FRONTGRADE

Creating Certus-NX-RT SoC Project in Radiant Software

4/8/2022

Version #: 1.0.0

Table 1: Cross Reference of Applicable Products

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

1.0 Overview

This document details the process of creating a **Certus-NX-RT SoC FPGA** project using the **Lattice Propel** software tools. For the purposes of this document, create a project named **Hello_World_SoC** and **Hello_World** using **Propel Build** 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.
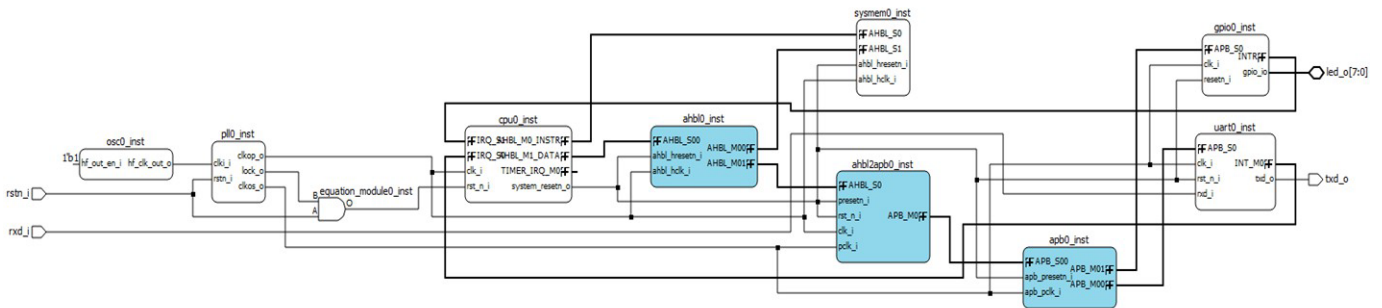

Figure 1: Example Design Block Diagram

2.0 Lattice Propel: Software Development Tools

Lattice provides software development tools for system-on-chip **(SoC)** designs: **Lattice Propel Development Suite,** which provides software build tools **(SBT)** for **Eclipse.** Eclipse is an integrated development environment **(IDE)** featuring:

- Creating and debugging applications.
- Using **Lattice SBT,** allows automatic creation of a board support package (BSP) for the given SoC.
- The **BSP** allows communication to the different peripherals in the **SoC.**
- The **Lattice SBT** automatically creates the make files for building the application.
- **Lattice Propel** for **Eclipse** software development kit **(SDK)** provides:
  — a **GUI** for code editing, compiling, and debugging using the GNU toolchain, which provides the compiler, assembler and linker, see **Figure 2.**
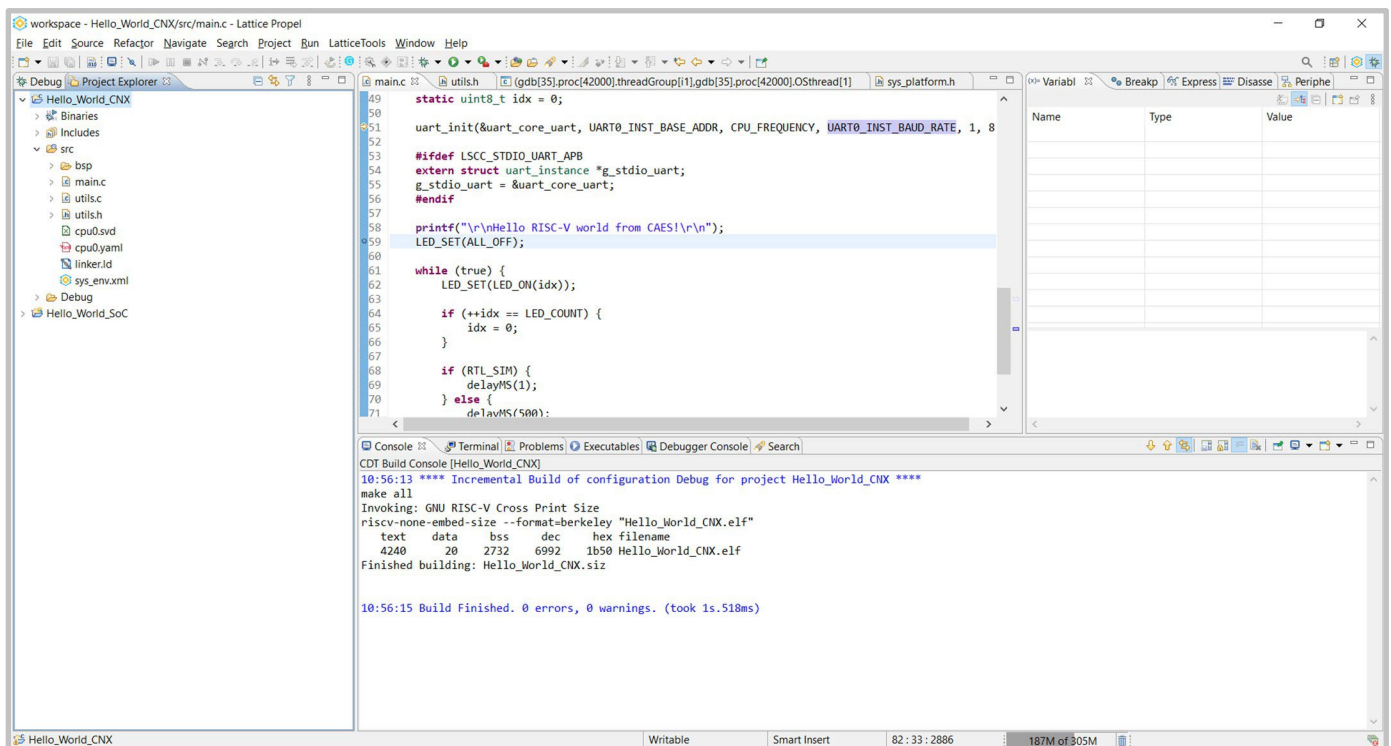

