



FRONTGRADE

APPLICATION NOTE

Creating Certus-NX-RT Project in Radiant Software

3/18/2022

Version #: 1.0.0

Table 1: Cross Reference of Applicable Products

Product Name	Manufacturer Part Number	Device Type
Lattice Certus-NX-RT FPGA	UT24C407	RL01

1.0 Overview

This document details the process of creating a **Certus-NX-RT** FPGA project using the **Lattice Radiant** software tools. For the purposes of this document, create a project named **led_brightness** and configure the **Radiant** tools to include all the source modules required for a successful build. Using this template, projects are created using (a) the preferred application source directory structures and (b) the directory structure for the **Radiant**-supplied files. **Figure 1** shows the block diagram of the design, with the switches SW[3:0] controlling the duty cycle to the pwm output, which controls the brightness to LED[0]; LED[7:4] are used as heartbeat of the system.

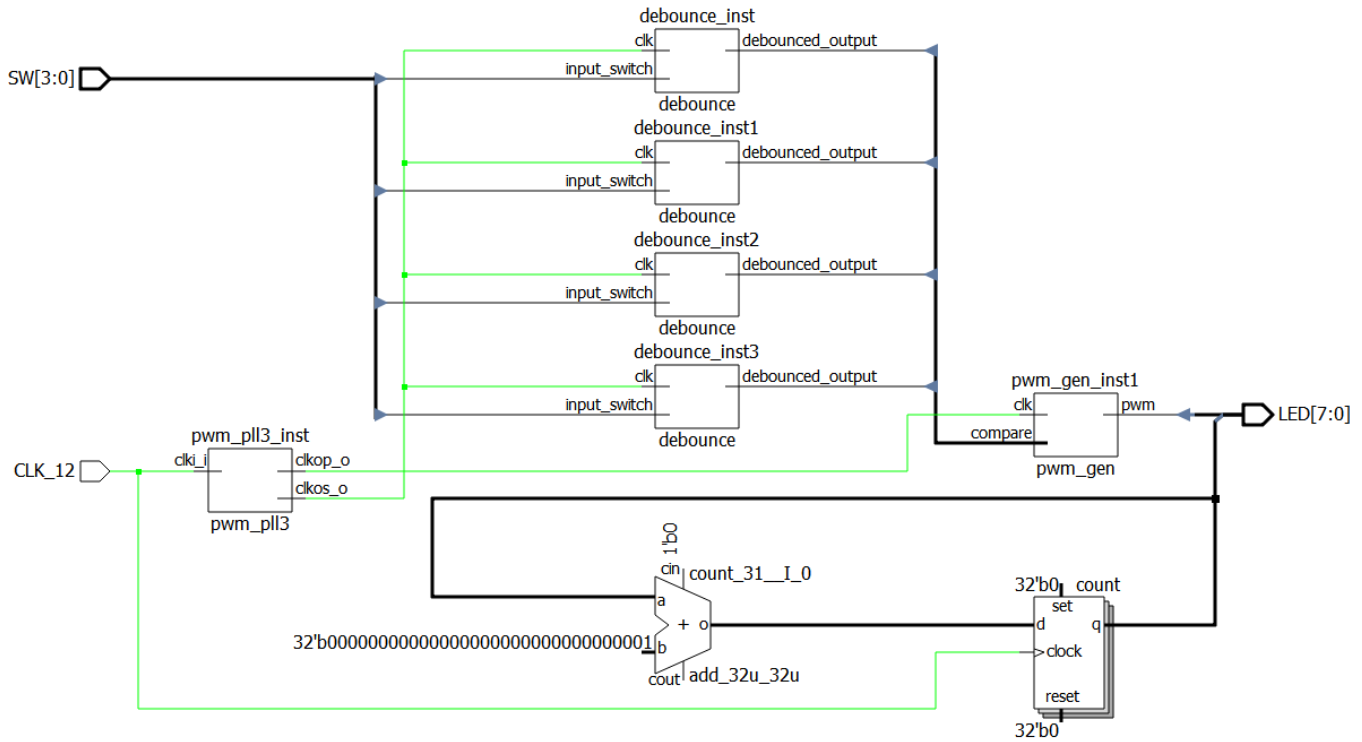


Figure 1: An Example Netlist View

2.0 Creating a Design Project with Radiant Software



1. Download **CreatingRadiantProject.zip** from frontgrade.com.
2. Launch **Radiant Software**
3. From the File menu, select **New > Project....**
4. Specify the project name as **led_brightness**, navigate to the **location** of your choice and click **Next**, see **Figure 2**.

New Project

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project:
Name: led_brightness
Location: C:/projects/Lattice/Certus-NX-RT/applications/led_brightness Browse...
Project will be created at C:/projects/Lattice/Certus-NX-RT/applications/led_brightness/ Create subdirectory

Implementation:
Name: impl_1
Location: C:/projects/Lattice/Certus-NX-RT/applications/led_brightness/led_brightness/impl_1

