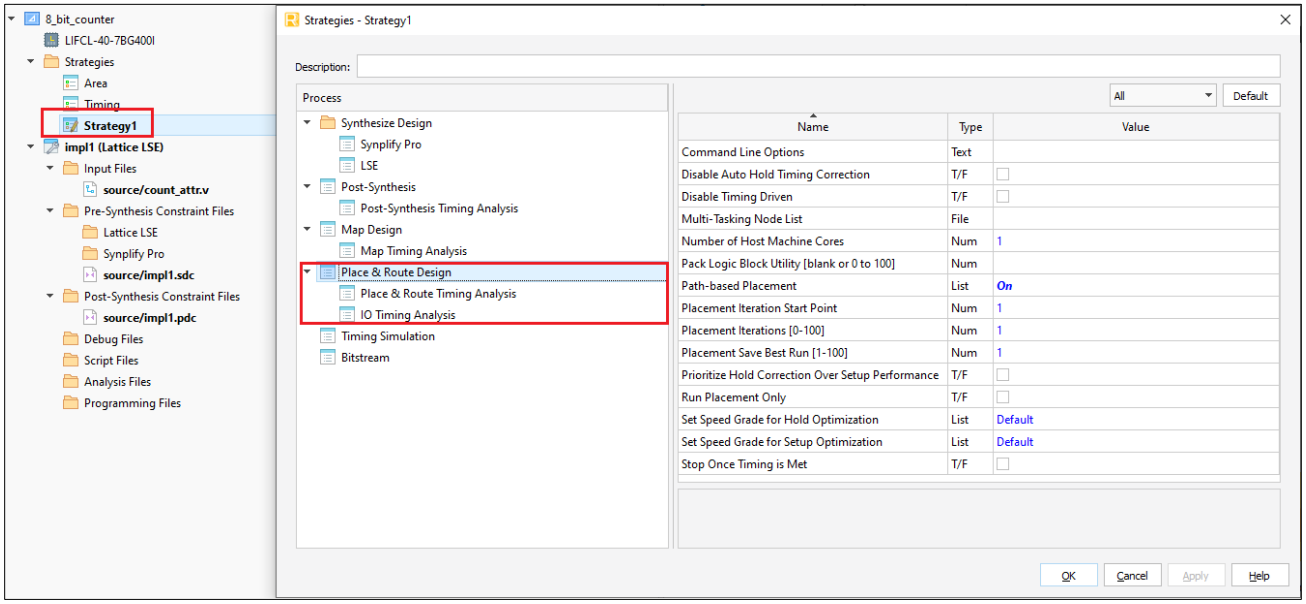


Figure 4.23. PAR Design Strategy Options

In most cases, the design requires timing-driven placement and routing, where the timing criteria the user specifies influence the implementation of the design. In the Lattice Radiant software, the static timing analysis results shows how constrained nets meet or do not meet the timing. In the Radiant Task Detail View, there is a Place and Route Timing Analysis process available that runs static timing analysis. This process reports any timing errors and generates a report.

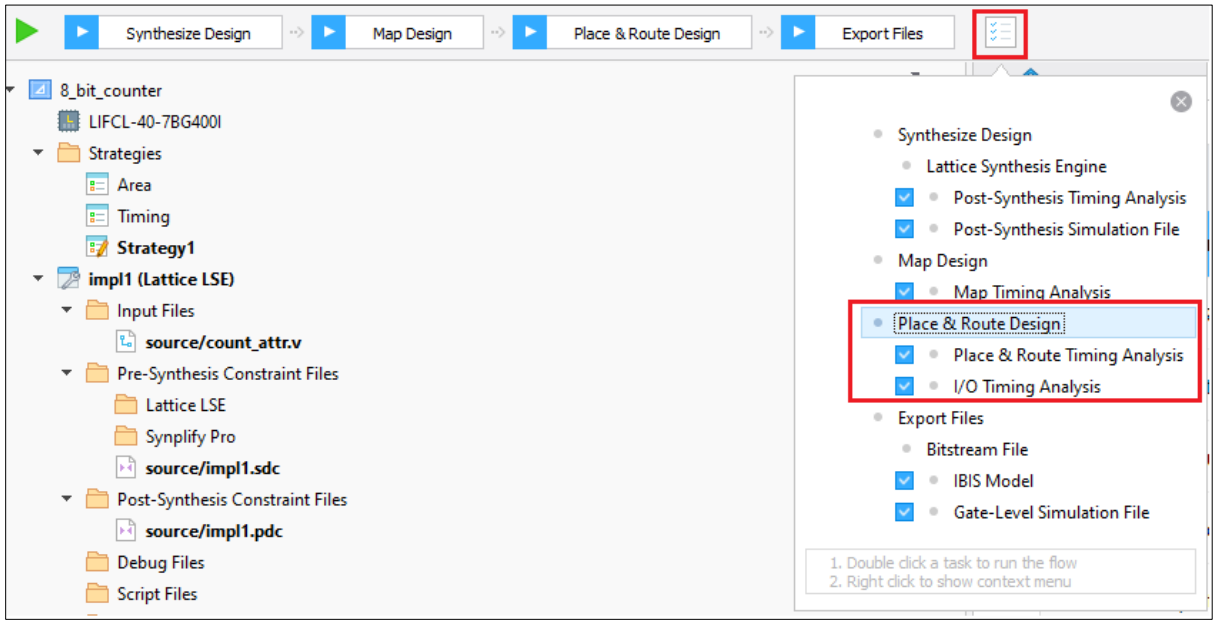


Figure 4.24. PAR Timing Analysis Option

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

After place and route or implementation, the Xilinx Vivado software has the Device and Package Window to allow editing of the post-Implementation design placement. In the Lattice Radiant software environment, the Physical Designer tool is the combined GUI interface of both the Placement Mode and Routing Mode accessible within this one Radiant software tool. This provides for one central location where a user can do all the floor-planning and be able to view the physical layout of the design. The tool has three modes option:

- Placement Mode
- Input/output (I/O) Mode
- Routing Mode

Placement Mode provides a large-component layout of the design. All connections are displayed as fly-lines. Placement Mode allows the user to create REGIONs and bounding boxes for GROUPS and specify the types of components and connections to be displayed. As the user hover the mouse over the floorplan layout, the details are displayed in tool tips: the number of resources for each GROUP and REGION; the number of utilized slices for each programmable logic controller (PLC) component; and the name and location of each component, port, net, and site.

