*Tandem PCIe Configuration Reference Design Tutorial*

March 1, 2020

Leon Yang

This tutorial was validated with VIVADO 2019.2. Minor procedural differences might be required when using later releases.

# Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Revision** |
| 03/01/2020 | 2020.1 | Initial version |
|  |  |  |
|  |  |  |

Table of Contents

[Revision History 2](#_Toc35467608)

[Introduction 4](#_Toc35467609)

[Tutorial Design Description 4](#_Toc35467610)

[Hardware and Software Requirements 4](#_Toc35467611)

[Locating Tutorial Design Files 5](#_Toc35467612)

[Step 1: Creating a New Project 5](#_Toc35467613)

[Step 2: Creating IP\_1 and example design 8](#_Toc35467614)

[Step 3: Creating IP\_2 and example design 11](#_Toc35467615)

[Step 4: Create the new file for top 13](#_Toc35467616)

[Step 5: Modify the related files 13](#_Toc35467617)

[Step 6: Synthesis the Design 14](#_Toc35467618)

[Step 7: Implement the Design 14](#_Toc35467619)

[Step 8: Adapt to customer project 17](#_Toc35467620)

[Conclusion 17](#_Toc35467621)

# Introduction

This tutorial introduces the Tandem PCIe configuration Reference Design method use the Xilinx® Vivado® Integrated Design Environment (IDE).

The reference design is designed to verify and solve the problems in the existing design, perhaps the problems encountered when adding the tandem configuration function to the existing board. The following limitations exist in the design of this reference design.

1. QUAD corresponding to pciex8 GT high-speed transceiver and CLOCK region corresponding to PCIe core are not in the same rows.

2. The GT high-speed transceiver corresponding to the PCIe core is occupied by other IP, Aurora with eight channels on this design.

3. Bank65 has other input and output signals besides PCIe preset.

The method of this reference design can help to solve similar limitations through modify necessary constraint outside the IP original constraint and can get the respect result.

# Tutorial Design Description

This tutorial is based on the PCIe and Aurora examples. It creates a top-level file, which integrates two example projects together. Through modify the constraints, it can solve the original PLACE DRC errors, and can route successfully.

In this reference design, you will learn about the solution for this conflict in the Vivado® IDE. It will take you through the steps of project creation, the process of problems reproduces and the method how to modify the relation code to get the successful results.

If you encountered the similar problems, you can use a similar solution.

# Hardware and Software Requirements

This tutorial requires that the Vivado Design Suite software 2019.2 release or later is installed. And it installed on Windows 10 operating system. For installation instructions and information, see the *Vivado Design Suite User Guide: Release Notes, Installation, and Licensing* (UG973).

The reference design is tested on the customer board used Vintex UltraScale Plus devices xcvu9p-flgb2104-2-i, it needs to modify the device or constraint then run on the Evaluation Board of Vintex® UltraScale Plus.

# Locating Tutorial Design Files

You can find the design file for this tutorial on the Xilinx website without the DCP and bit files: **Reference Design File.**

# Step 1: Creating a New Project

To create a project, use the New Project wizard to name the project, to add RTL source files and constraints, and to specify the target device.