< Back Next > Cancel Help

Figure 2: Creating a Radiant Design Project

5. Select **Add Source...** and click **Open**, see **Figure 3** and **Figure 4**, click **Next**.

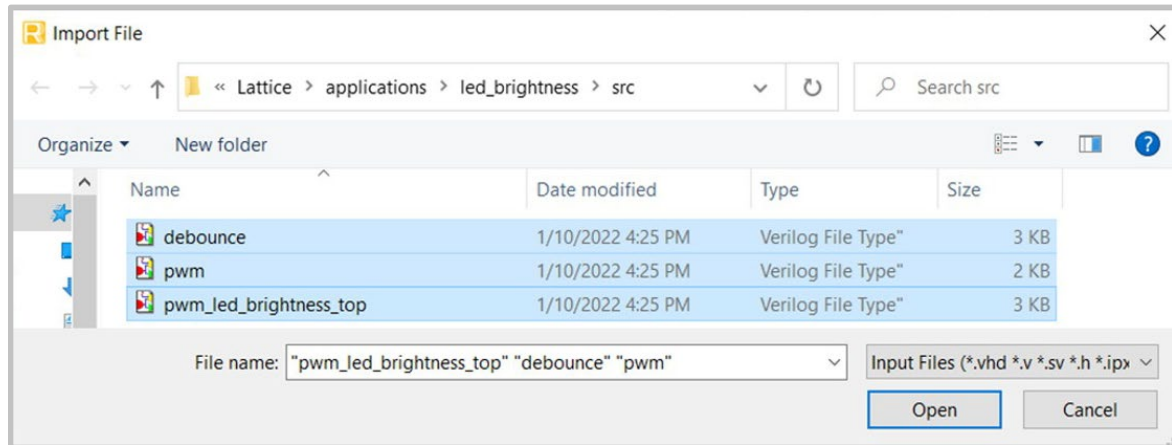


Figure 3: Selecting Sources

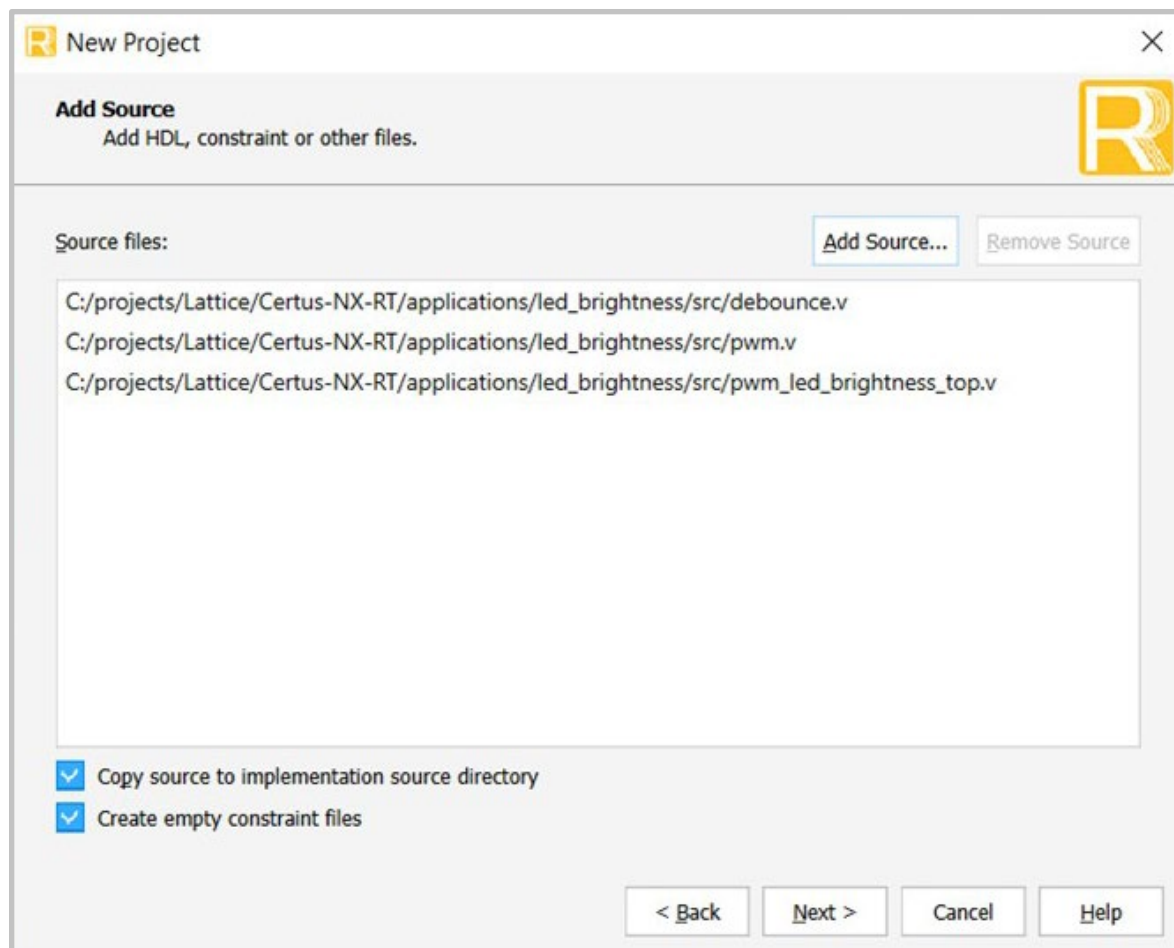


Figure 4: Adding Sources

6. Select the appropriate device, click **Next** and choose **Lattice LSE**; click **Next** and **Finish**, see **Figure 5**.