Placement Mode is for GROUPS and REGIONs assignment. User can use the I/O Mode tool for I/O planning and I/O assignment such as Double Data Rate (DDR) interface, DQS and clock assignments. As the user move the mouse pointer slowly over the I/O Abstract layout, the details are displayed in tool tips: the number of resources for each items such as ECLCK, DDR, Phase Locked Loop (PLL), and I/O Banks utilization are displayed.

Routing Mode provides a read-only detailed layout of the design that includes switch boxes and physical wire connections. Routed connections are displayed as Manhattan-style lines, and unrouted connections are displayed as fly-lines. As the user move the mouse slowly over the layout, the name and location of each REGION, group, component, port, net, and site are displayed as tool tips.

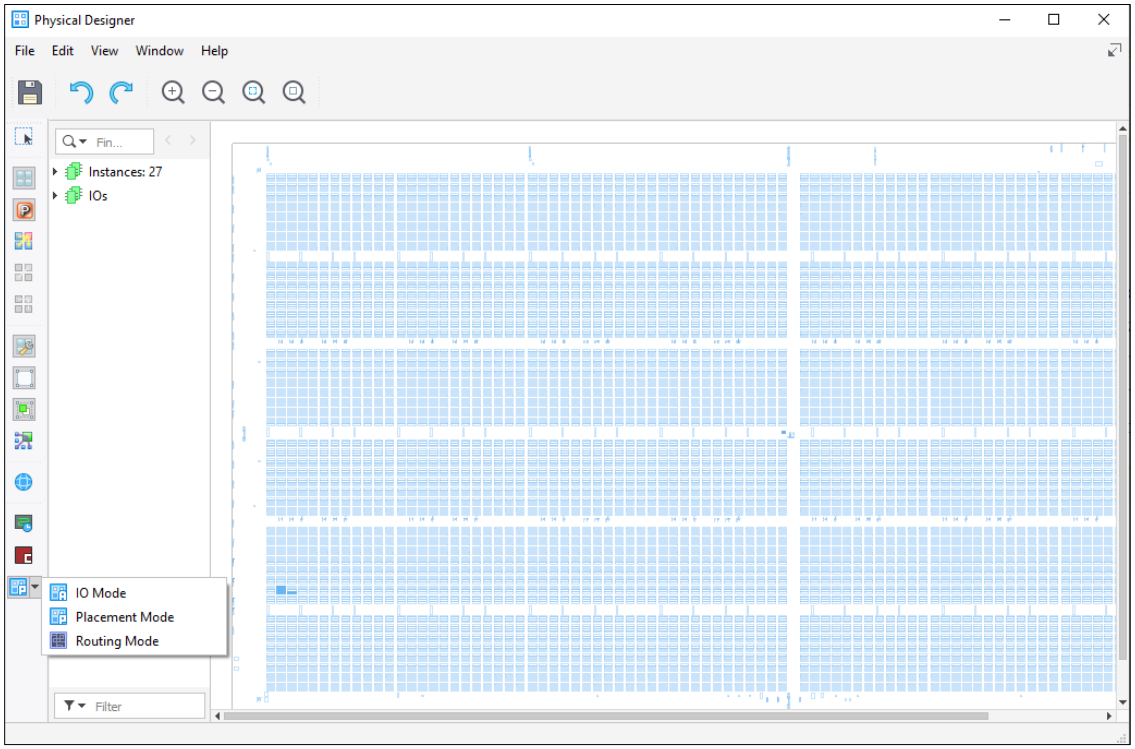


Figure 4.25. Physical Designer Placement Mode

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4.9. Cross-Probing

In Xilinx Vivado, if the user wants to check the design elements from one tool to another tool, user needs to jump over different windows. In the Lattice Radiant, there is a cross-probing feature where it allows the software for seamless bi-directional communication between the different tools within the software itself. Users can select a design element from one tool and locate them in another tool.

In the Lattice Radiant software, aside from the tools that can use cross-probing, users can also click a hyper-link icon in the reports to cross probe into that tool.

- Post-Synthesis and Map timing report links to Netlist Analyzer when using LSE as synthesis tool.
- In the PAR timing report, user can cross probe to Netlist Analyzer, Physical Designer Placement Mode and Routing Mode.

4.10. Programming Files

After the user created and verified the design, the user can use the final output data file to download or upload a bitstream to or from an FPGA device using the Radiant Programmer. Use the Process Toolbar to generate files for exporting – see Figure 4.26. Bitstream generation is automatically performed when the user runs the Export Files process for either a full or partial design flow.

Figure 4.26. Running Export Files on the Process Toolbar

Similar to Xilinx Vivado's Hardware Manager, the Lattice Radiant Programmer supports serial and microprocessor programming of Lattice devices in PC and Linux environments.

Table 4.7. File Formats

File Format	Description
Data File	A data file can be a hex, or bitstream file. Each of these files is based upon an IEEE programming standard:
Bitstream	Data files used for configuring volatile memory (SRAM) of our FPGAs.
Hex	Hexadecimal PROM data files used for Programming into external non-volatile memory, such as parallel or Serial Peripheral Interface (SPI) Flash devices.

For more information, refer to the Lattice Radiant software help or to the following sections in Lattice Radiant Software 3.1 User Guide:

- Programming Files
- Programmer

4.11.Run Manager

Large designs can quickly get complicated, and the best approach is not always obvious or definite. There may be many ways to get optimal performance out of the design—the key is to find one option that works.