Figure 2: Lattice Propel

3.0 Lattice Propel Builder

Lattice Propel Builder allows for easy design of a **SoC** by simply dragging and dropping modules into a schematic view. **Propel Builder** provides:

- **GUI** for designing a **SoC** system.
- Generate the **SoC** design.
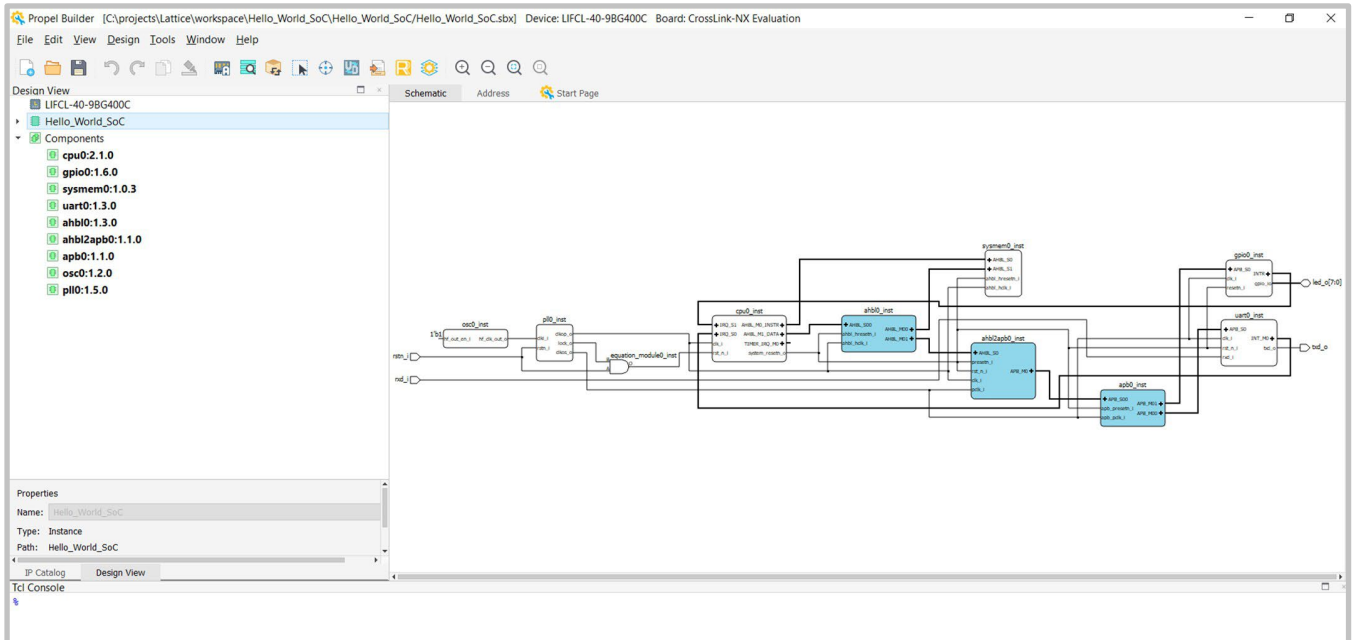- Integrate the **SoC** design with **Lattice Radiant Software.**

Figure 3: Lattice Propel Builder

4.0 Lattice Propel: Creating a SoC Design Project

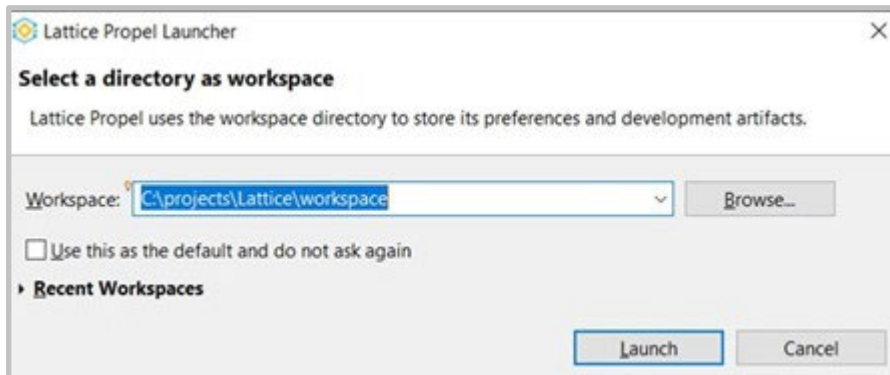1. Launch **Lattice Propel** and choose the workspace directory, see **Figure 4**.

Figure 4: Setting a Workspsace Directory

2.   From the **Project Explorer,** select **Create a new Lattice SoC Design Project,** see **Figure 5.**
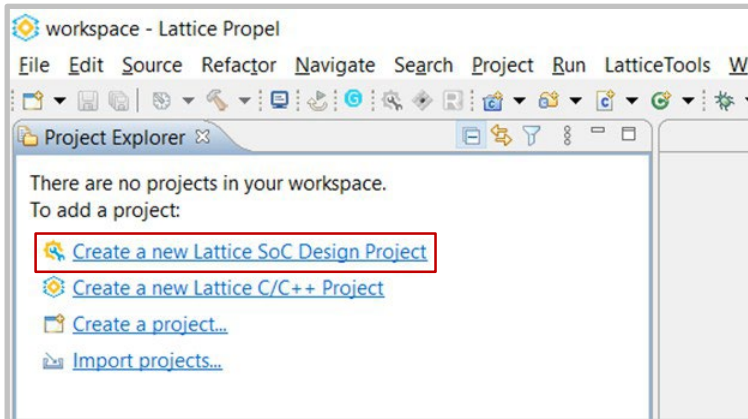