1. Launch the Vivado IDE:

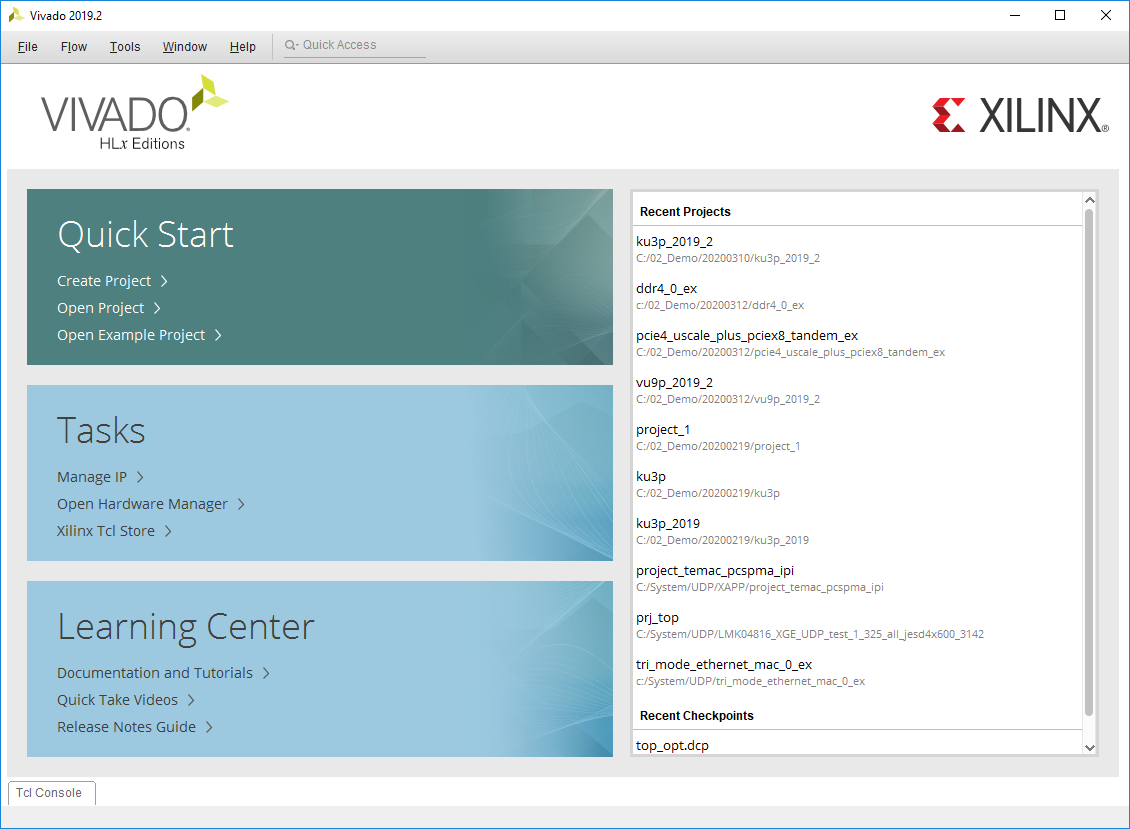


Figure 1: Vivado IDE – Getting Started Page

1. Create New Project to start the New Project wizard

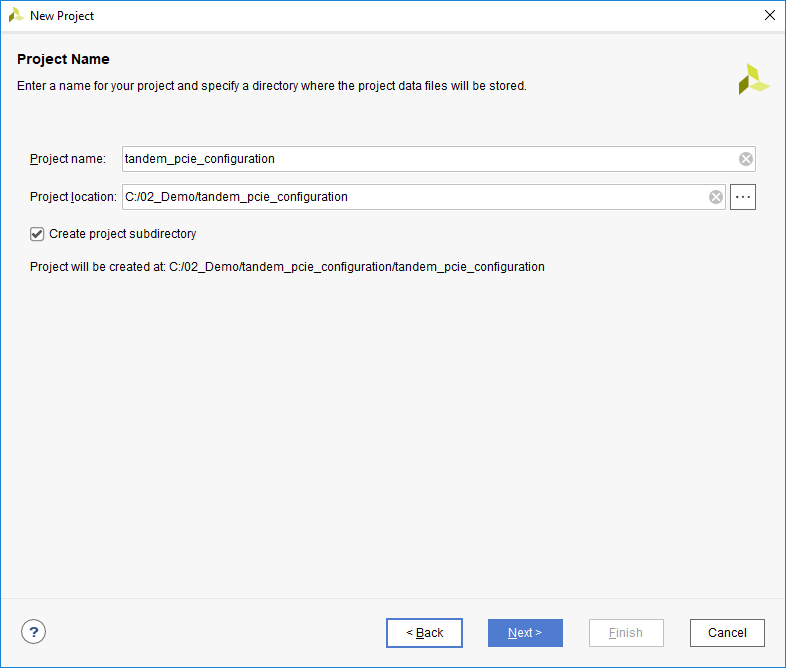


Figure 2: Creating a New Project

1. In the Project Type page, specify the type of project to create as RTL

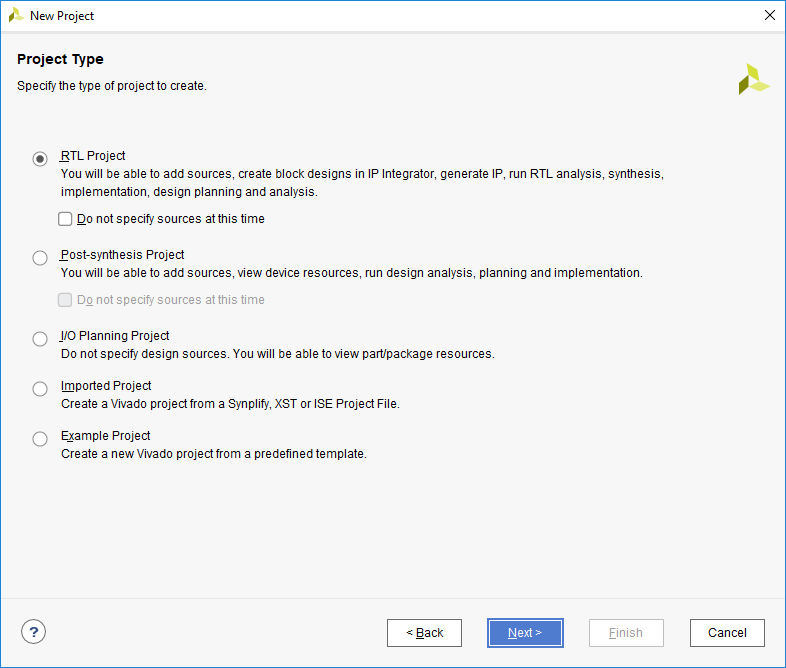


Figure 3: Setting Project Type

1. Don’t add any sources

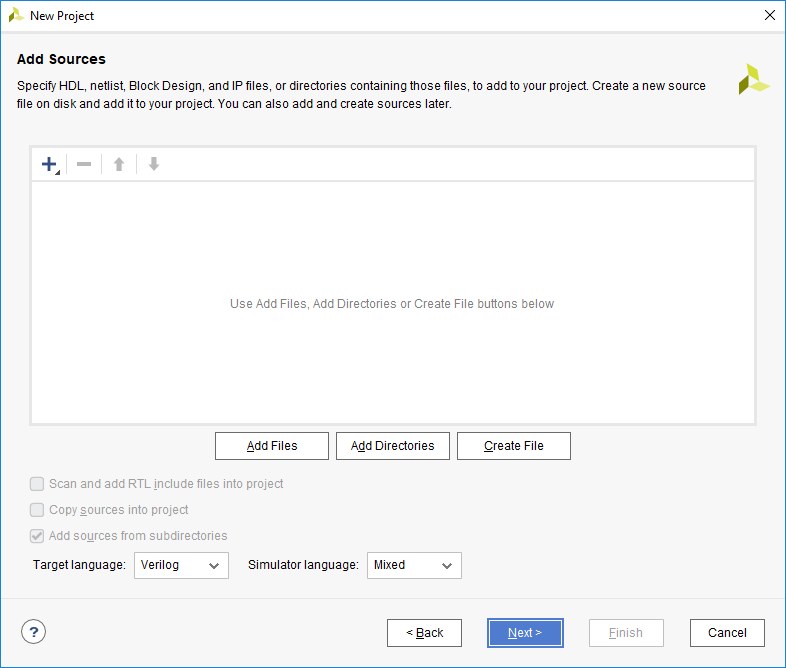


Figure 4: Setting HDL Source Type

1. In the Default Part dialog box, select the devices of xcvu9p-flgb2104-2-i.