In Xilinx Vivado, user can perform multiple runs of synthesis and/or implementation (which is the equivalent of MAP and PAR in the Lattice Radiant) with different run and report strategies by right-clicking on the Design Runs tab and then going to Create runs.

In the Lattice Radiant, this is similar to the Run Manager where user can run multiple implementations. Implementations are comprised of selected source files and only one strategy. User can create multiple implementations to bind different strategies.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

After creating implementations, use Run Manager to run multiple synthesis and place and route passes, compare the results of multiple implementations for further analysis to get best solutions. Run Manager helps the user to manage running the design implementation process with multiple project implementations (versions) and to compare the results. User can monitor progress, view reports, and quickly identify the best implementation.

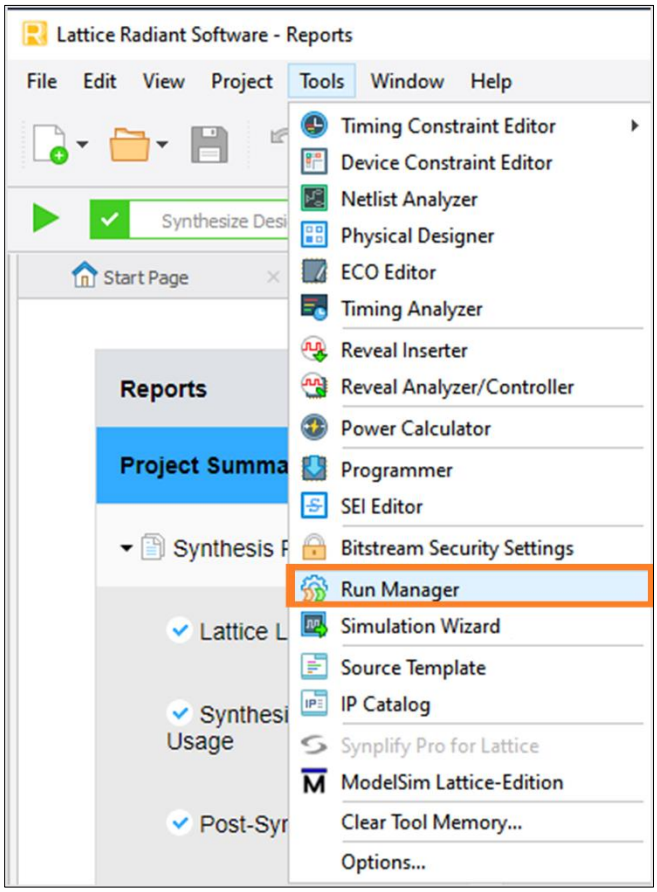


Figure 4.27. Opening the Run Manager

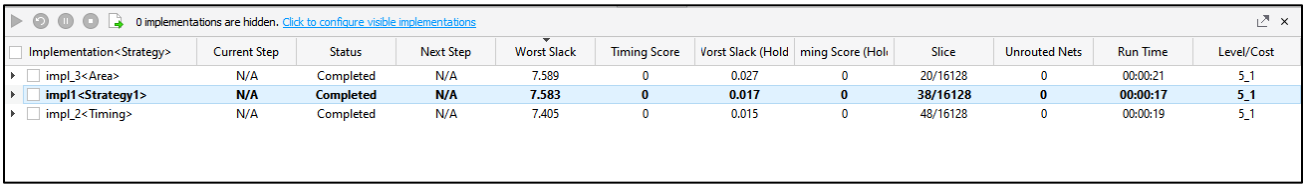


Figure 4.28. Run Manager Tab

4.12. Reports

In Xilinx Vivado, there are report strategies which defines the reports that are created during synthesis or implementation. User can check the report strategies by going to the Tool Settings > Strategies > Report Strategies on the Settings window. User can also generate reports for Timing Summary, Clock Networks, Clock Interaction, Methodology, DRC, Noise, Utilization, and Power on the Flow Navigator pane by going to Synthesis (or Implementation) > Open Synthesized Design (or Open Implemented Design). In the Lattice Radiant software, user can click the Reports tab, which is open by default on the workspace. User can also click View > Reports.

Each step on the design flow has its own set of reports from resource usage to timing analysis provided by the user-enabled timing analysis on the process toolbar.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