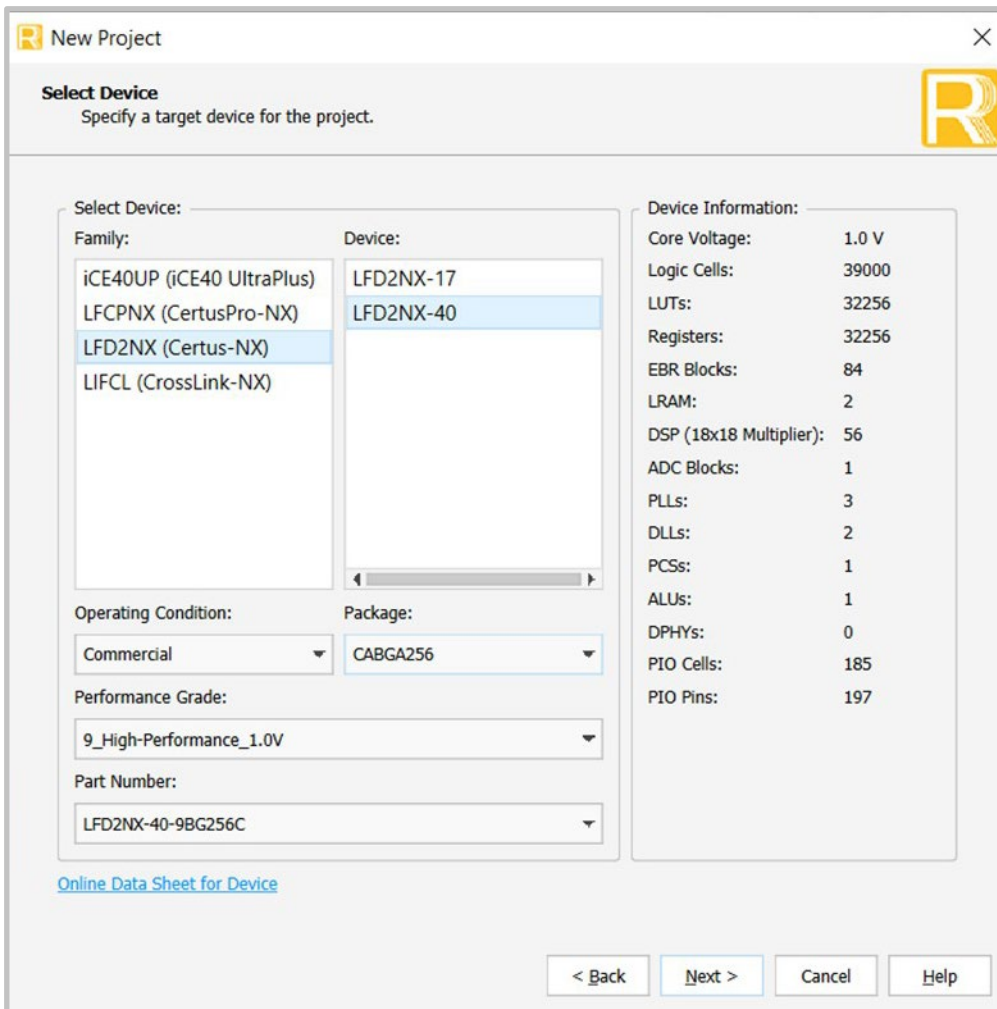


Figure 5: Selecting the FPGA Device

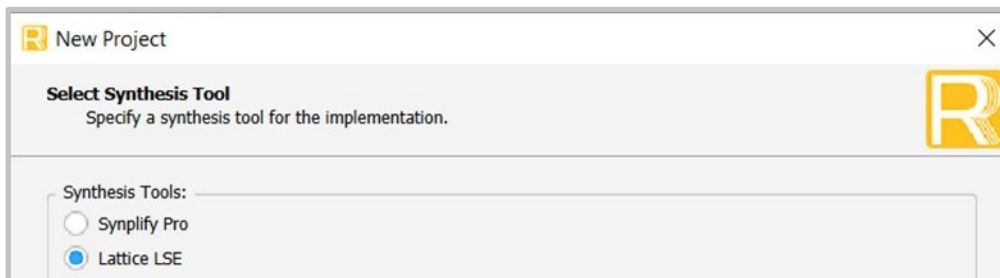


Figure 6: Selecting the Synthesis Tool

7. The initial project with the provided files should look like **Figure 7**.

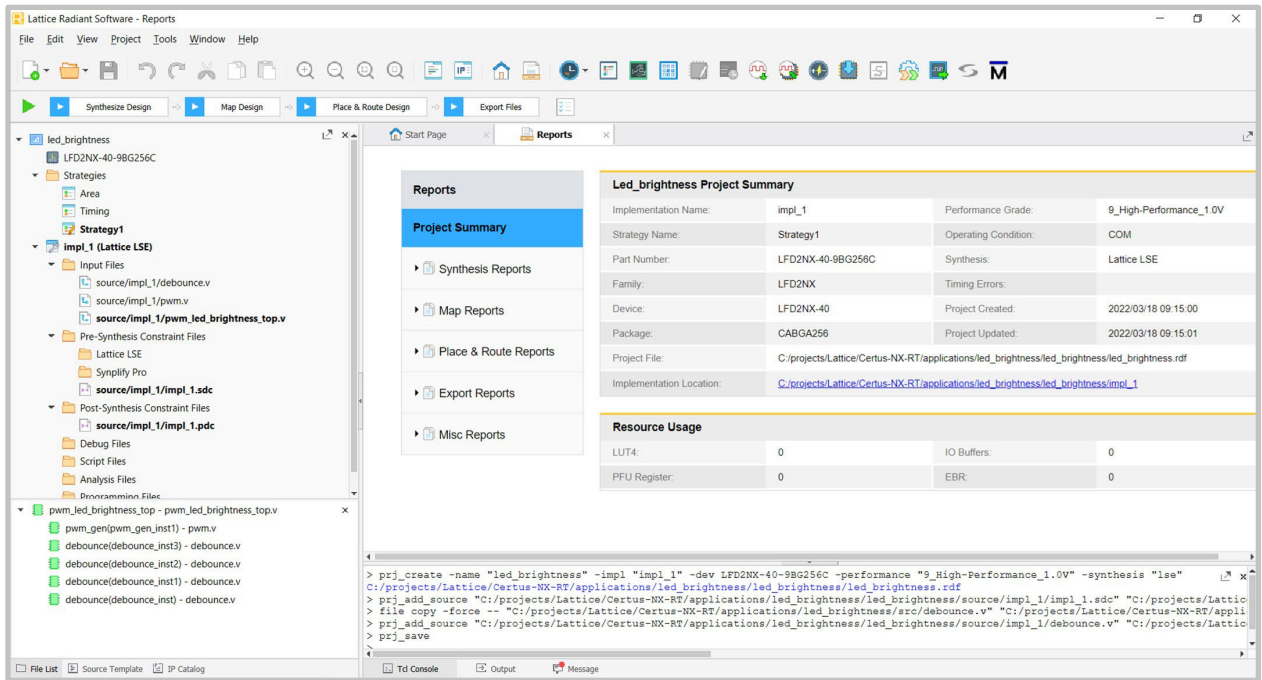


Figure 7: An Example of an Initial Project

8. From IP Catalog, add a PLL with the options below and name it **pwm_pll3**, see **Figure 8** and **Figure 9**.

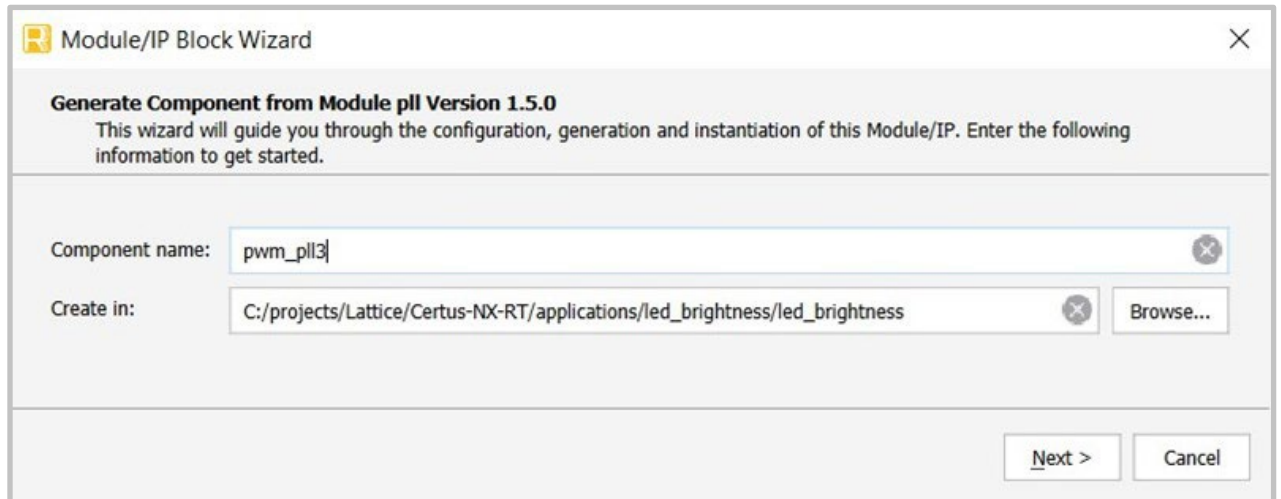


Figure 8: Opening the IP Catalog

The screenshot displays the configuration interface for a PLL block named 'pwm_pll3'. On the left, a block diagram shows the 'pll' block with an input 'clki_i' and two outputs: 'clkop_o' and 'clkos_o'. On the right, the 'Configure pwm_pll3' window is open, showing various configuration parameters.

Property	Value
General	
Configuration Mode	Frequency
Enable Fractional-N Divider	<input type="checkbox"/>
Enable Spread Spectrum Clock Generation	<input type="checkbox"/>
Enable User Feedback Clock	<input type="checkbox"/>
Enable PMU Wait for Lock	<input type="checkbox"/>