Figure 5: Creating a new Propel Design Project

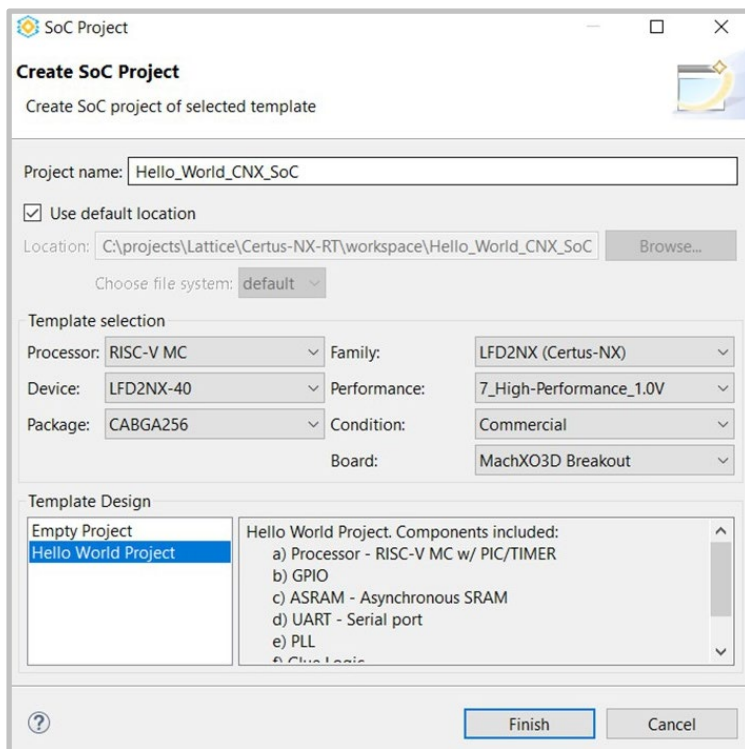3.   Specify the project name as **Hello_World_CNX_SoC** and click Finish, see **Figure 6.**

Figure 6: Hello World Project Setup

4. After clicking **Finish, Propel Builder** opens with a view of the template **SoC** design, see **Figure 7.**
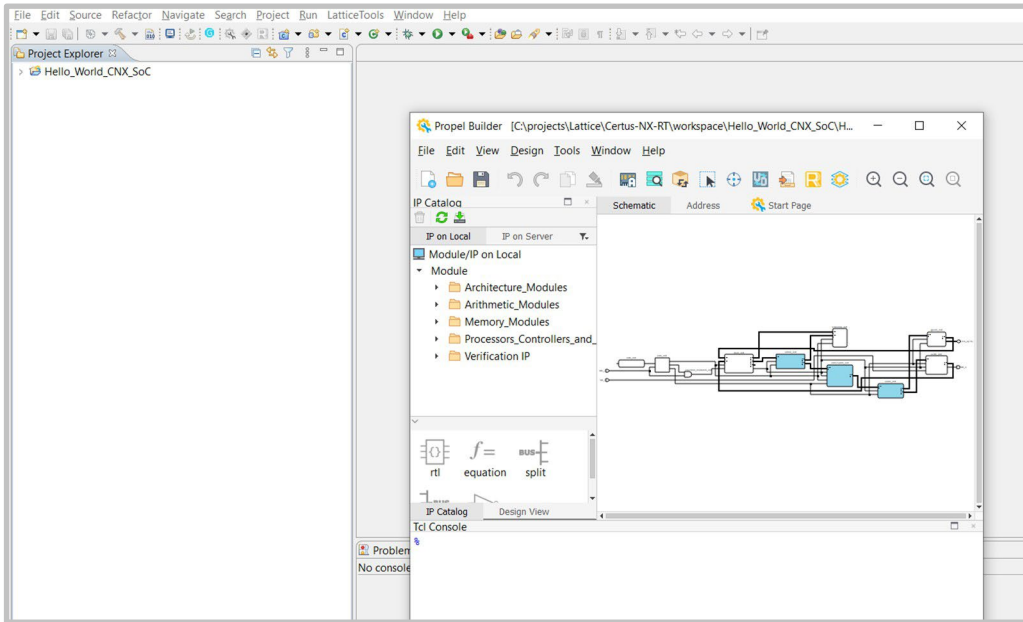

Figure 7: Default Propel Builder View

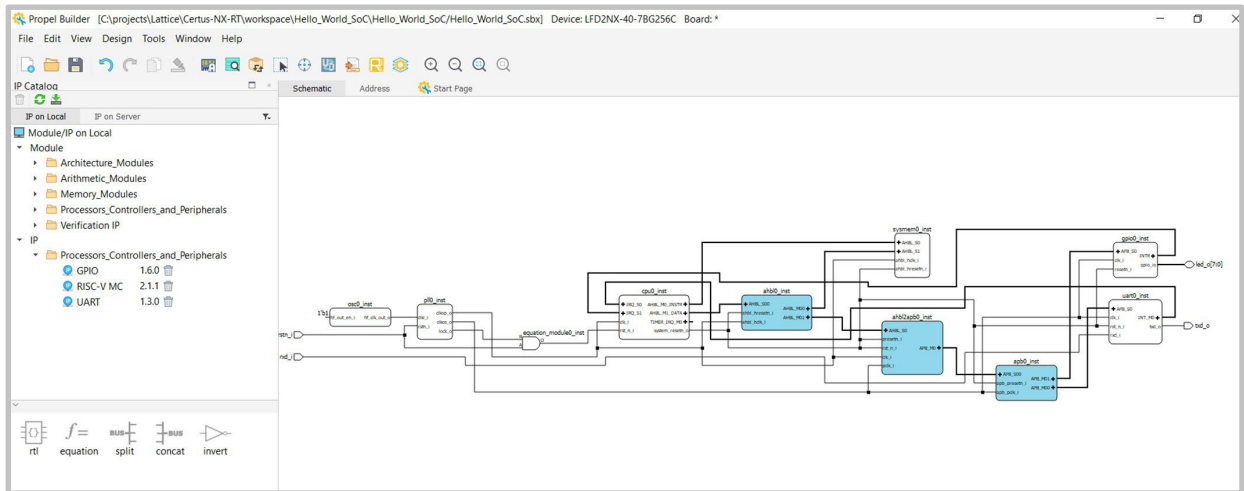5. **Design View** shows all the components for creating the template **SoC** design, see **Figure 8**.