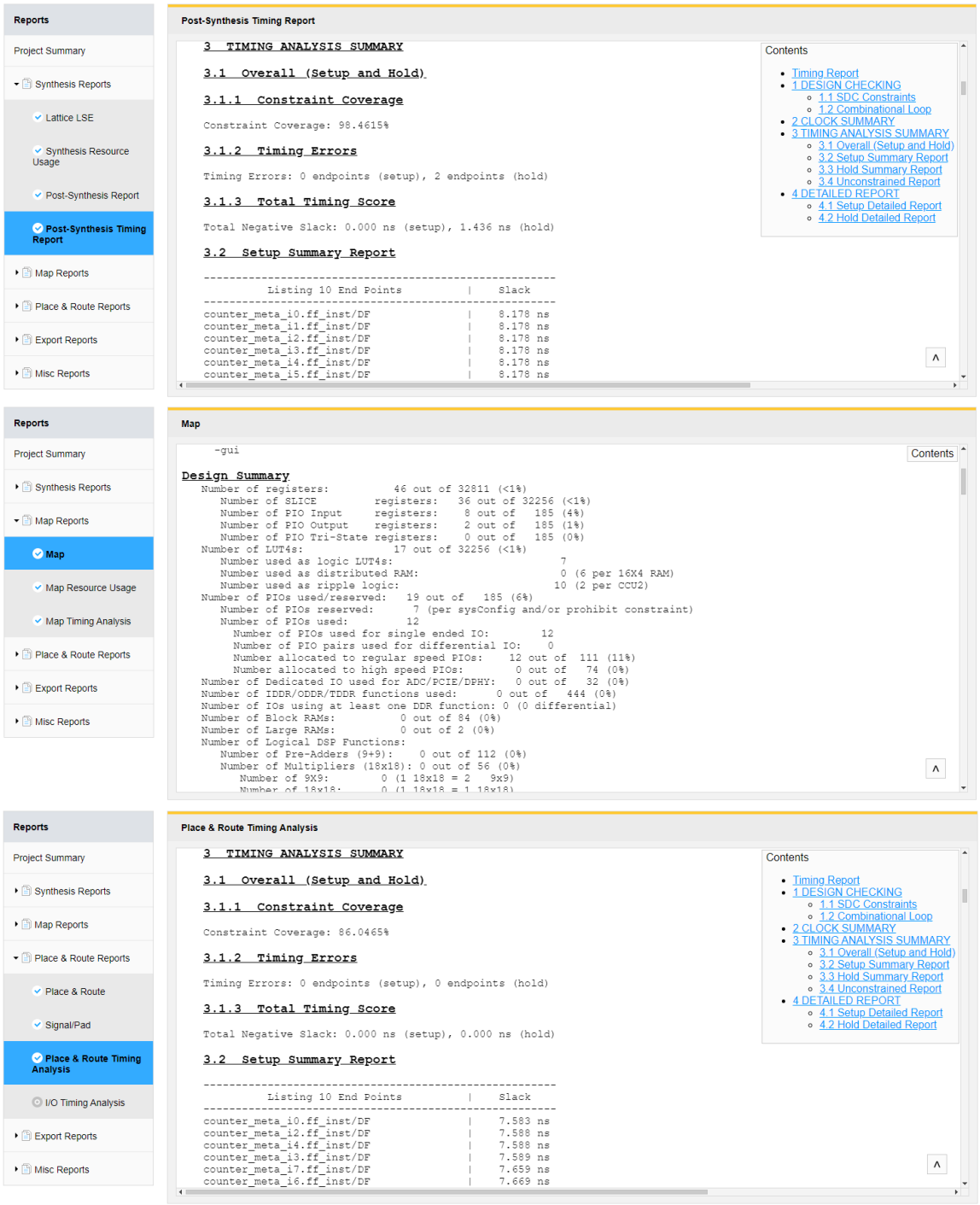


Figure 4.29. Reports Tab

For more information, refer to the Lattice Radiant software help or to the following sections in Lattice Radiant Software 3.1 User Guide:

- Reports and Messages Views
- Reports

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5. Design Verification Tools

5.1. Simulation

Both Xilinx Vivado and Lattice Radiant software support a number of third-party simulators, as shown in Table 5.1.

Table 5.1. Supported Simulators

Xilinx Vivado	Lattice Radiant
Vivado simulator, Mentor Graphics Questa Advanced Simulator, Mentor Graphics ModelSim Simulator, Cadence Incisive, Enterprise Simulator (IES), Synopsys Verilog Compiler Simulator (VCS), Aldec Rivera-PRO Simulator, Aldec Active-HDL, Cadence Xcelium Parallel Simulator	Cadence Xcelium, Synopsys VCS, Siemens Questasim, and Siemens ModelSim

In Xilinx Vivado, user can run a simulation by clicking Run Simulation under Simulation on the Flow Navigator pane. User can choose between Behavioral Simulation, Post-Synthesis Simulation, and Post-Implementation Simulation. In the Lattice Radiant software, user can click on the Simulation Wizard icon to run simulation using ModelSim, which is a directly linked third-party simulator.

Figure 5.1. Opening Simulation Wizard from the Toolbar

The primary use of simulation wizard is to generate a script that sets up and run the simulations. The Lattice Radiant software also comes with a standalone version of ModelSim that can be used for project simulation. User can open the standalone version by clicking on the ModelSim icon.

Figure 5.2. Opening the standalone ModelSim

5.1.1. Simulation Levels

Depending on the output files chosen on the process toolbar, user can perform different simulation levels. In Xilinx Vivado, there are three levels of simulation: behavioral, post-synthesis, and post-implementation with options of performing a functional or a timing simulation for post-synthesis and post-implementation.

Table 5.2. Simulation levels comparison between Xilinx Vivado and Lattice Radiant

Simulation Level in Xilinx Vivado	Equivalent Simulation Level in Lattice Radiant
Behavioral	RTL
Post-synthesis (Functional or Timing)	Post-Synthesis
Post-implementation (Functional or Timing)	Post-route gate-level, Post-route gate-level+timing

In the Lattice Radiant software, there are four simulation levels: RTL, post-synthesis, post-route gate-level, and post-route gate-level plus timing simulations. These provide the user more options when simulating the designs prior to hardware verification. User can choose between the simulation levels on the Process Stage of the Simulation Wizard window.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