Enable Internal Path Switching	<input type="checkbox"/>
VCO Frequency [800 - 1600]	960
Reference Clock	
CLKI: Frequency (MHz) [10 - 800]	12
CLKI: Divider Actual Value [1 - 80]	1
Phase Detector Frequency (MHz) [10 - 500]	12
Enable Reference Clock Monitor	<input type="checkbox"/>
Feedback	
CLKFB: Feedback Mode	CLKOP
CLKFB: FBK Divider Actual Value (Integer) [1 - 128]	2
Primary Clock Output	
CLKOP: Frequency Desired Value (MHz) [10 - 800]	25
CLKOP: Frequency Actual Value (MHz) [10 - 800]	24
CLKOP: Divider Actual Value [1 - 128]	40
CLKOP Tolerance (%)	5.0
CLKOP: ERROR (PPM)	40000
CLKOP: Enable Trim for CLKOP	<input type="checkbox"/>
Secondary Clock Output	
CLKOS: Enable	<input checked="" type="checkbox"/>
CLKOS: Bypass	<input type="checkbox"/>
CLKOS: Frequency Desired Value (MHz) [6.25 - 800]	10
CLKOS: Divider Actual Value [1 - 128]	96
CLKOS Tolerance (%)	0.2
CLKOS: ERROR (PPM)	0
CLKOS: Static Phase Shift (Degrees)	0
CLKOS: Enable Trim for CLKOS	<input type="checkbox"/>

Figure 9: Adding a PLL from the IP Catalog

- Once the PLL is created, it is added to the project files, see **Figure 10**.