Figure 8: Design View of SoC Design

6. For the provided template, the default **SoC** design is sufficient; hit **Generate** to generate the different files used by the software, see **Figure 9**.

   NOTE: Depending on the Synthesis tool used, the PLL frequencies might need to be adjusted to compile the design.

Figure 9: Generating the Default SoC Design

7. Back in **Lattice Propel,** create a software project. From the **File** menu, select **New > Lattice C/C++ Project** and click **Next,** see **Figure 10**.

Figure 10: Creating a New Software Project

8. Name the project **Hello_World** and accept all defaults, click **Next** and **Finish,** see **Figure 11.**
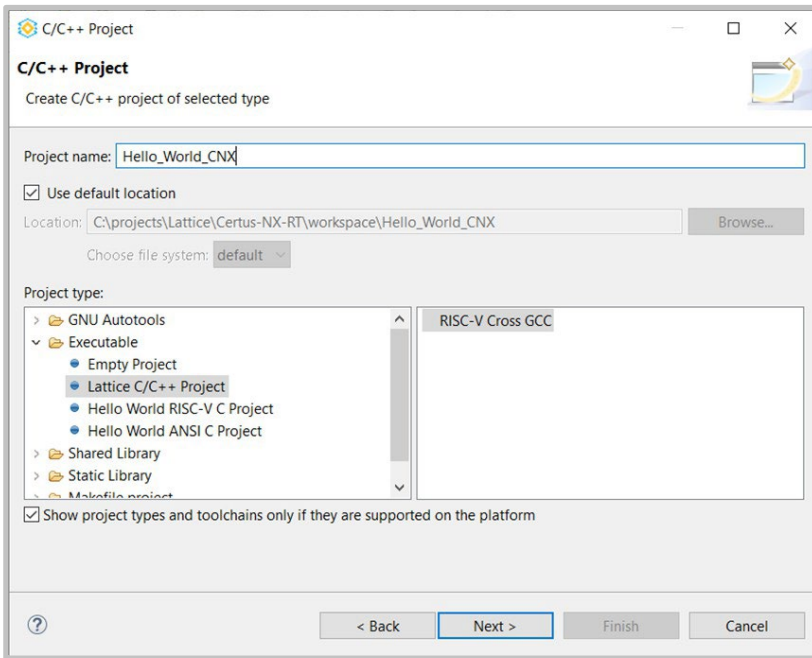
Figure 11: Naming the Project

9. The template creates the source code for the "Hello World" project and the **BSP** based on the **RISC-V SoC** design, see **Figure 12.**
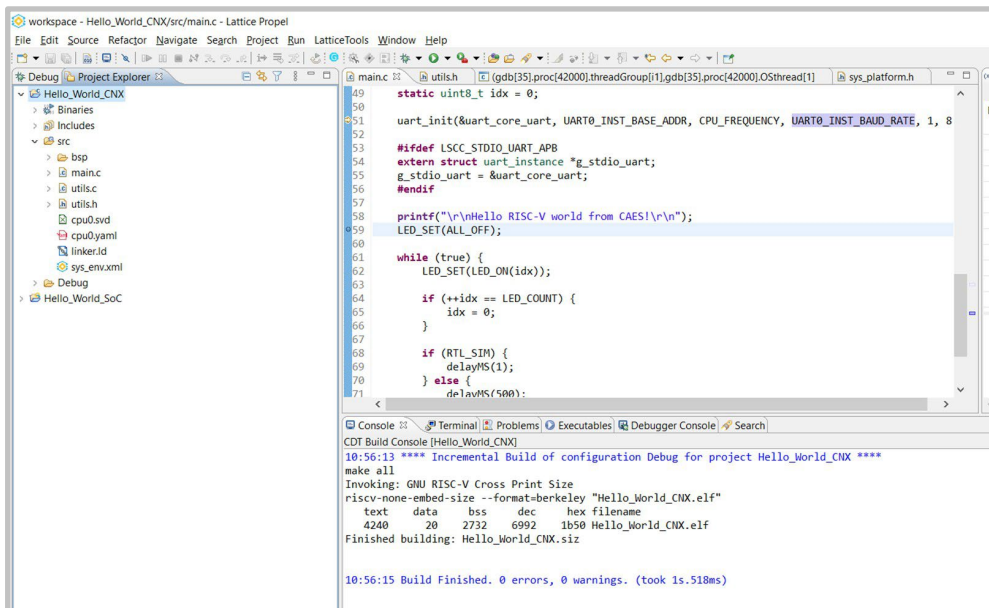