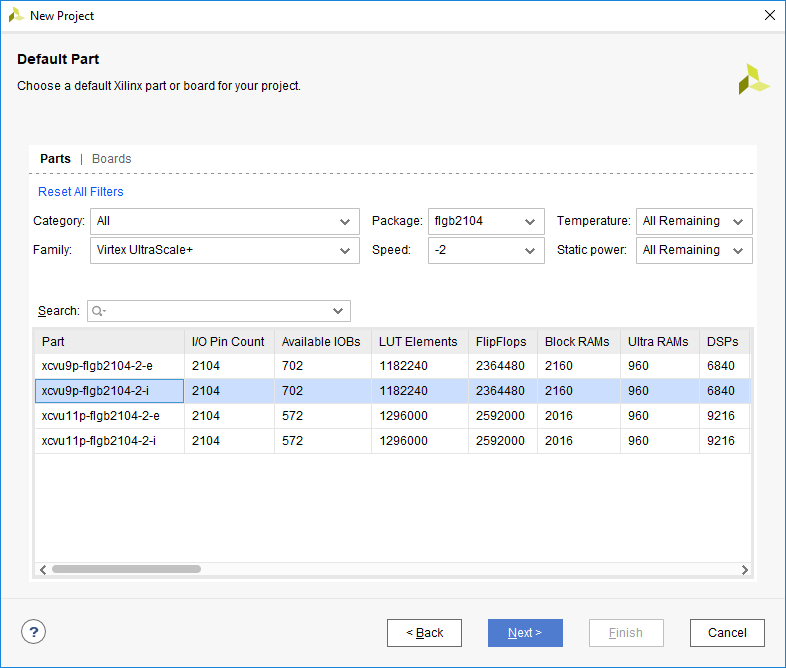


Figure 5: Setting Default Part

1. Review the New Project Summary page. Verify that the data appears as expected, per the steps above, and click Finish.

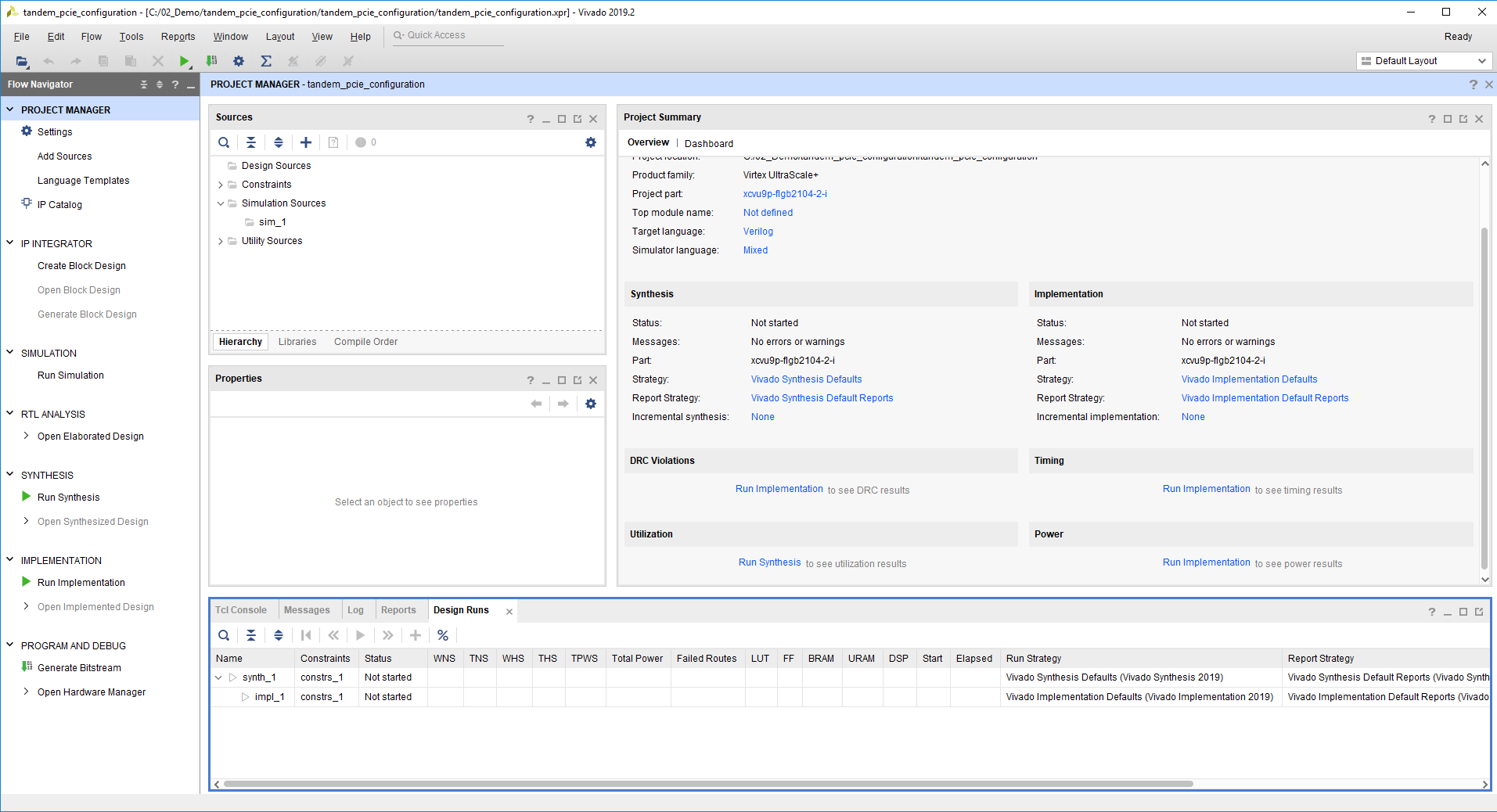


Figure 6: Project in Vivado IDE

# Step 2: Creating IP\_1 and example design

To create the first IP, it’s named pcie4\_uscale\_plus\_tandem is configured as PCIe gen3x8, the PCIe core is X1Y2, the GT located in QUAD 226 and QUAD 227. The configuration worked as tandem PCIe mode. The steps as below.