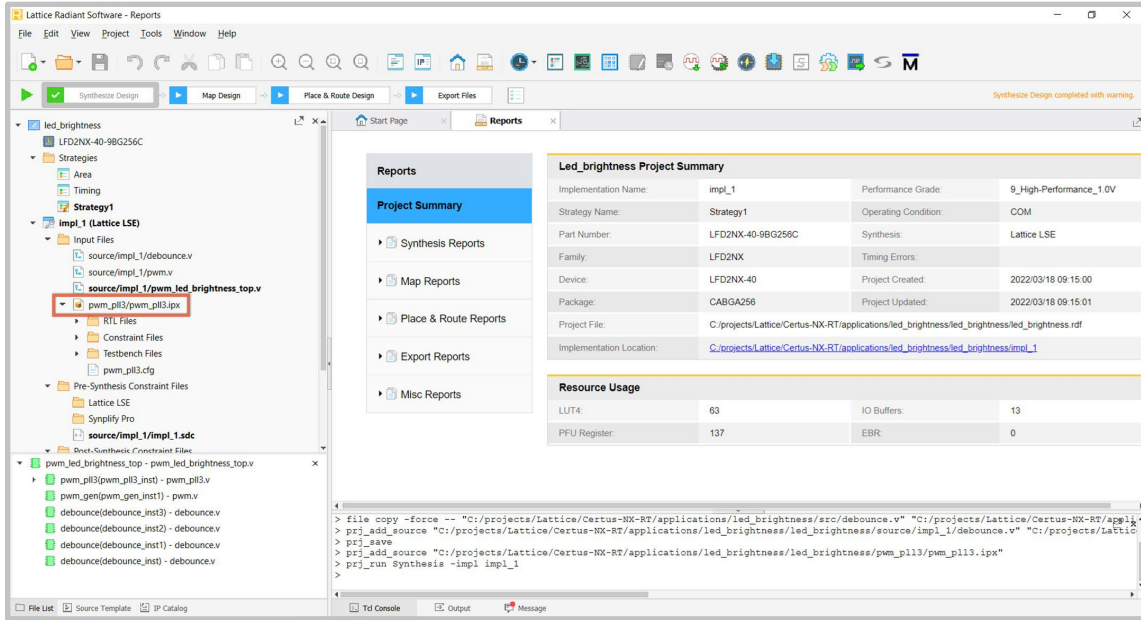


Figure 10: Looking at the Recently Added PLL IP

3.0 Compile Design using Radiant Software

3.1 Set Top Level File

- From the **Project** menu, select **Active Implementation > Set Top-Level Unit...** and set **pwm_led_brightness_top.v** as the top level file, see **Figure 11**.