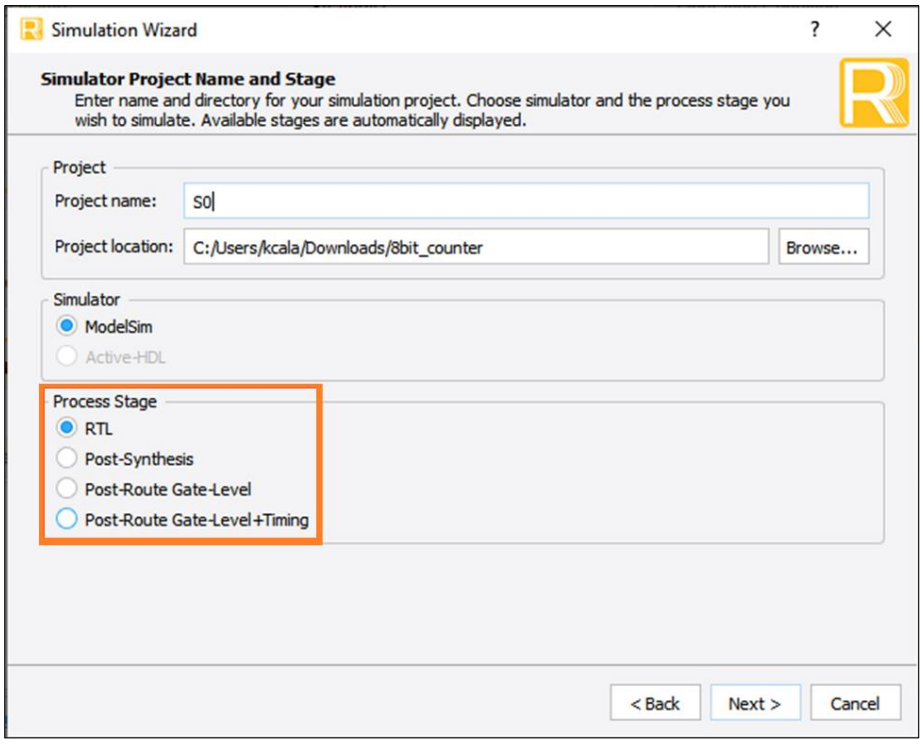


Figure 5.3. Simulation Levels in Simulation Wizard

Each simulation requires specific output files that can be generated on the design flow before simulation can be performed:

- RTL Simulation can be run with just an HDL file and a testbench file.
- Post-Synthesis Simulation requires a post-synthesis netlist file in addition to an HDL file and a testbench file. The post-synthesis netlist file is generated during the Synthesize Design process when the user selects the Post-Synthesis Simulation File.

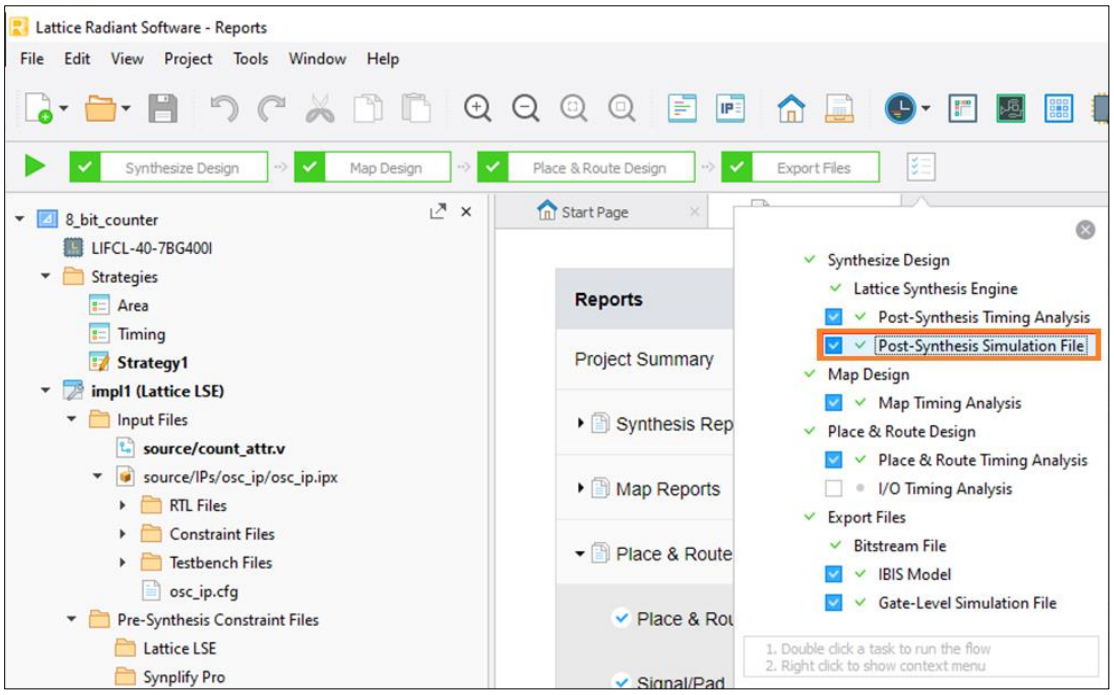


Figure 5.4. Generating a Post-Synthesis Simulation File

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

- Post-Route Gate-Level requires a gate-level netlist file in addition to an HDL file and a testbench file. The gate-level netlist file is generated during the Export Files process when Gate-Level Simulation File is selected.
- Post-Route Gate-Level+Timing requires a gate-level netlist file and a standard delay format file (.sdf) in addition to an HDL file and a testbench file. The gate-level netlist file is generated during the Export Files process when Gate-Level Simulation File is selected.