1. Search the PCIe used IP catalog:

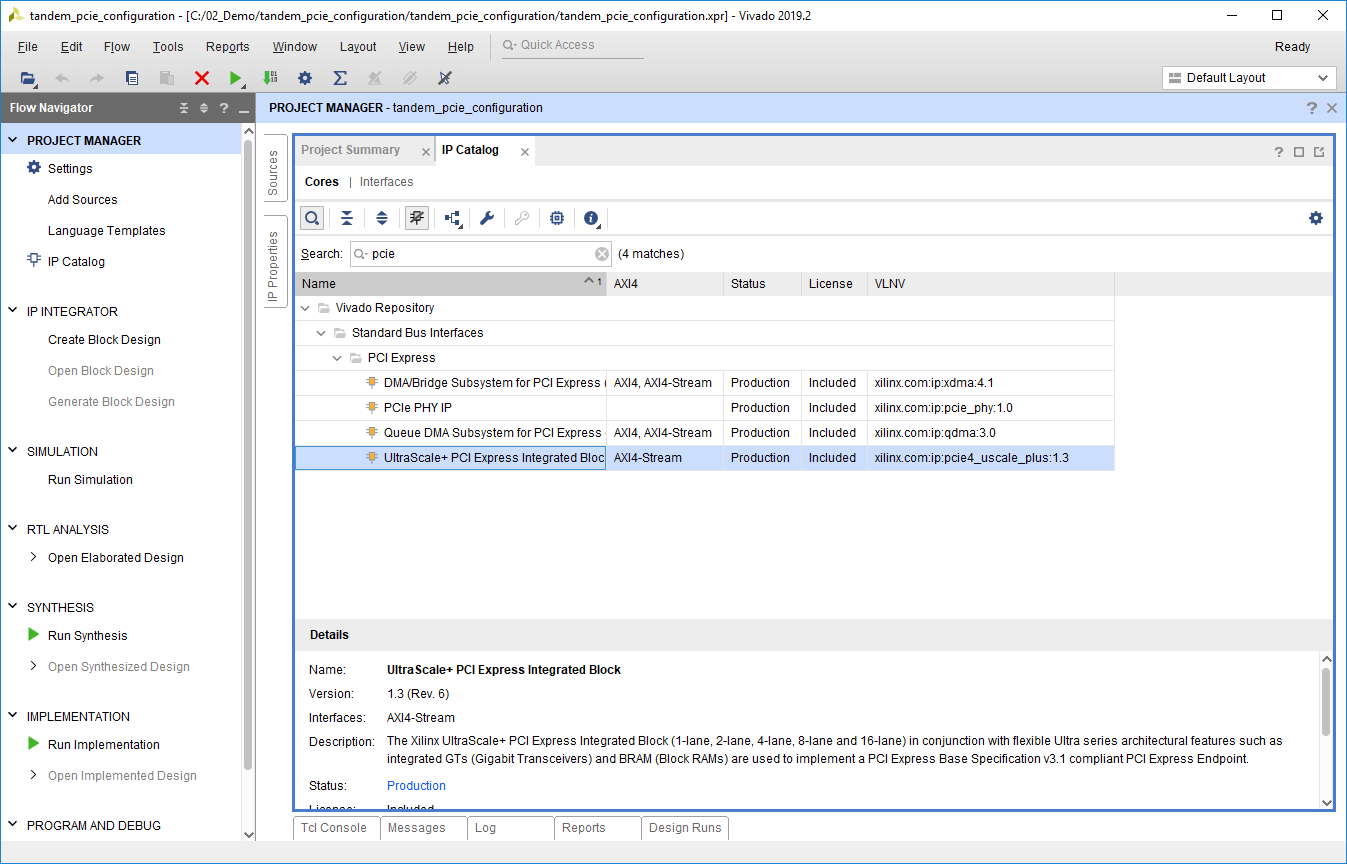


Figure 7: PCIe wizard in IP catalog

1. Customize the IP pcie4\_uscale\_plus\_tandem basic:

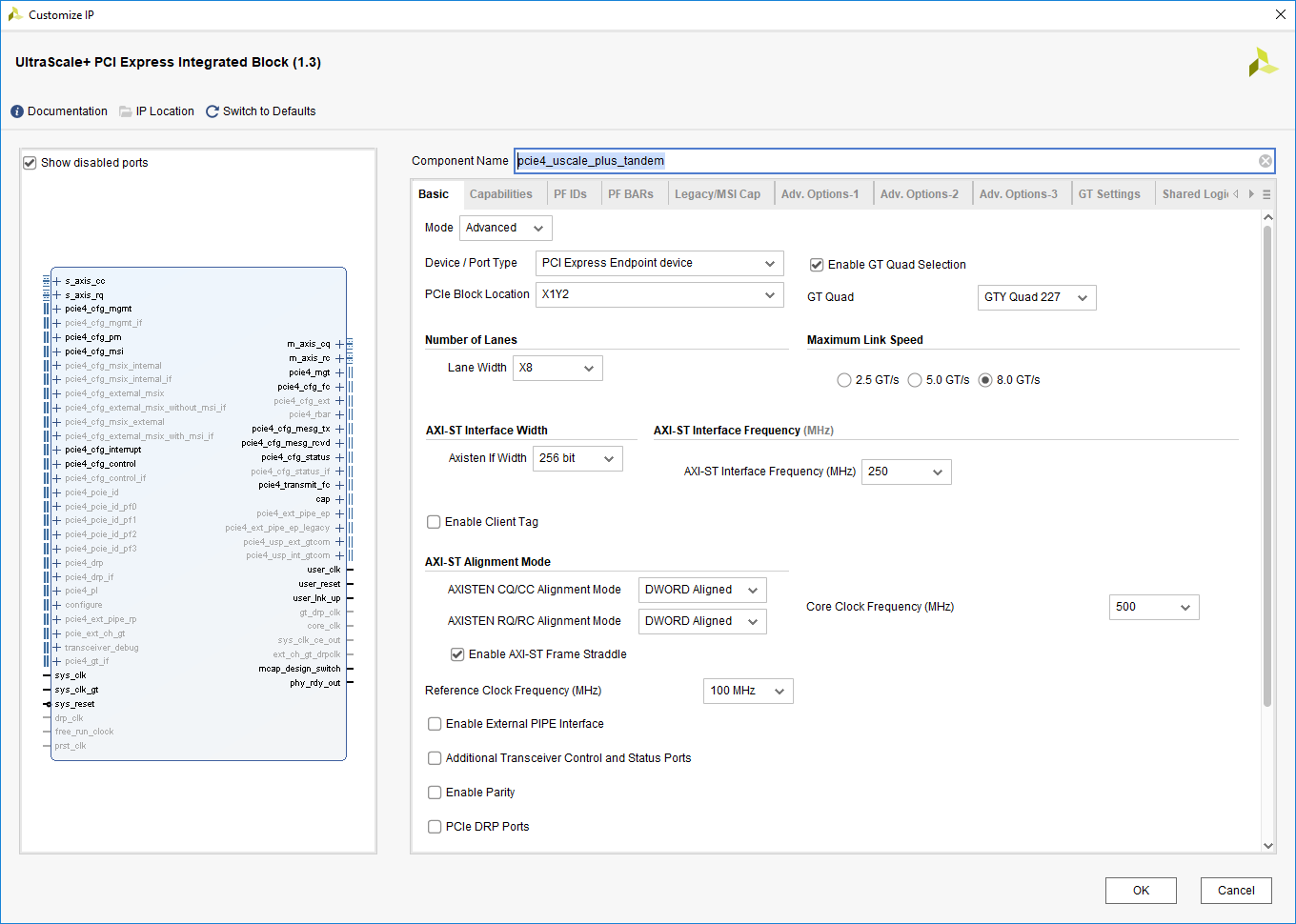


Figure 8: Setting the basic parameters

1. Customize pcie4\_uscale\_plus\_tandem advance:

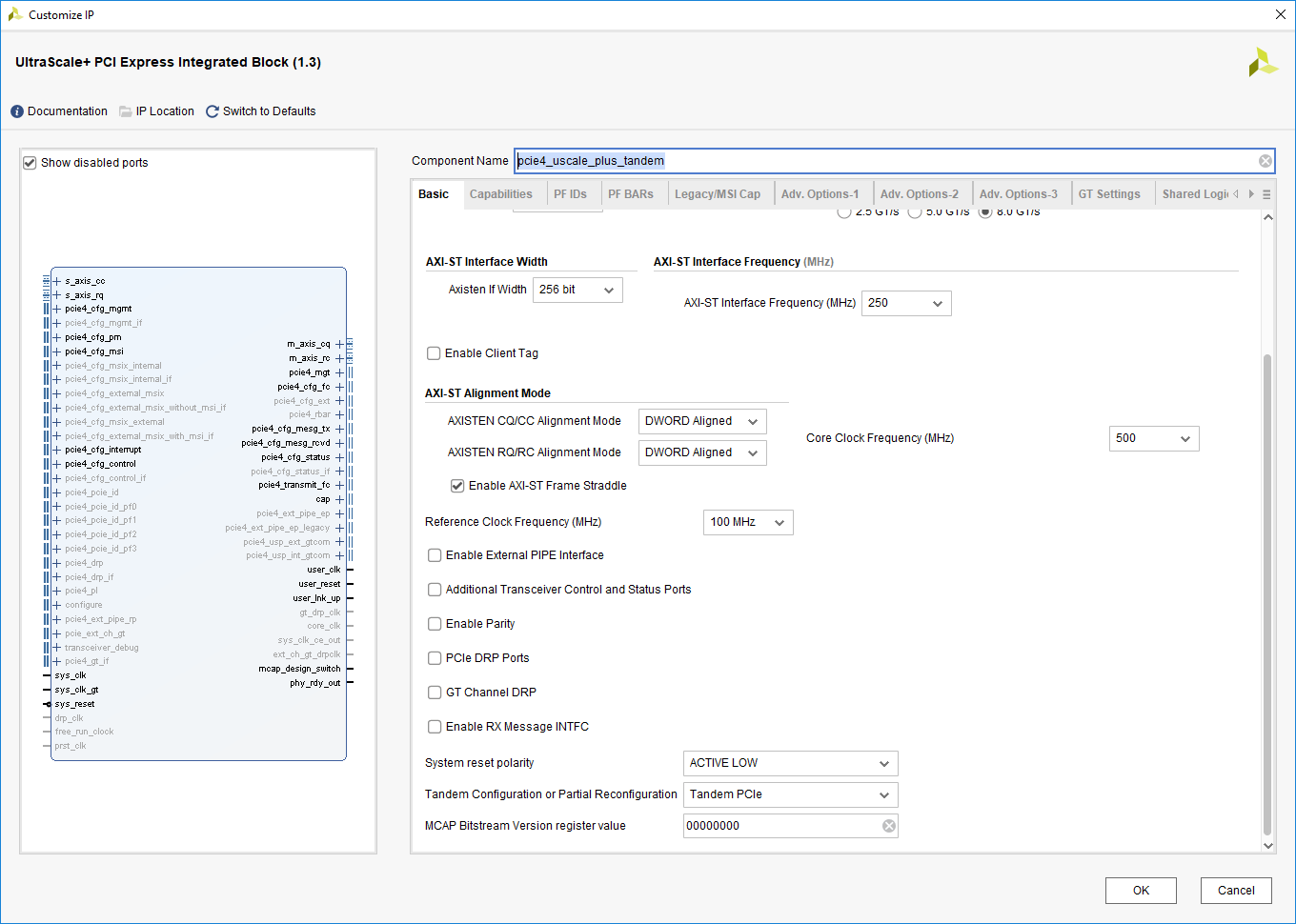


Figure 9: Setting the Tandem Configuration

1. Generate the IP pcie4\_uscale\_plus\_tandem:

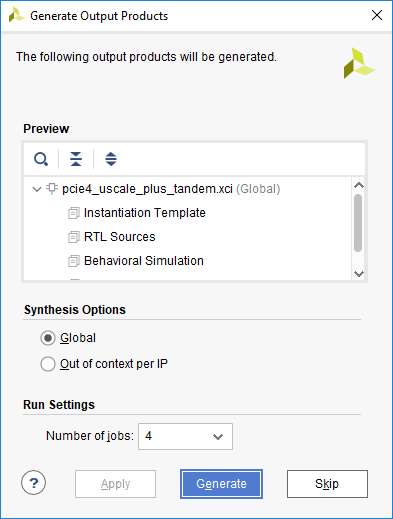


Figure 10: Setting synthesis Options and Generate

1. Open IP Example Design of pcie4\_uscale\_plus\_tandem:

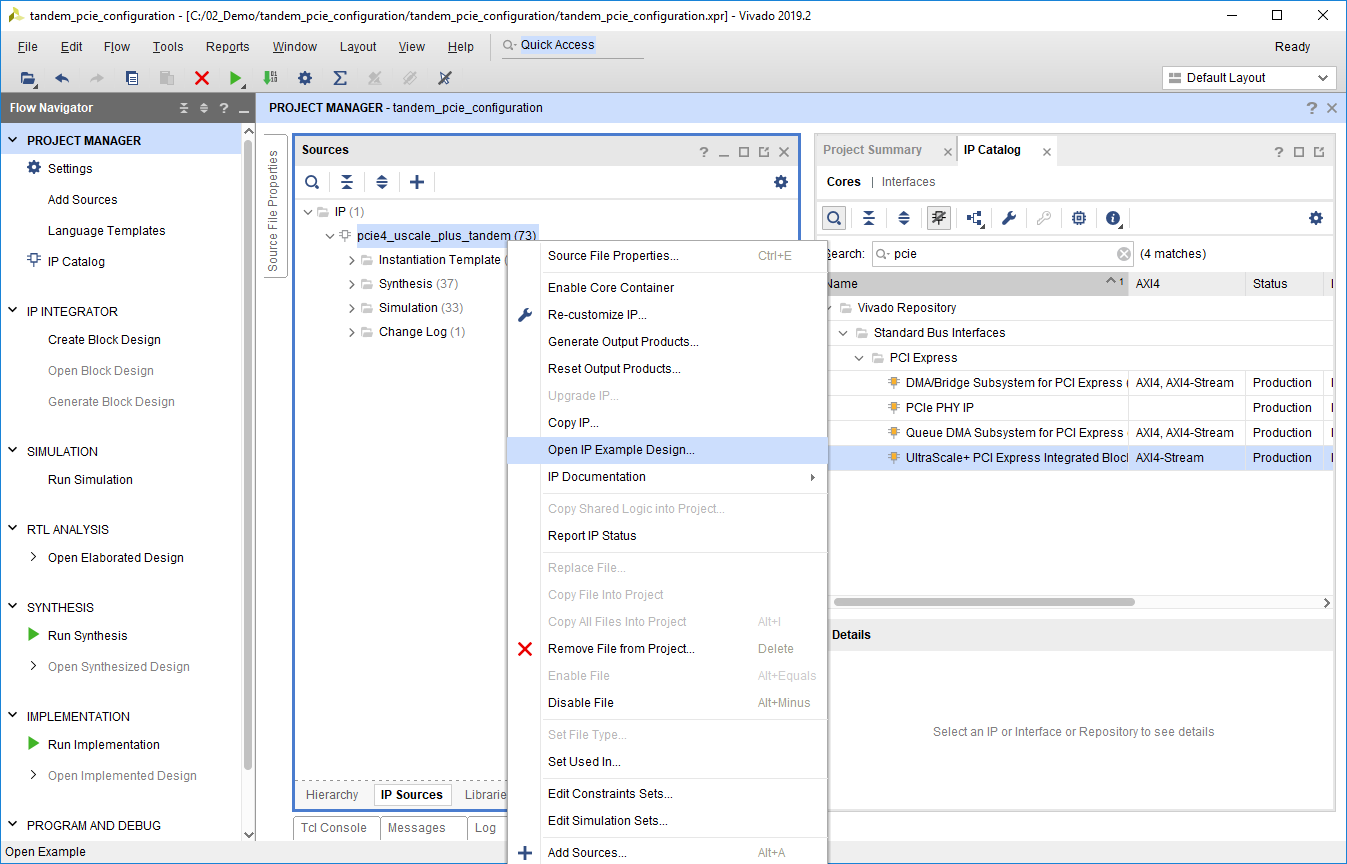


Figure 11: Open IP Example Design

# Step 3: Creating IP\_2 and example design

To create the second IP, it’s named aurora\_64b66b\_x8 is configured as 8 channels, the GT located in QUAD 224 and QUAD 225. The steps as below.