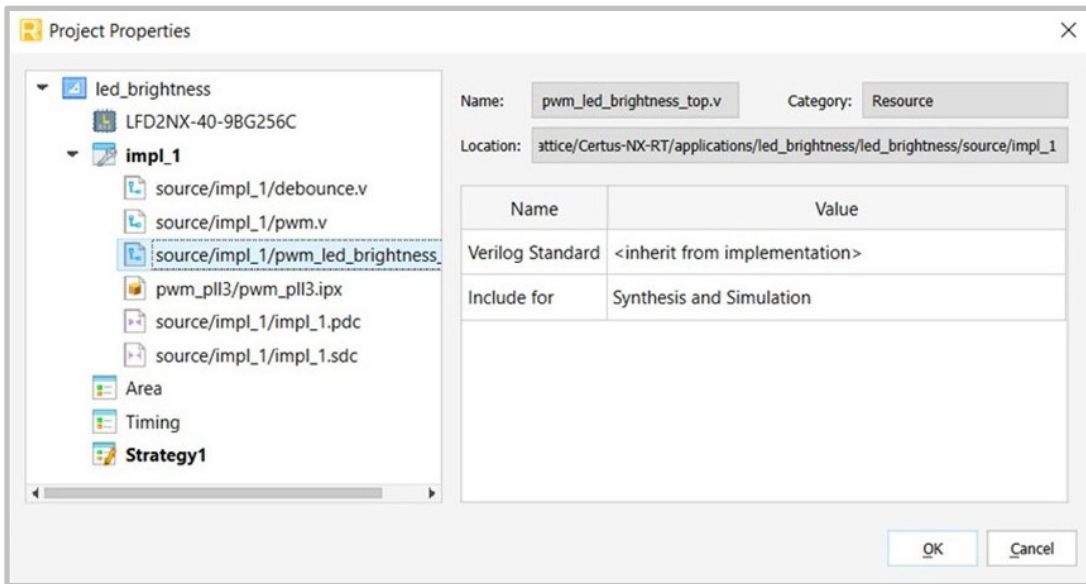


Figure 11: Setting a Top Level File

3.2 Synthesize Design

11. Synthesize the design by clicking on the **Synthesize Design** arrow, see **Figure 12**.

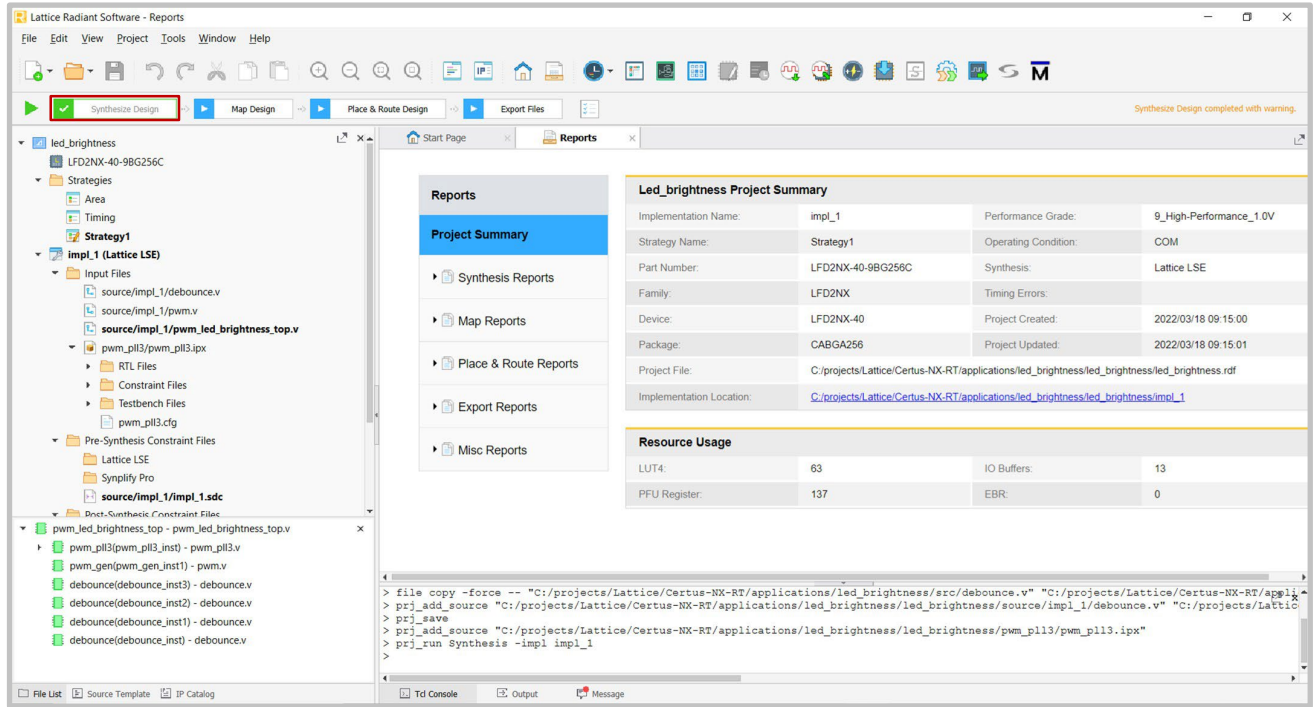


Figure 12: Synthesizing the Design

3.3 Pin Assignment (Device Constraint Editor)

12. Once the design has been synthesized, assign the pin numbers to their respective signals, see **Figure 13**.

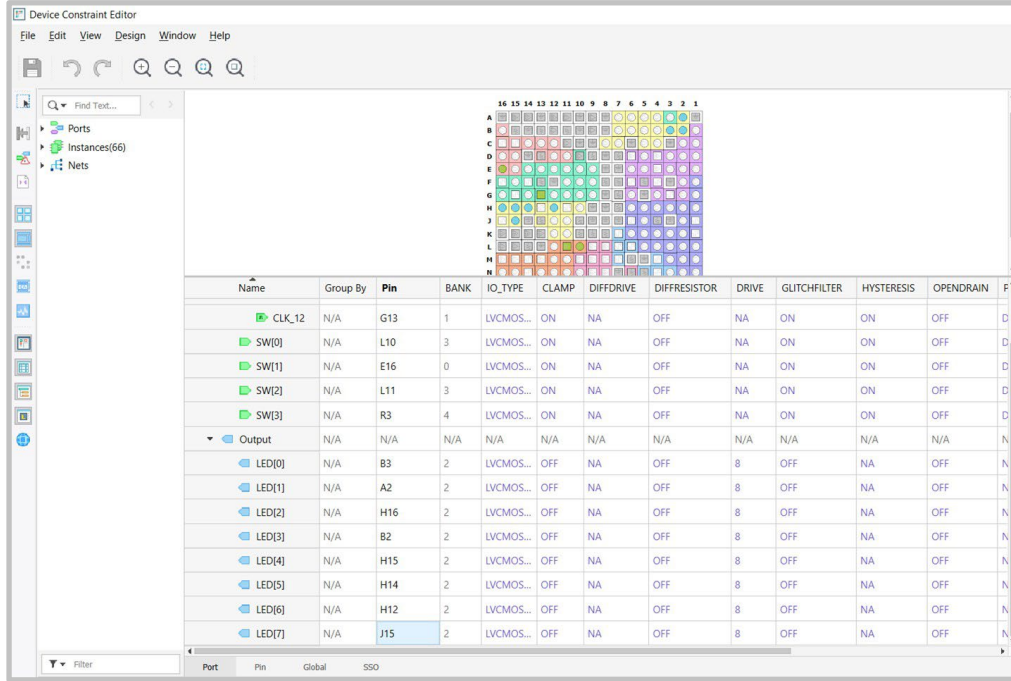


Figure 13: Performing Pin Assignment

3.4 Compile Design

13. Compile the design by clicking on the **Run All** arrow, see **Figure 14**.

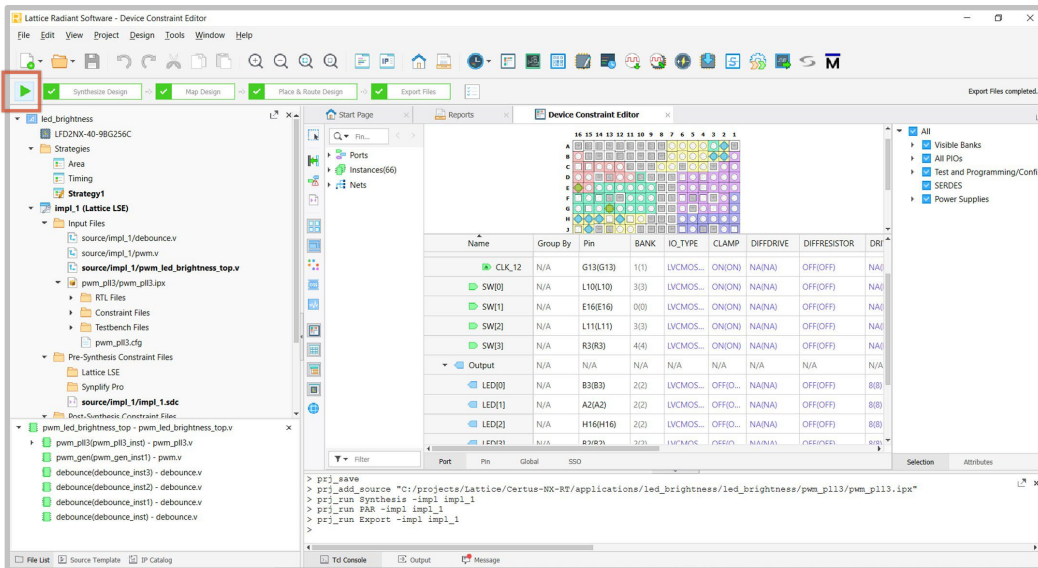


Figure 14: Compiling the Design

3.4 Simulate Design

14. From the **Tools** menu, select **Simulation Wizard** and click **Next**, see **Figure 15**.

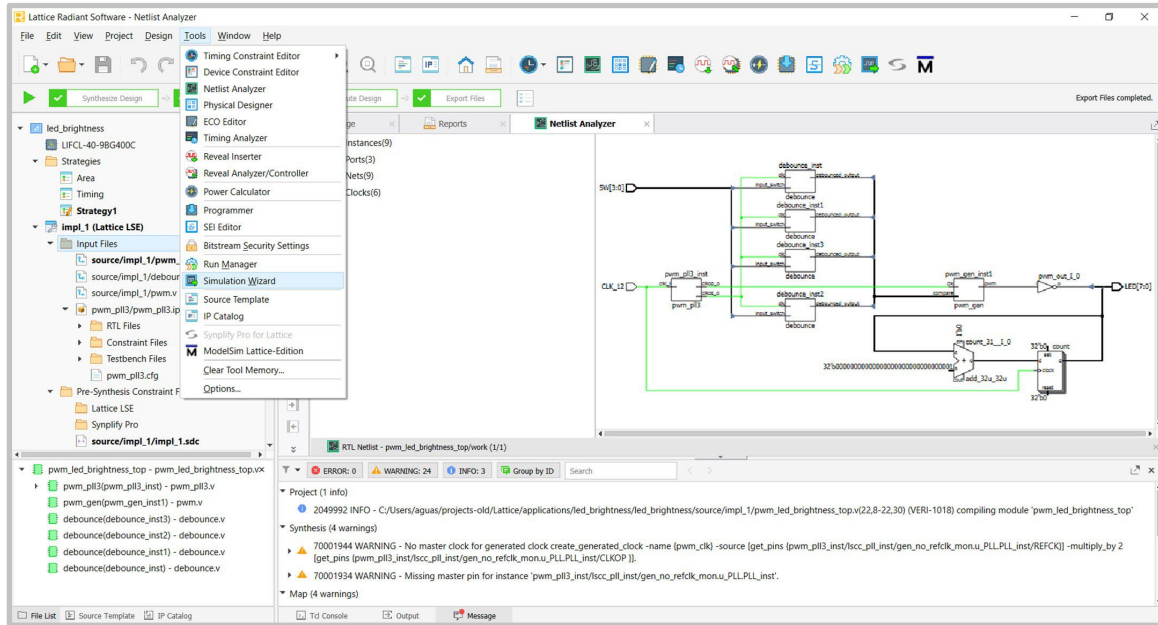


Figure 15: Simulating the Design

15. Name the project **led_brightness_sim** and click **Next**, see **Figure 16**.

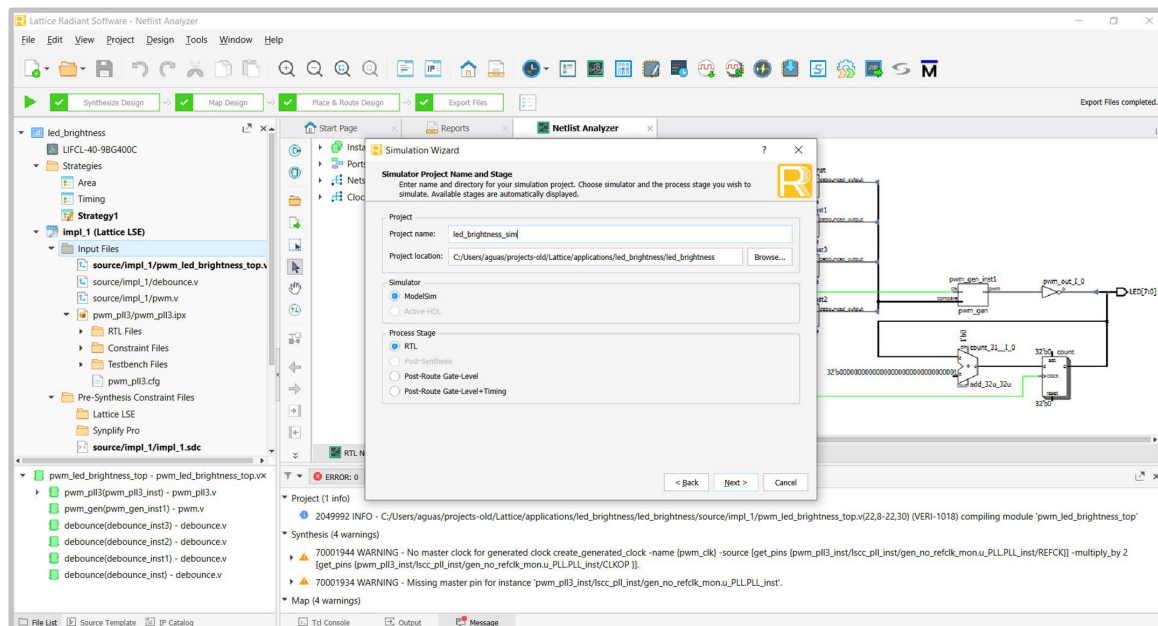