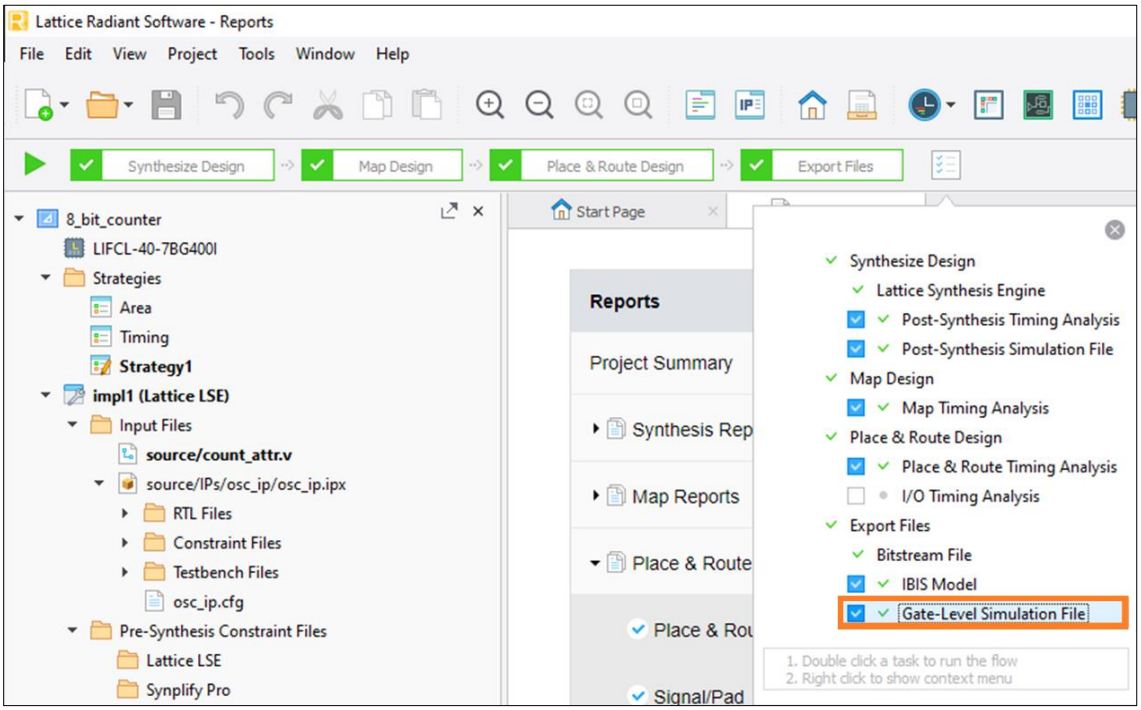


Figure 5.5. Generating Gate-Level Simulation Files

For more information, refer to the Lattice Radiant software help or to the following sections in Lattice Radiant Software 3.1 User Guide:

- Siemens ModelSim
- Simulation Wizard
- Simulation Flow
- Simulation Wizard Flow

5.2. Static Timing Analysis (STA)

Vivado has a number of reports related to STA that would guide the user on how to close timing. The Lattice Radiant timing analyzer analyses and reports timing performance of all logic in the design while checking all possible paths with timing violations.

Timing Analyzer analyzes timing constraints that are present in the pre-synthesis constraint files (.sdc, .ldc, or .fdc) and post-synthesis constraint file (.pdc). These timing constraints are defined in the Timing Constraint Editor or in a text editor before the design is mapped. A Timing Analysis report file, which shows the results of timing constraints, is generated each time the user runs the synthesis engine, Map Timing Analysis process, or the PAR Timing Analysis process. PAR Timing Analysis results can then be viewed in the Timing Analyzer window, which can be opened by clicking on the Timing Analyzer icon on the toolbar.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

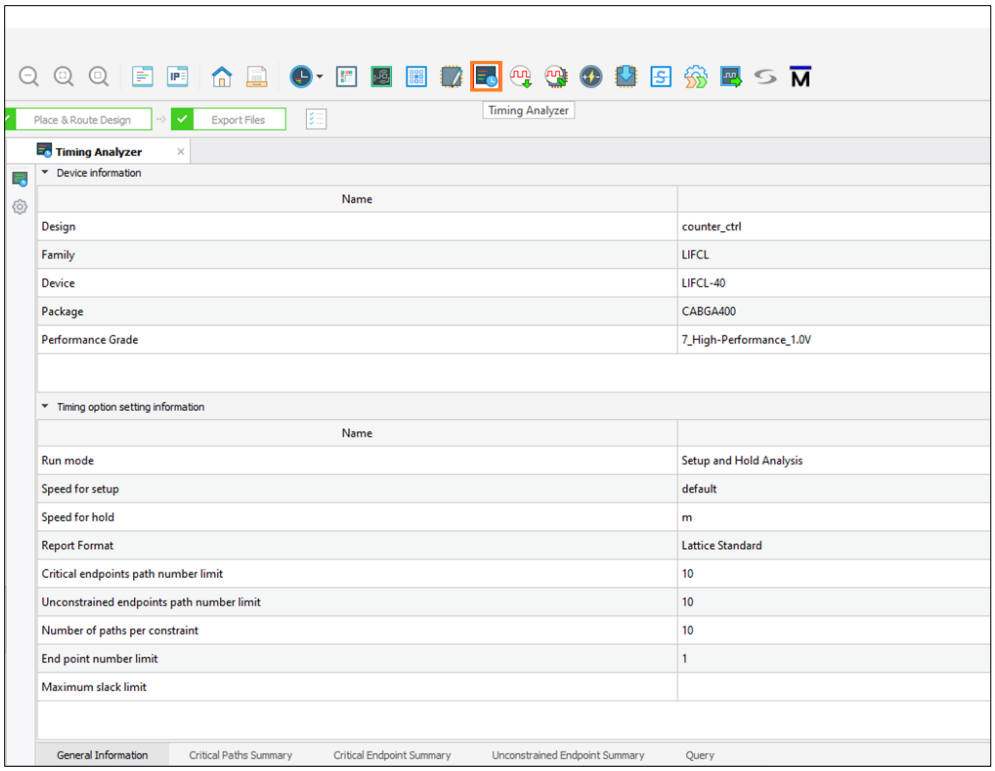


Figure 5.6. Opening the Timing Analyzer Tab

For more information regarding Timing Analyzer, refer to the *Analyzing Static Timing* topic in the Lattice Radiant software help or to the following sections in Lattice Radiant Software 3.1 User Guide:

- Timing Analyzer
- Using Stand Alone Timing Analyzer

5.3. On-Chip Hardware Debugging using Reveal

The final stage of developing the design is the actual verification process on either a test board or in the system. The on-chip debugging tools allow live hardware aspect checking in the design which helps to quickly do a verification without the use of any external equipment.

While Xilinx Vivado software offers multiple portfolios of on chip debugging tools integrated in the Hardware Manager, the Lattice Radiant environment offers the Reveal Inserter and Reveal Analyzer/Controller tool to continuously monitor signals within the FPGA for specific conditions, which can range from simple to quite complex and see what is happening inside the FPGA and to even change register values while the system is running.