1. Search the Aurora used IP catalog:

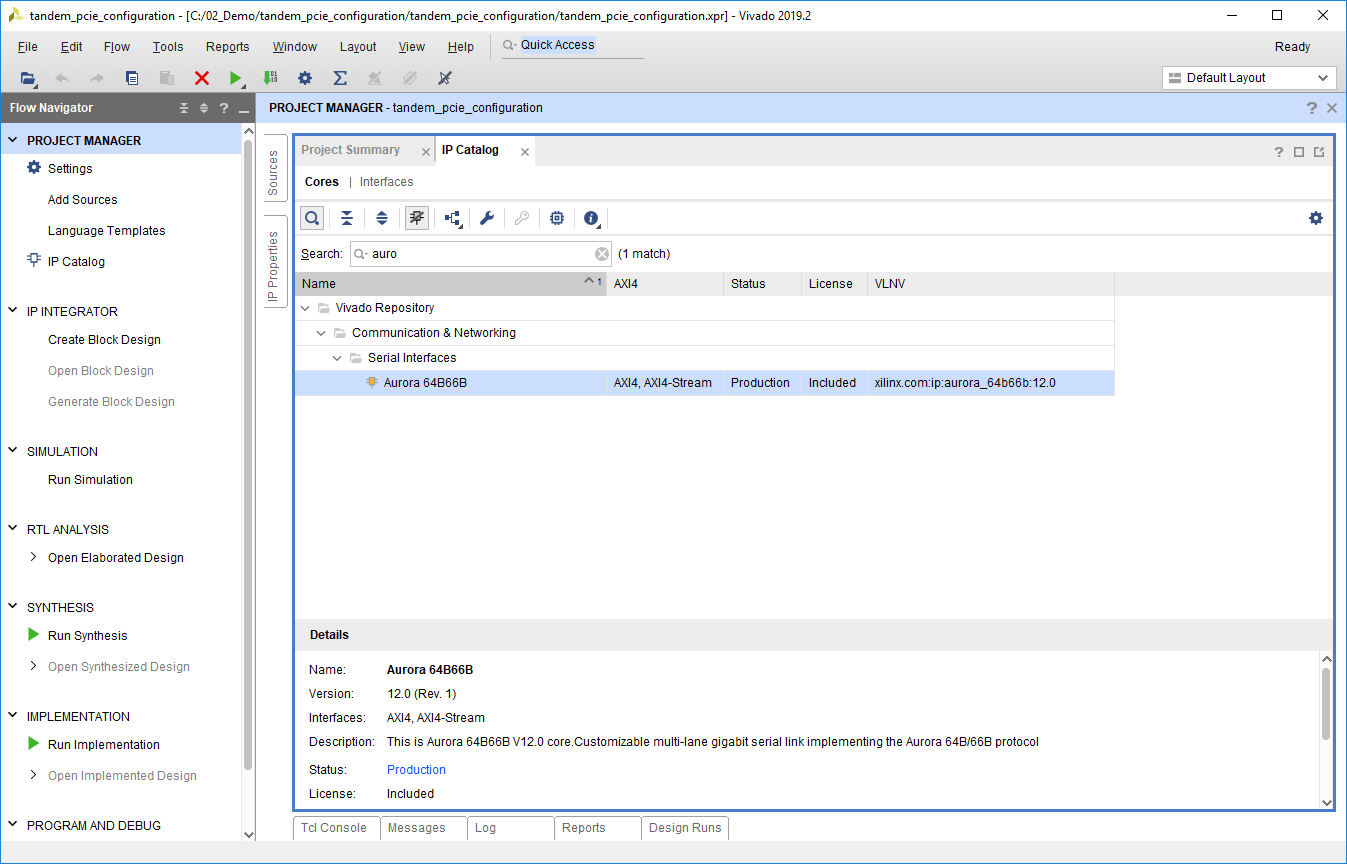


Figure 12: aurora\_64b66b in IP catalog

1. Customize the IP aurora\_64b66b\_x8 basic:

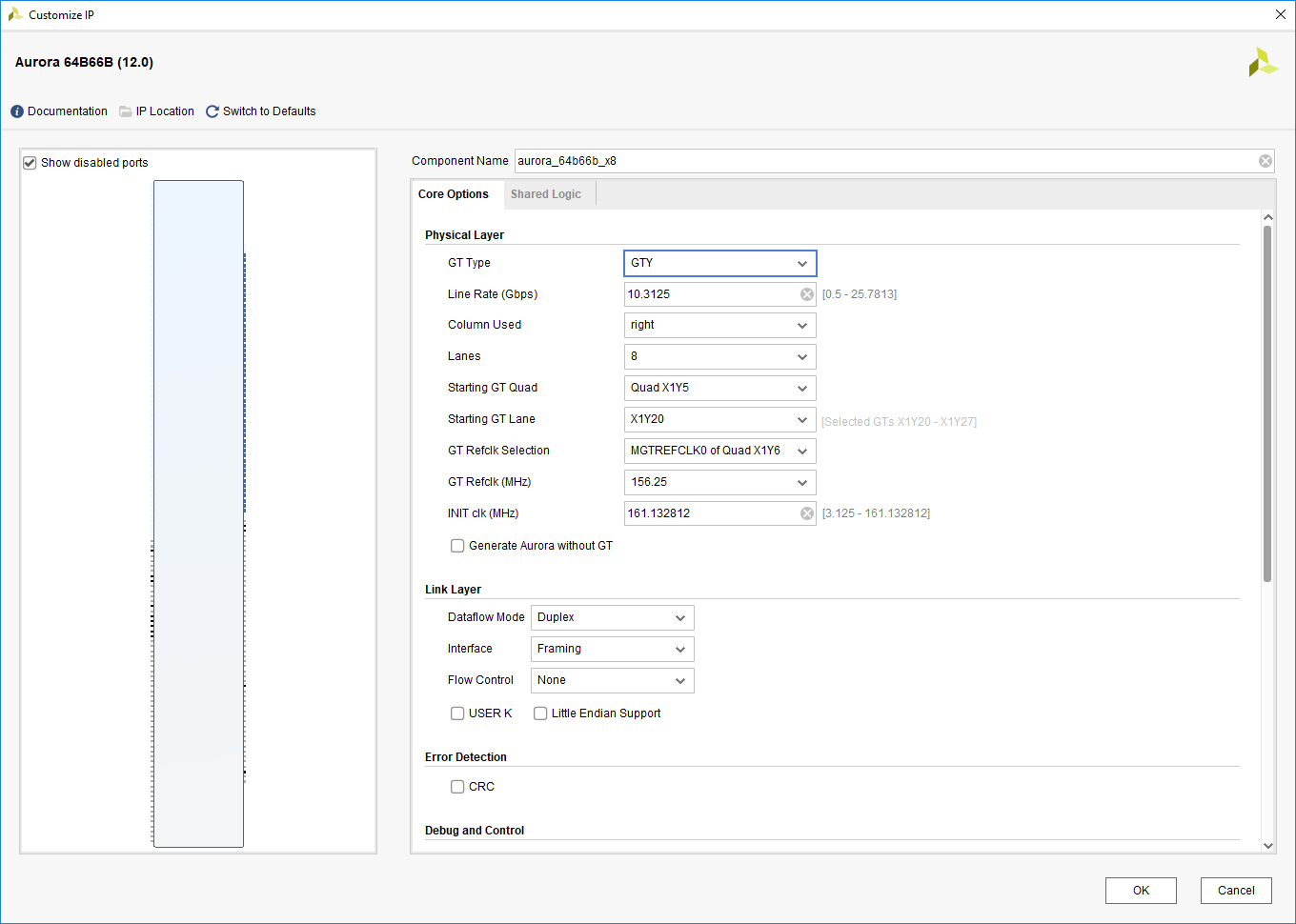


Figure 13: Setting the basic parameters

1. Generate the IP aurora\_64b66b\_x8:

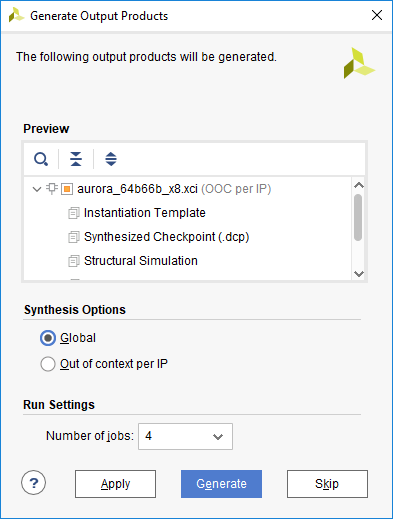


Figure 14: Setting synthesis Options and Generate

1. Open IP Example Design of aurora\_64b66b\_x8:

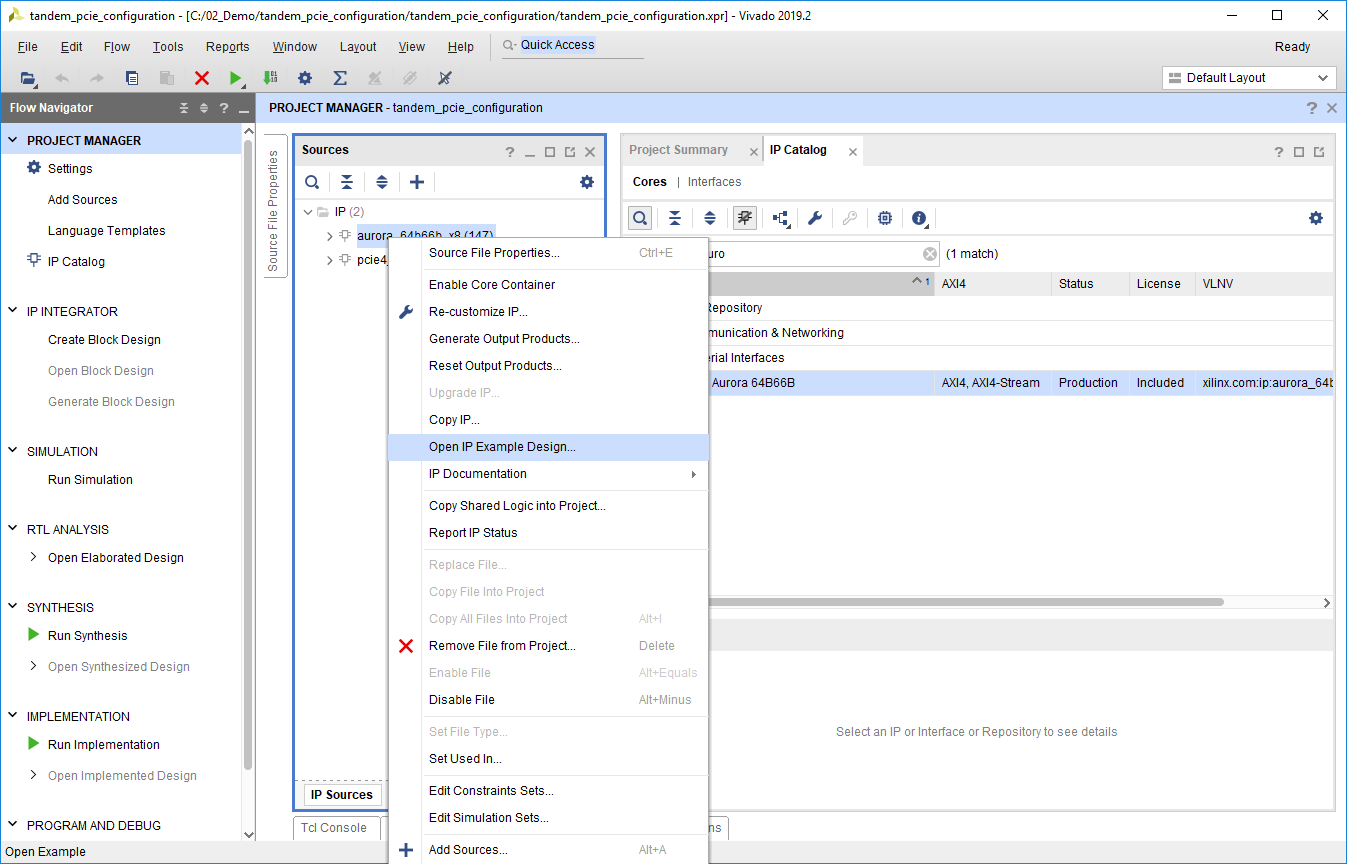


Figure 15: Open IP Example Design

# Step 4: Create the new file for top

To create a top file called top.v, which used to integrate these two IP examples into one project. And it will include two modules, xilinx\_pcie4\_uscale\_ep and aurora\_64b66b\_x8\_exdes.



# Step 5: Modify the related files

To modify the related constraint files to ensure that the pblock have no conflict, and the design can solve PLACE DRC, and can routed successfully. The related files as below.

# Step 6: Synthesis the Design

Synthesis the design focus on constraint, make sure the constraints are complete and the resources are normally.