Figure 16: Naming the Simulation

16. When finished, ModelSim will open and run the simulation, see **Figure 17**.

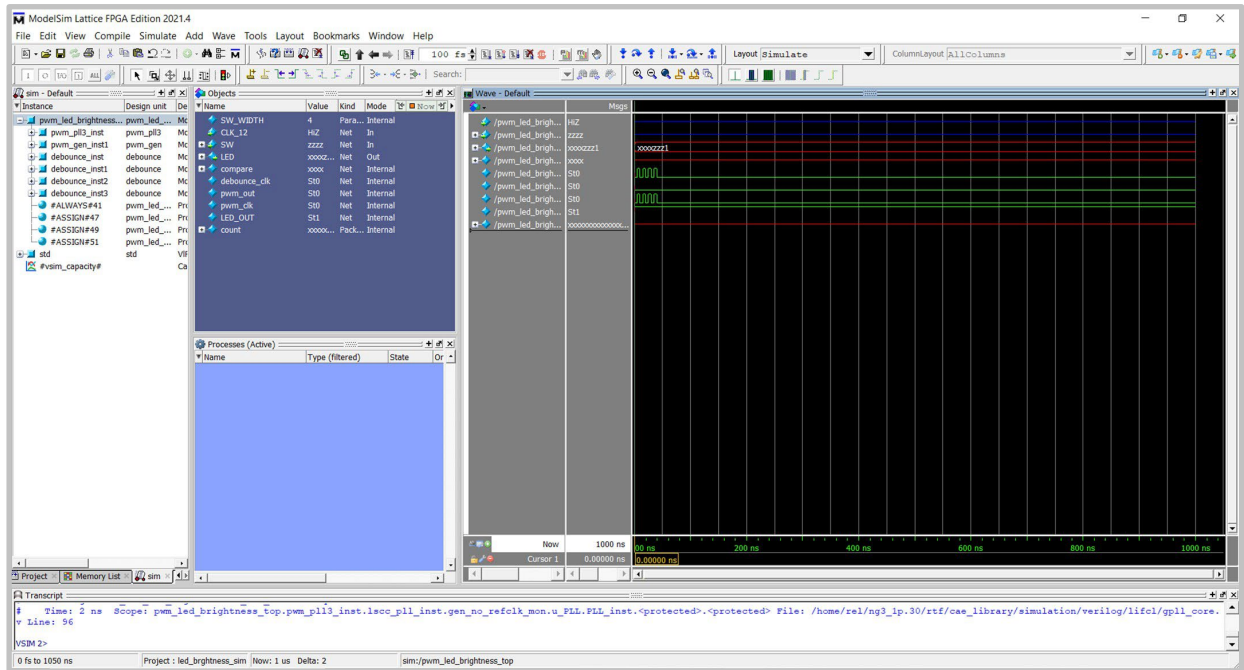



Figure 17: The ModelSim Simulation Wave View

4.0 Program the Device with Radiant Programmer

- From the **Tools** menu, select **Programmer** and a new window opens, see **Figure 18**. Once the cable is detected and settings are set, program the device by clicking the Program Device icon  , **Figure 18**.

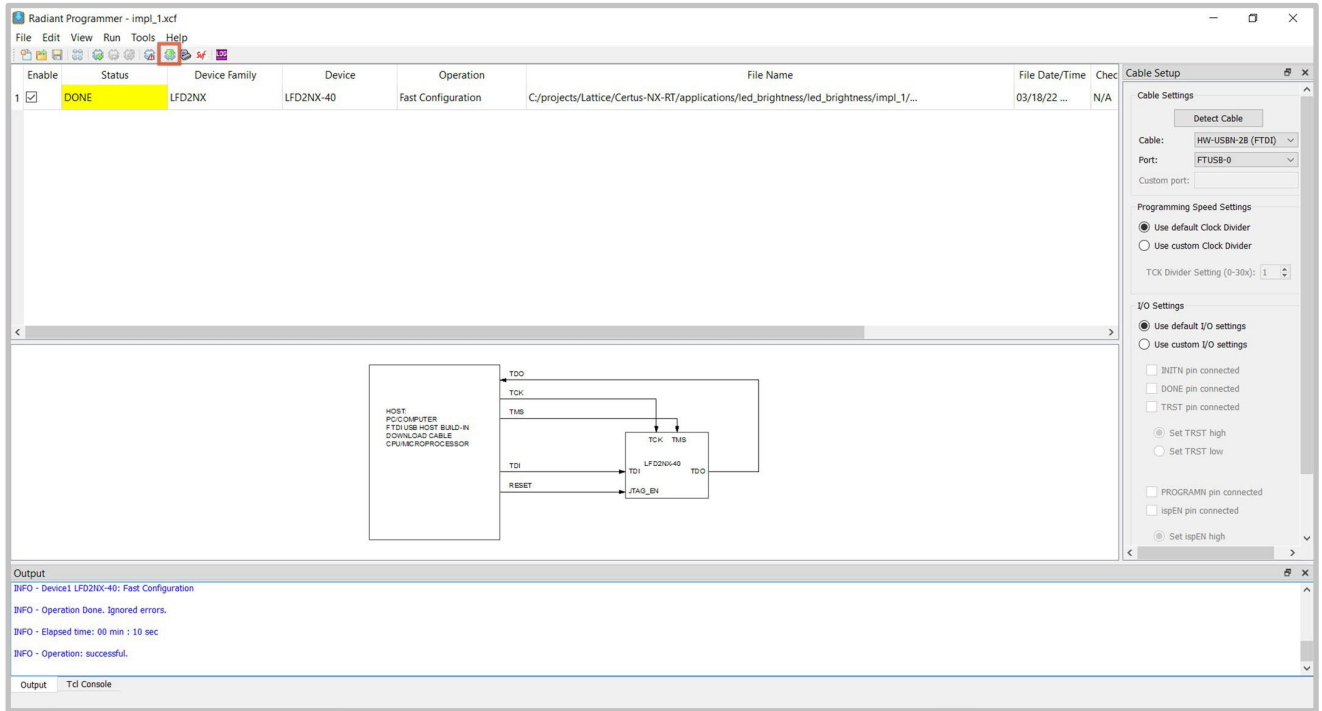


Figure 18: Programming the FPGA

Revision History

Date	Revision #	Author	Change Description	Page #
3/18/2022	1.0.0	JA	Initial Release.	

Datasheet Definitions

	Definition
Advanced Datasheet	Frontgrade reserves the right to make changes to any products and services described herein at any time without notice. The product is still in the development stage and the datasheet is subject to change . Specifications can be TBD and the part package and pinout are not final .
Preliminary Datasheet	Frontgrade reserves the right to make changes to any products and services described herein at any time without notice. The product is in the characterization stage and prototypes are available.
Datasheet	Product is in production and any changes to the product and services described herein will follow a formal customer notification process for form, fit or function changes.

Frontgrade Technologies Proprietary Information Frontgrade Technologies (Frontgrade or Company) reserves the right to make changes to any products and services described herein at any time without notice. Consult a Frontgrade sales representative to verify that the information contained herein is current before using the product described herein. Frontgrade does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by the Company; nor does the purchase, lease, or use of a product or service convey a license to any patents, rights, copyrights, trademark rights, or any other intellectual property rights of the Company or any third party.