- Reveal Inserter, which user can use to create a debug module and add it to the design.
- Reveal Analyzer/Controller, which user can use to control the debug module and to view test results. Reveal Analyzer/Controller is used after programming the FPGA with the combined design and debug module.

For more information on using Reveal, refer to the following documents:

- Lattice Radiant Software Reveal User Guide
- Reveal Troubleshooting Guide for Lattice Radiant Software

User can also refer to the following sections in Lattice Radiant Software 3.1 User Guide:

- Reveal Inserter
- Reveal Analyzer
- Reveal Controller

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.4. Power Analysis

Similar to Xilinx Vivado's Power Estimator (XPE), the Lattice Radiant software offers the Power Calculator, which estimates the power dissipation for a given design.

Table 5.3. Power Analysis Tools Comparison

Xilinx Vivado	Lattice Radiant Software
Xilinx Power Estimator (XPE) Report Power	Power Calculator

The Power Calculator uses parameters such as voltage, temperature, process variations, air flow, heat sink, resource utilization, activity, and frequency to calculate the device power consumption. It reports both static and dynamic estimated power consumption. The tool also allows user to import frequency and activity factors from the post-PAR simulation *value change dump* file (.vcd file). After the design information is added, the Power Calculator provides accurate power consumption analysis for the design.

The Power Calculator provides two modes for reporting power consumption:

- Estimation mode is used before completing the design.
- Calculation mode is based on the physical netlist file (.udb) after placement and routing.

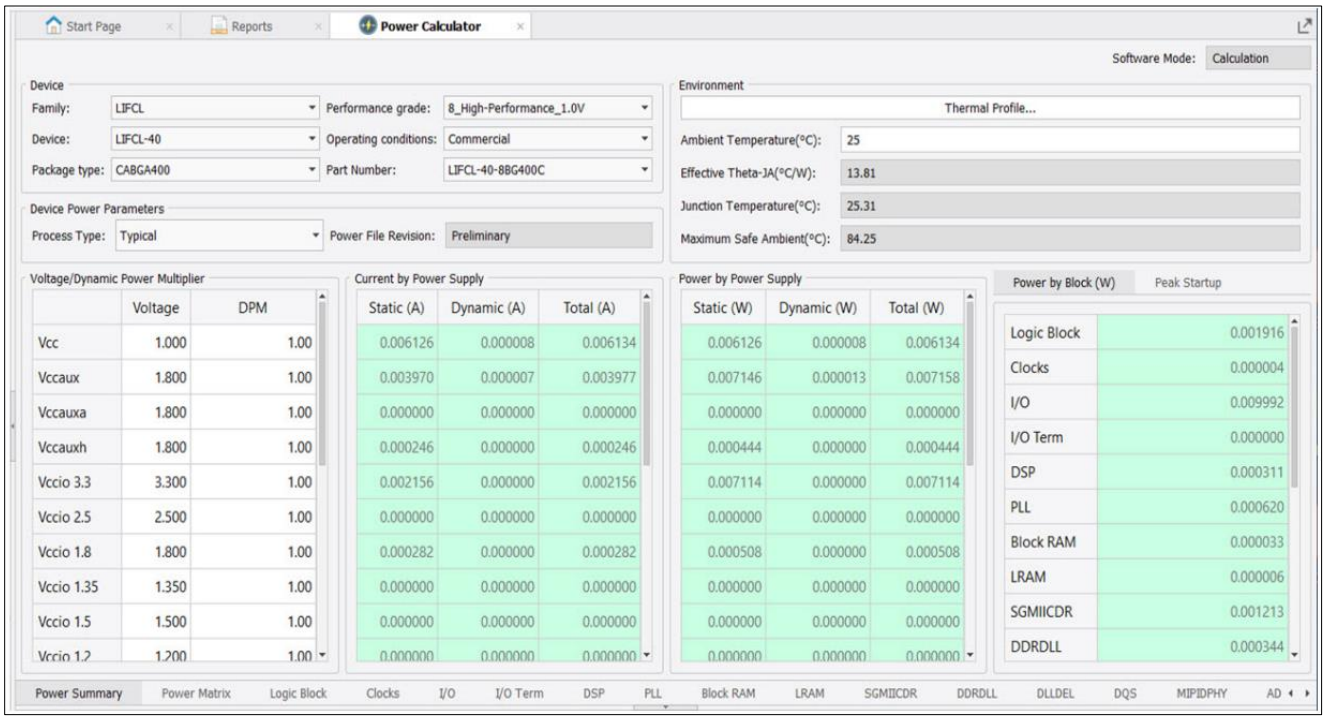


Figure 5.7. Power Calculator

For more information, refer to the Lattice Radiant software help or to Lattice Radiant Software 3.1 User Guide Power Calculator section.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6. TCL Scripting

Similar to Xilinx Vivado software batch mode, the Lattice Radiant also support TCL (Tool Command Language) scripting feature that enables a batch capability for running tools in the Lattice Radiant software's graphical interface. TCL commands can be used through the command line/terminal or the Lattice Radiant Stand-Alone TCL console that is included in the software package. The command set and the TCL console used to run it affords the user the speed, flexibility and power to extend the range of useful tasks that the Radiant software tools are already designed to perform.

Using the command line tools permits the user to do the following:

- Develop a repeatable design environment and design flow that eliminates setup errors that are common in GUI design flows
- Create test and verification scripts that allow designs to be checked for correct implementation
- Run jobs on demand automatically without user interaction

Note: The environments for both the Lattice Radiant TCL console window or the Lattice Radiant Standalone TCL console window are already set. User can start entering the TCL tool command or core tool commands in the console and the software performs them. When running the Lattice Radiant software from the Windows command line (through cmd.exe) or Linux terminal (bash), user needs to set up the environment variables as stated in the *Setting Up the Environment to Run Command Line* section of the Lattice Radiant software help.