Figure 12: Generated Hello World Project

10. To build the "Hello World" project, right click on **Hello_World_CNX** and select **Build Project.** The compiler builds the project and generates the necessary output files, see **Figure 13.**
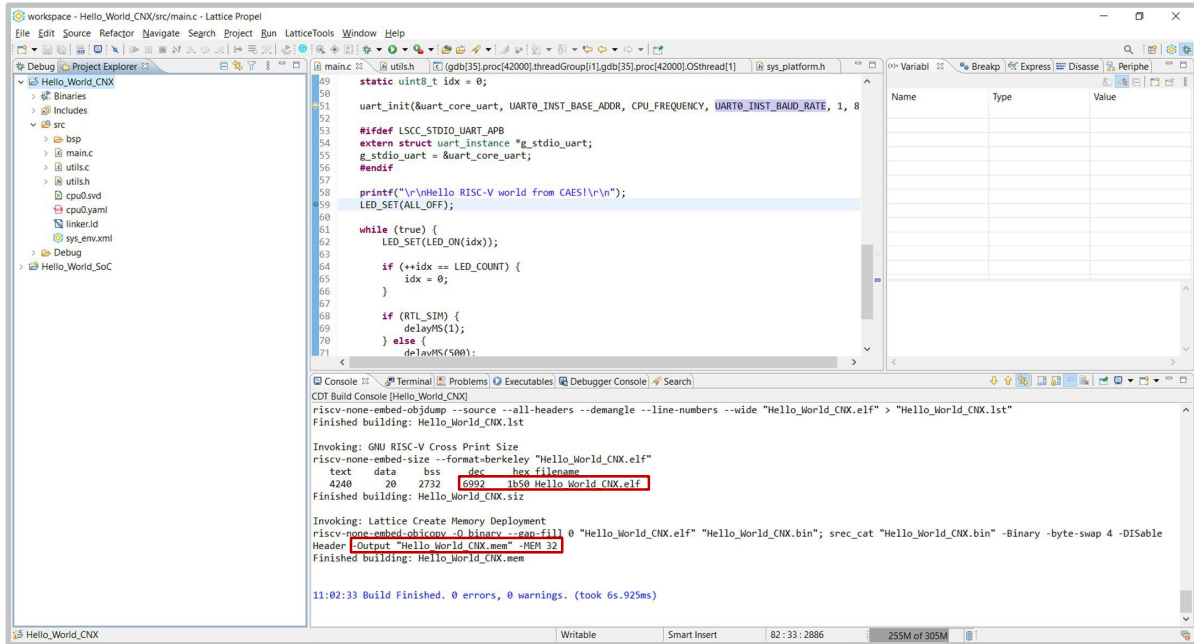
Figure 13: Building the Software Project

11. Back in the **Project Explorer,** highlight **Hello_World_CNX_SoC** and select run **Lattice Radiant, Lattice Radiant Software** opens in a new window, see **Figure 14.**
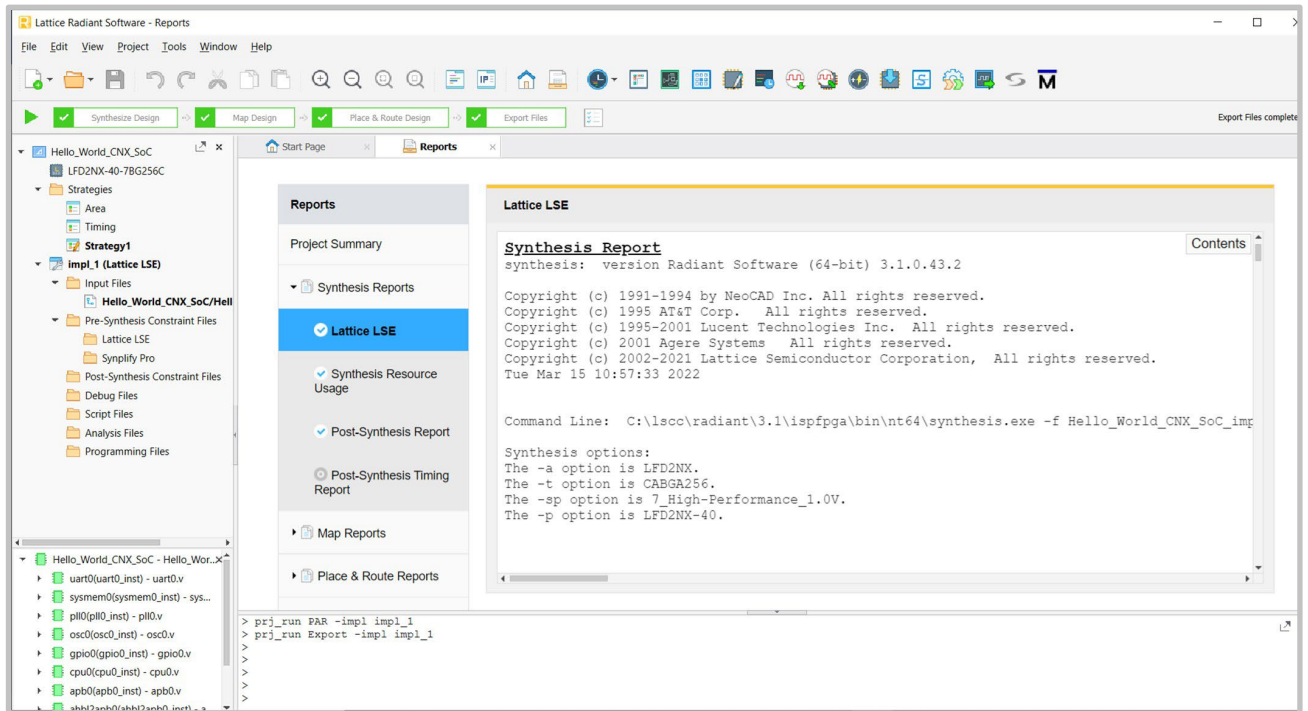