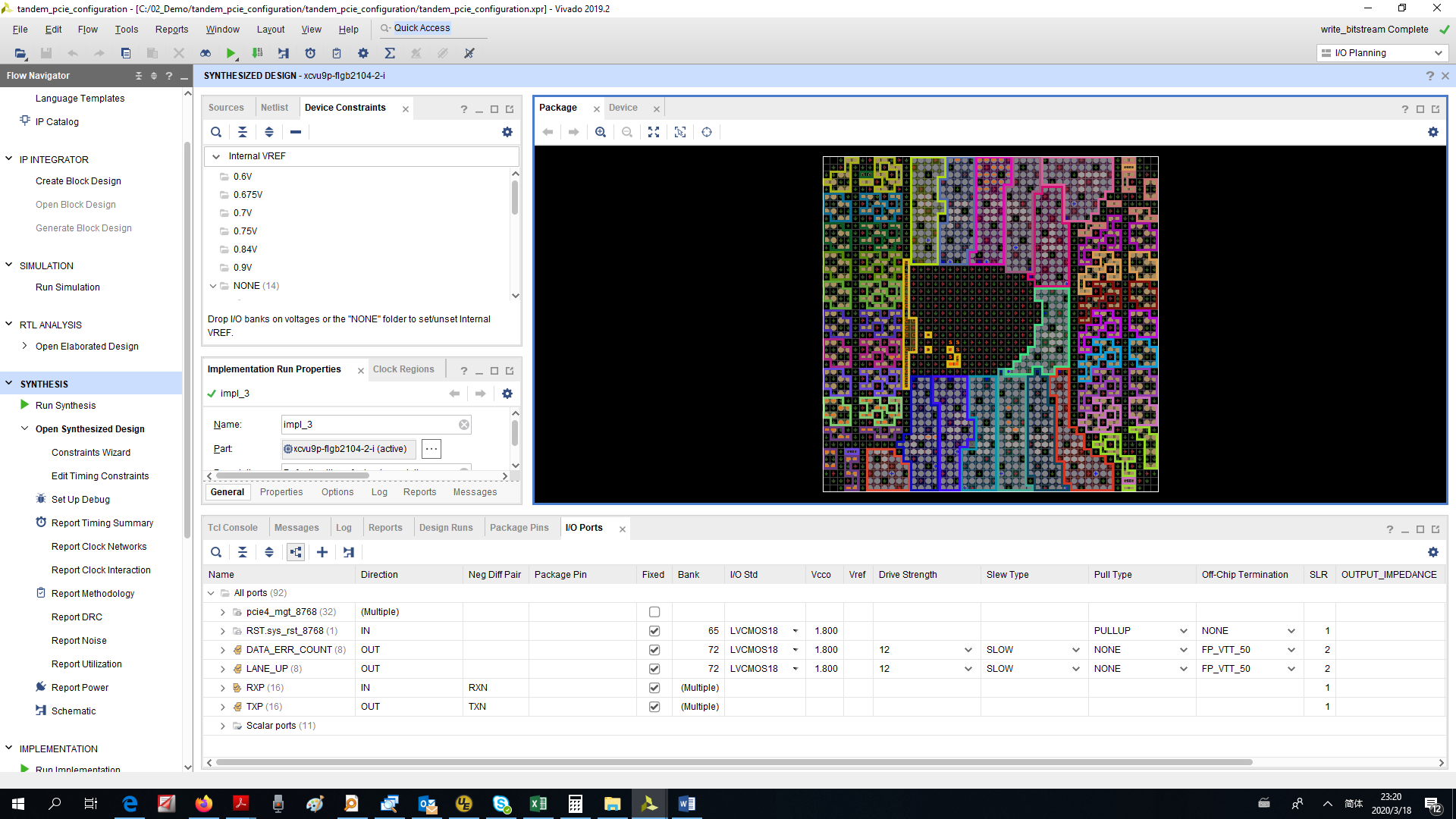


Figure 16: The Synthesized Design

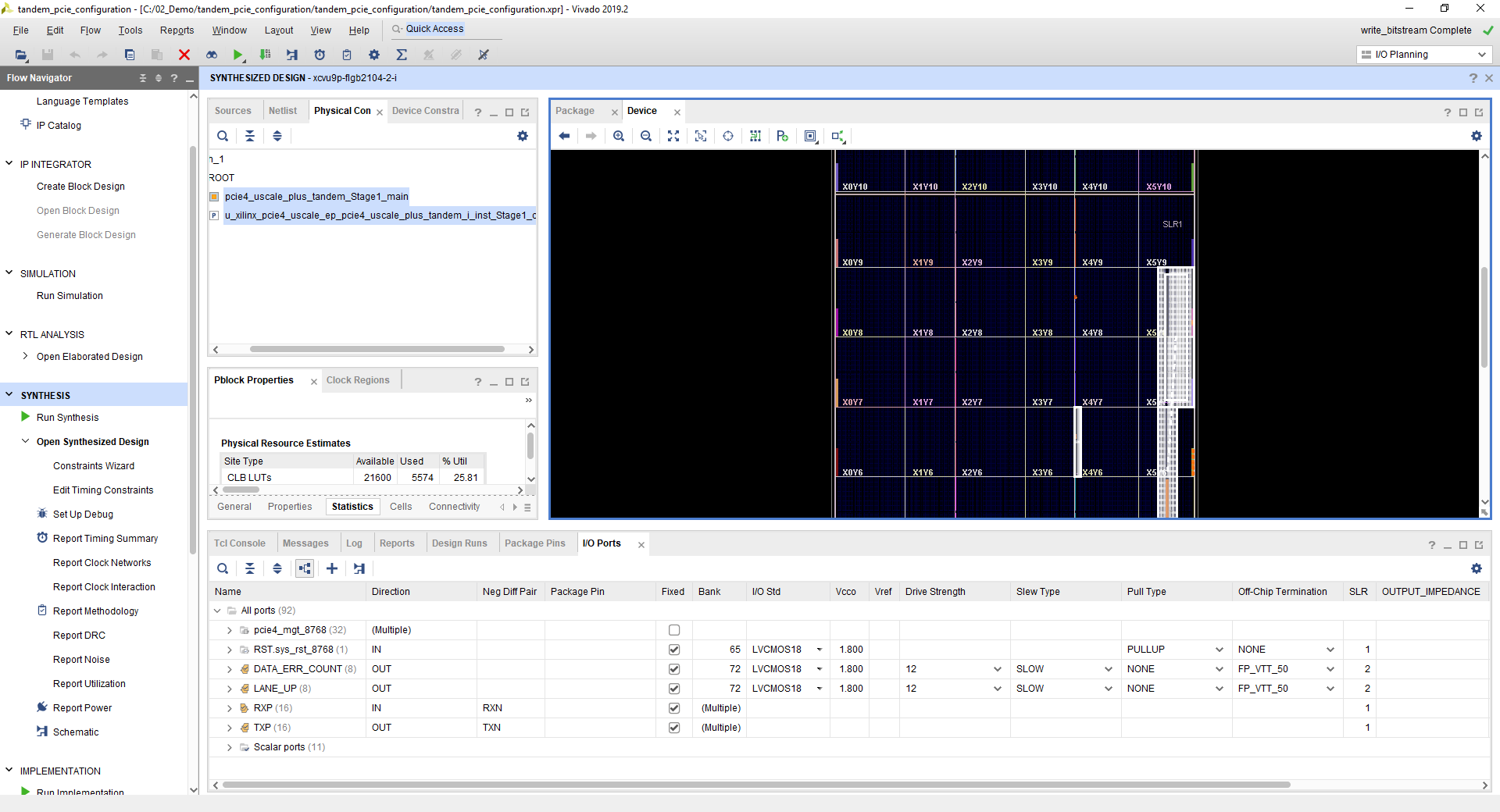


Figure 17: The Synthesized Design

# Step 7: Implement the Design

Implement the design focus on all the report, make sure all of the constraints are valid and no DRC error, the timing can meet the requirement.