For more information, refer to Lattice Radiant Software 3.1 User Guide.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

7. Lattice Propel

Similar to the Vitis Core Development Kit in Xilinx, the Lattice Propel is a design environment for Lattice FPGA-based processor system designs. The development suite includes:

- Integrated development environment (IDE)
- Lattice Propel Builder graphical user interface for System-on-Chip (SoC) design
- Lattice Propel Software Development Kit (SDK) for system software development based on Eclipse Embedded C/C++ Development tools (CDT)

The Lattice Propel flow starts with the creation on an SoC system design in the Lattice Propel Builder using the RISC-V CPU together with APB and AHB-L buses, and peripherals.

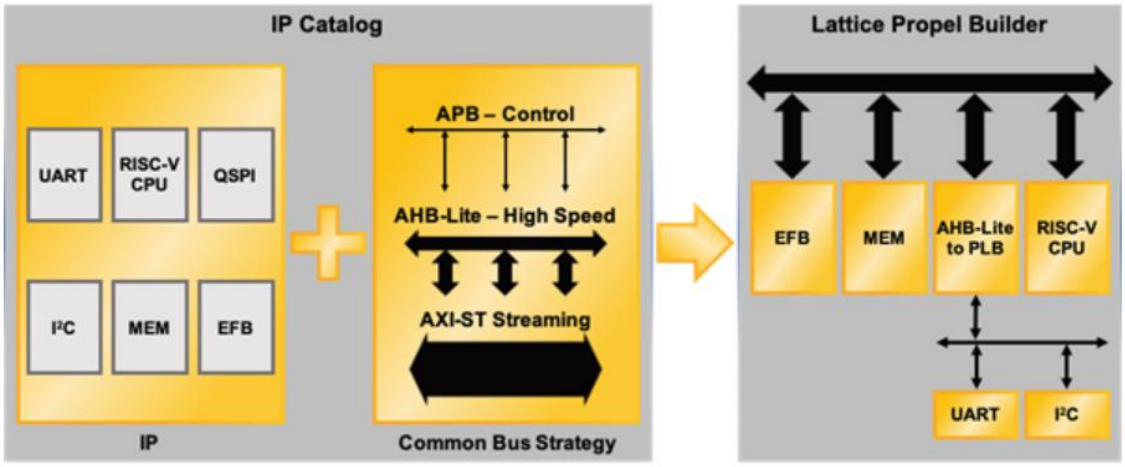


Figure 7.1 Lattice Propel Builder Design Flow

A C/C++ project is then created using the environment file of the SoC design. User can input the executable code on the SoC design as part of the RTL. Typically, this process can be done by loading the memory file on the system memory of the SoC design.

For an SOC design, if the memory file of the System memory module has been updated in a C project the initialization section of the system memory module instance should be re-configured or use the ECO Editor tool to update the information. The ECO (Engineering Change Order) Editor enables the user to make changes to an implemented design without having to rerun the entire process flow.

For more information on ECO editor, refer to Lattice Radiant Software 3.1 User Guide.

The next step is to generate the RTL files and then go through the design flow using Lattice Diamond® or the Lattice Radiant software. Once the configuration is loaded onto the SRAM, the design can be verified through Reveal or through GNU debugger of the SDK.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

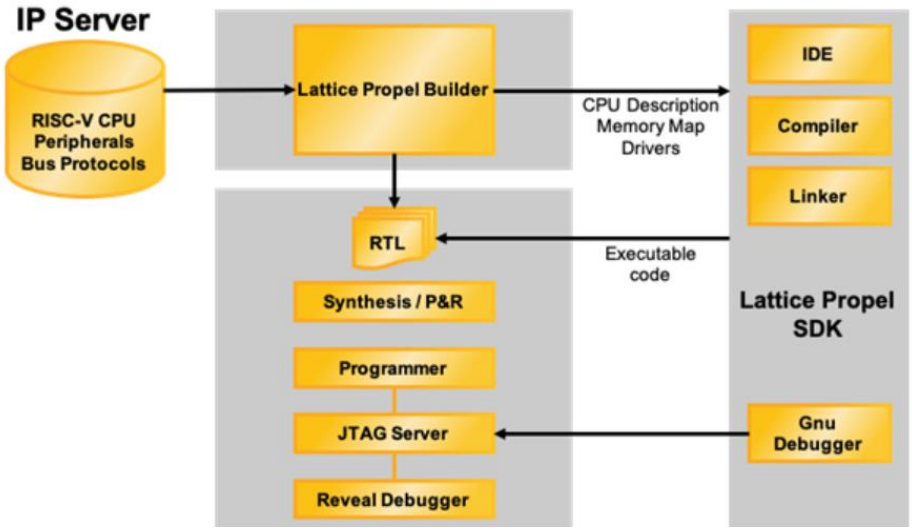


Figure 7.2. Lattice Propel Design Environment

For more information on Lattice Propel, user can find all materials on the Lattice Propel page of the Lattice website.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

References

- Lattice Radiant Software 3.1 Release Notes
- Lattice Radiant Software 3.1 User Guide
- Lattice Radiant Software Guide for Lattice Diamond Users
- Migrating iCEcube2 iCE40 UltraPlus Designs to Lattice Radiant Software
- Lattice Radiant Software IP User Guide
- Programming Tools User Guide for Radiant Software
- Reveal User Guide for Radiant Software
- Reveal Troubleshooting Guide for Lattice Radiant Software
- Lattice Radiant Software 3.1 Help (PDF)
- Lattice Radiant Software 3.1 Installation Guide for Windows
- Lattice Radiant Software 3.1 Installation Guide for Linux/Ubuntu
- Xilinx Documentation Portal
- Vivado Design Suite User Guide: Getting Started

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision History

Revision 1.0, July 2022

Section	Change Summary
All	Initial release.

© 2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

www.latticesemi.com