Figure 14: Opening the Project in Lattice Radiant

12. Before compiling the entire design, assign the proper pin number to the signals based on the eval board schematics, see **Figure 15.**
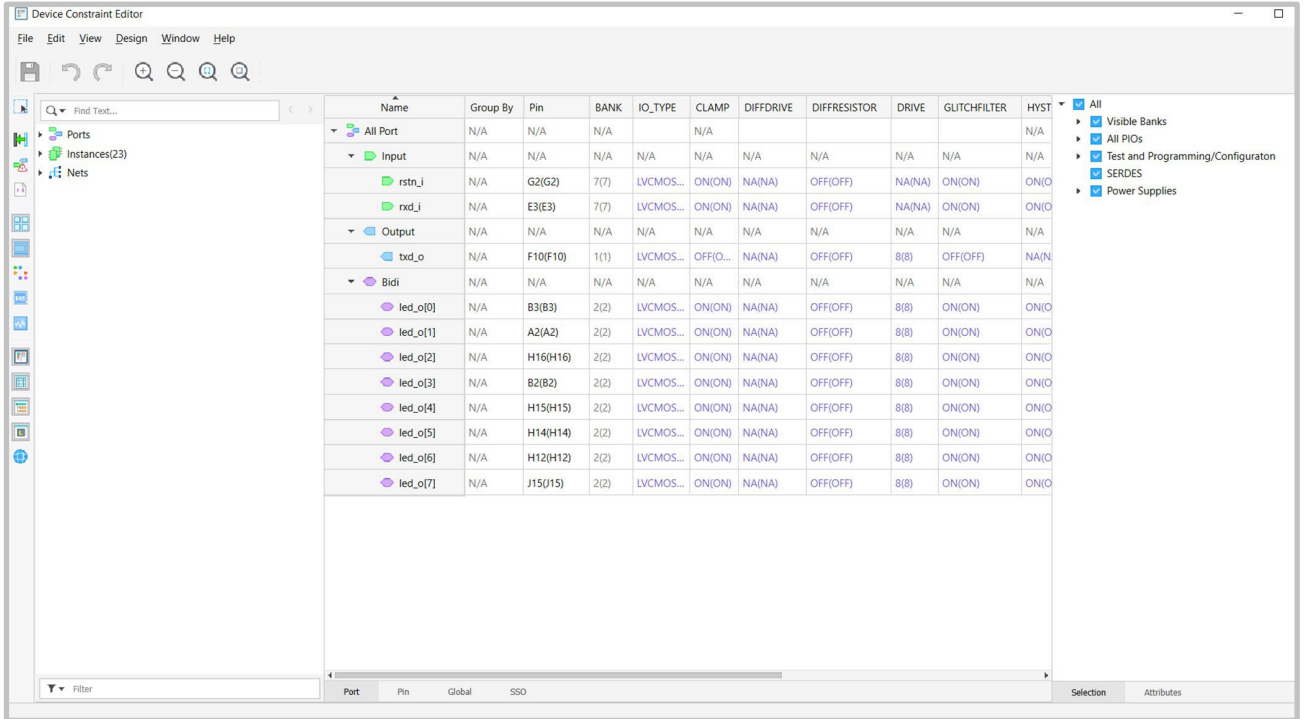

Figure 15: Pin Assignment

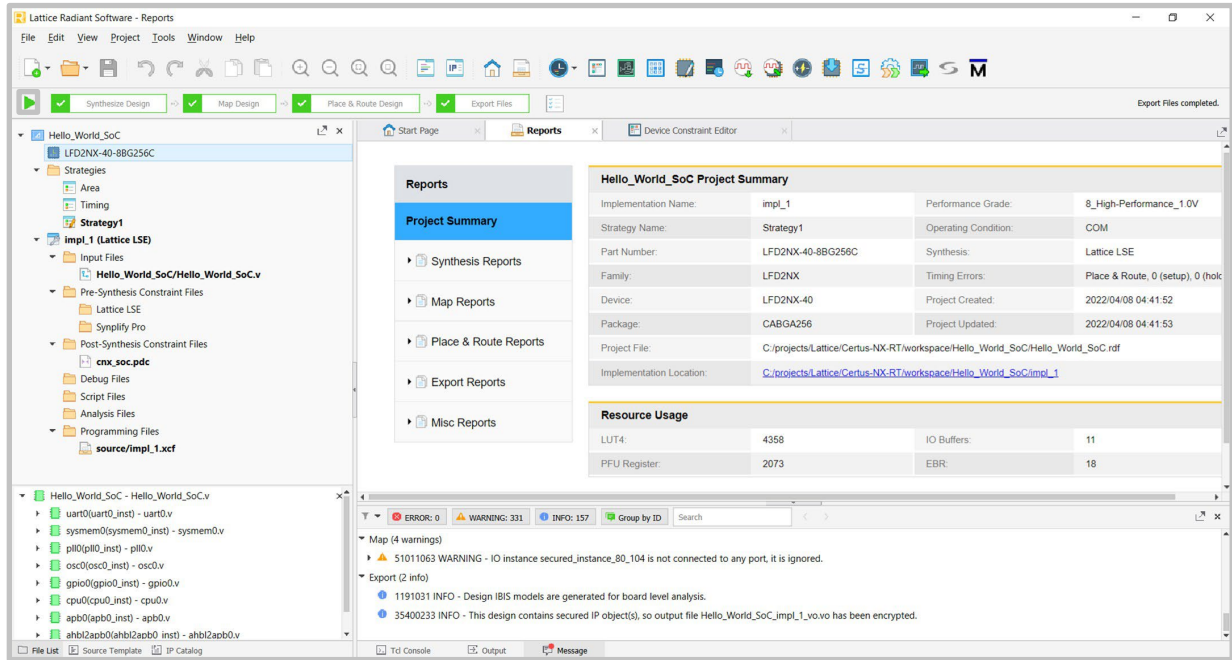**13.** Compile the design by clicking on the **Run All** arrow, see **Figure 16.**

Figure 16: Compiling the Lattice Radiant Design

5.0 Program the Device with Radiant Programmer

From the **Tools** menu, select **Programmer** and a new window opens.

Once the cable is detected and settings are set correctly, program the device by clicking the **Program Device** icon, see **Figure 17.**
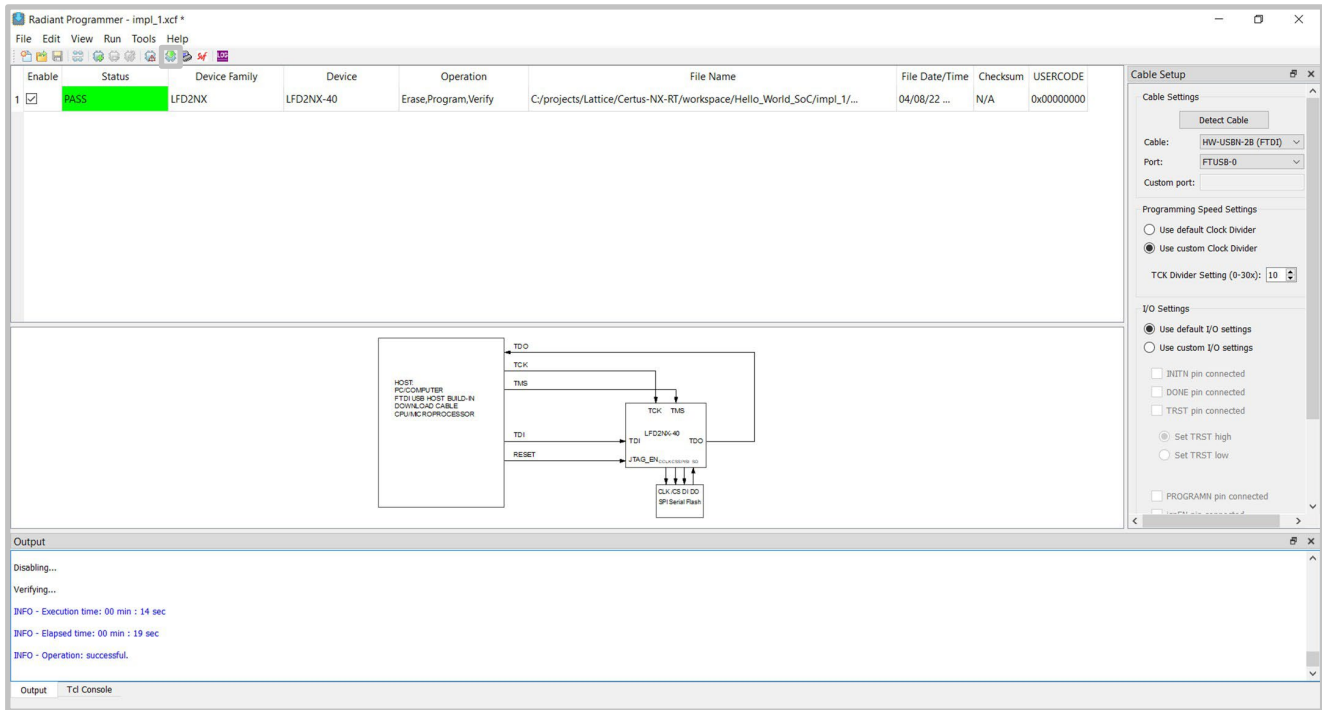

Figure 17: Programming the FPGA

Back in **Lattice Propel,** right click on the "Hello World" project and choose **Debug Configuration;** leave the default configurations and click debug, see **Figure 18.**
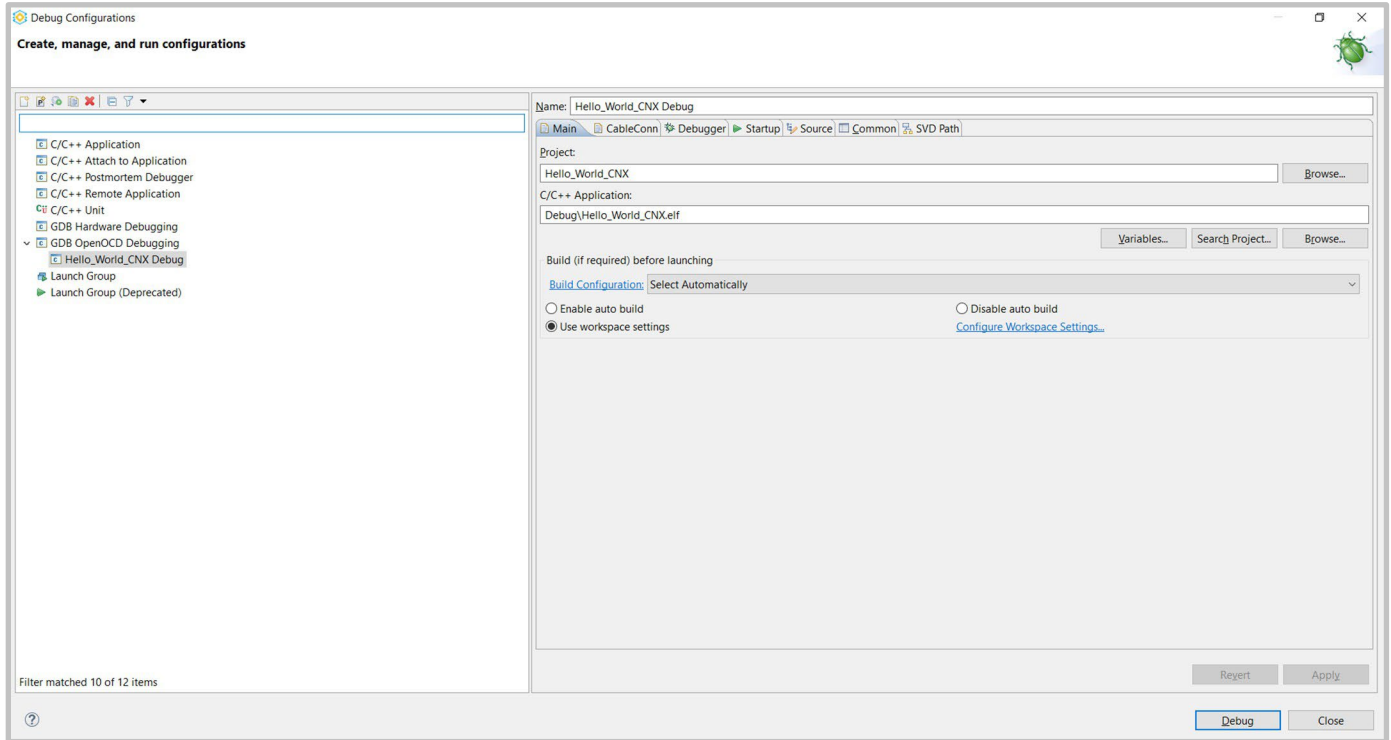

Figure 18: Debug Configurations

The debugger opens the Debug perspective; set a breakpoint at line 59 to print "Hello RISC-V world from Frontgrade!" message on Tera Term terminal, see **Figure 19.** Variable values can be seen in the **Variables** window. As with any other **IDE** running **C** code, you can **Step Into** , **Step Over** and **Step Out** .
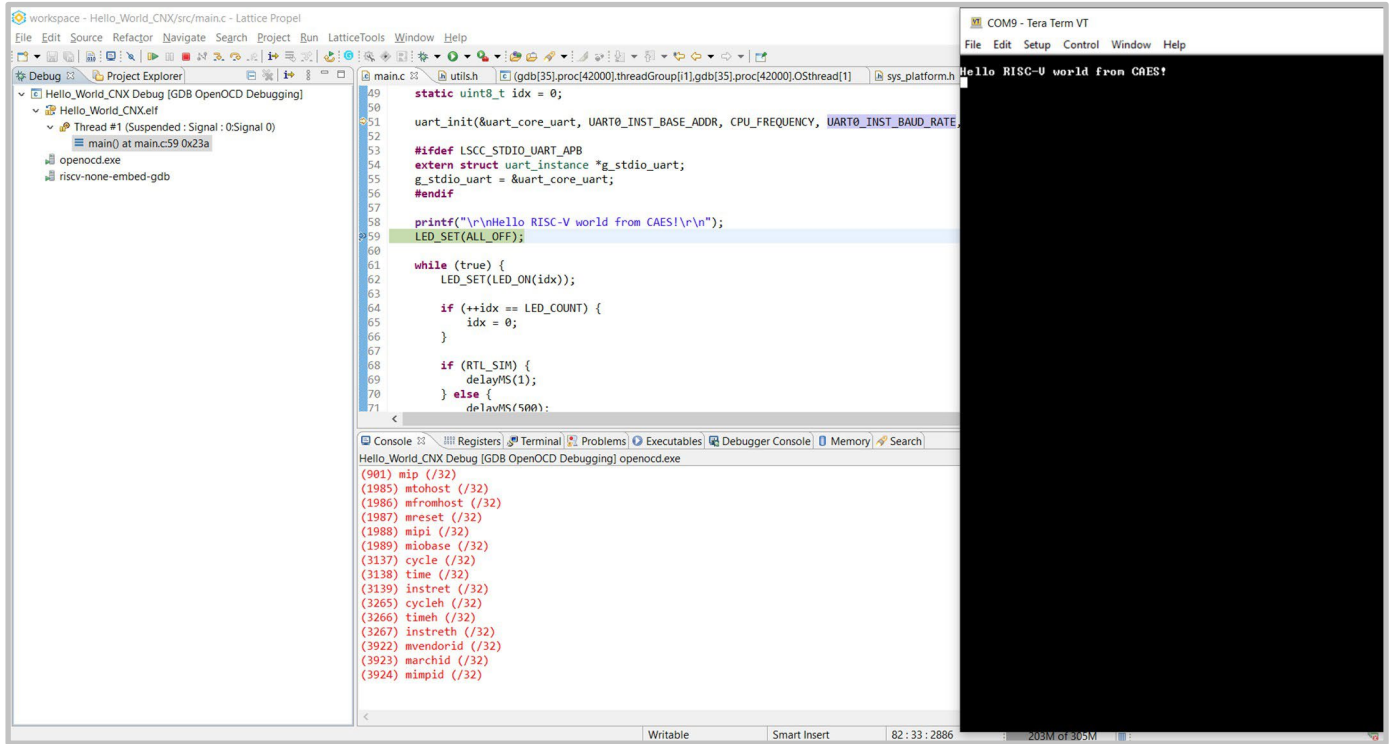

Figure 19: Debug Perspective

Revision History

Date	Revision #	Author	Change Description	Page #
4/8/2022	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.