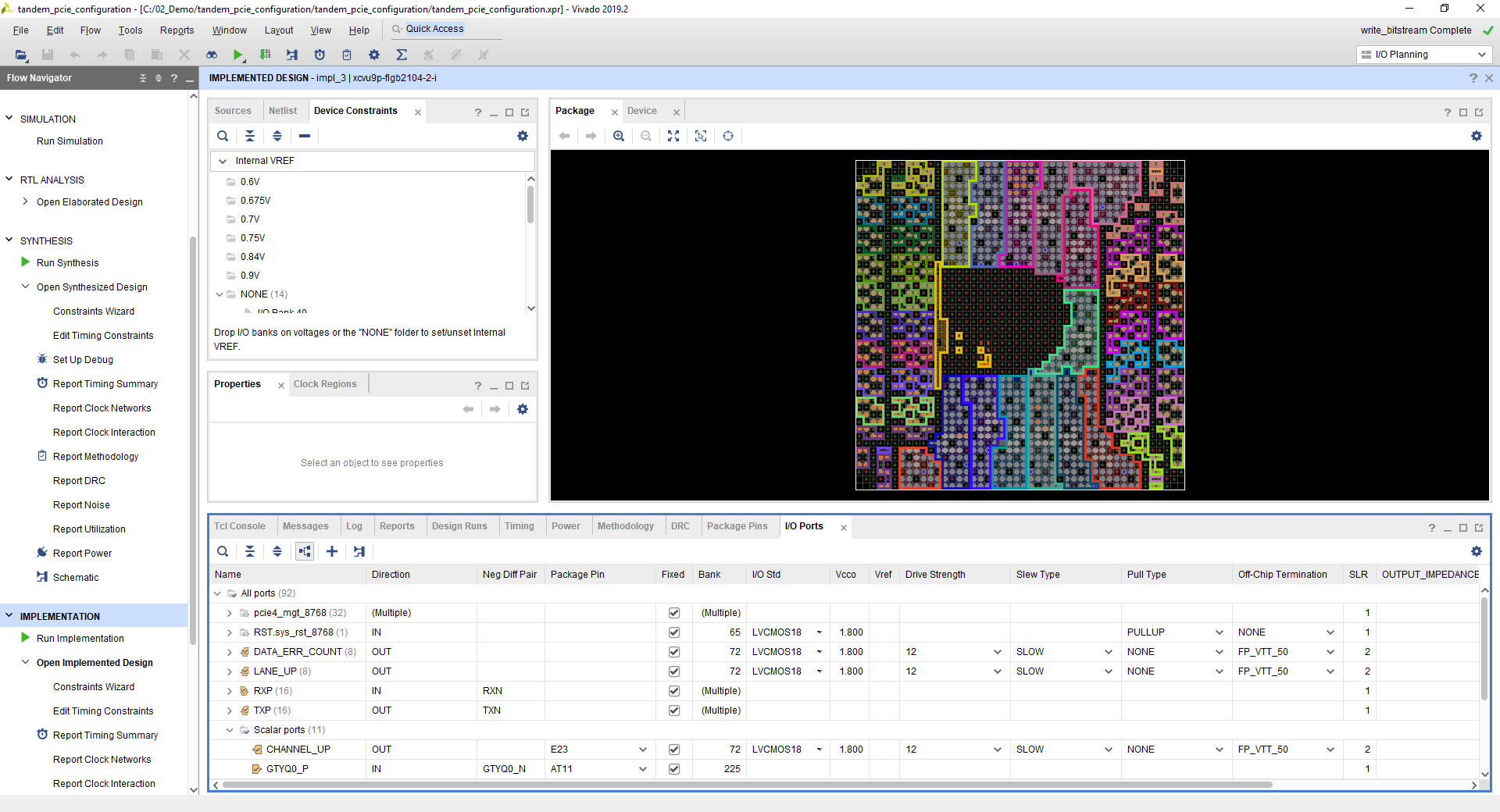


Figure 18: The Implemented Design Package

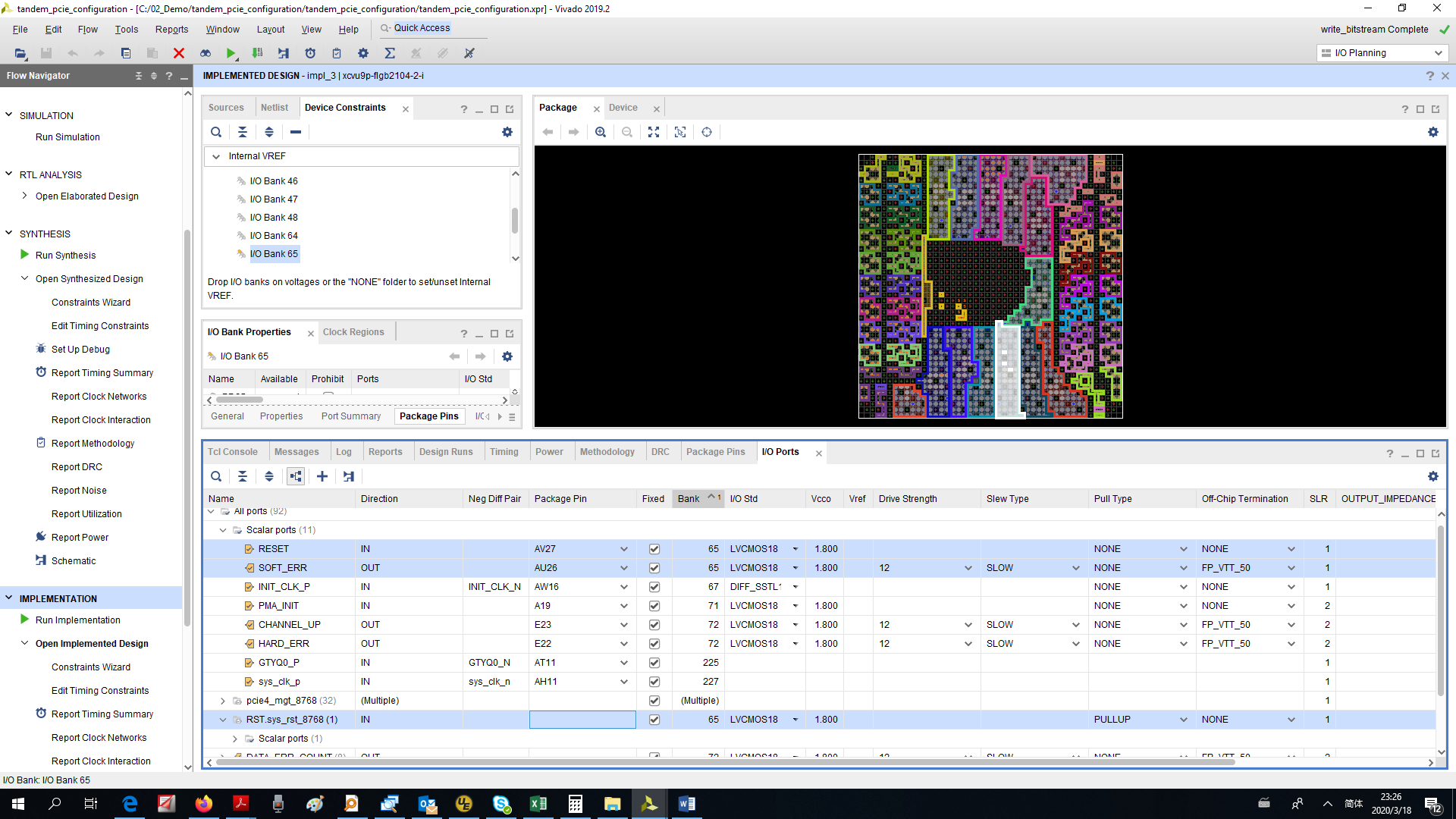


Figure 19: The Implemented Design BANK65

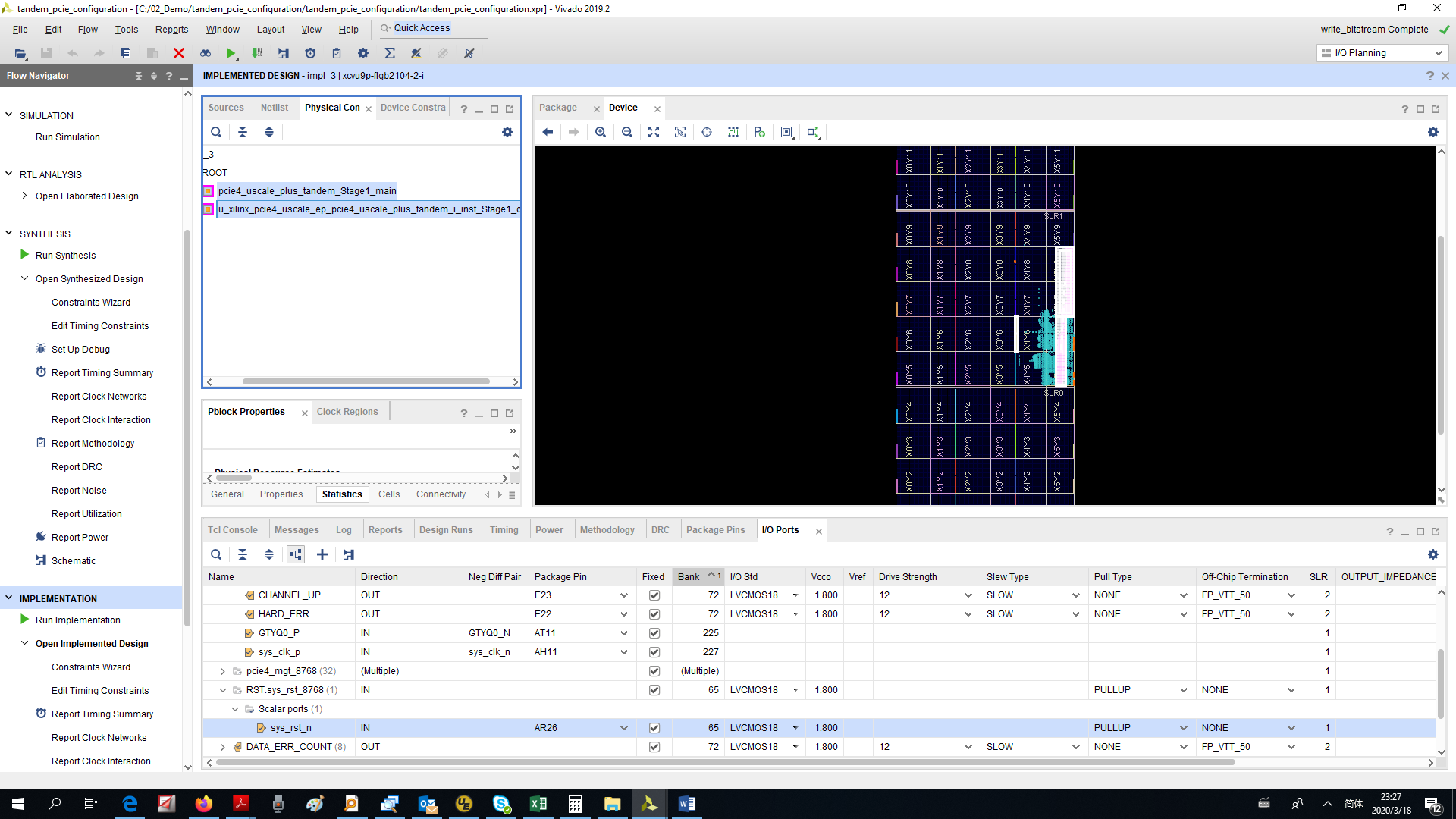


Figure 20: The Implemented Design Stage 1 location

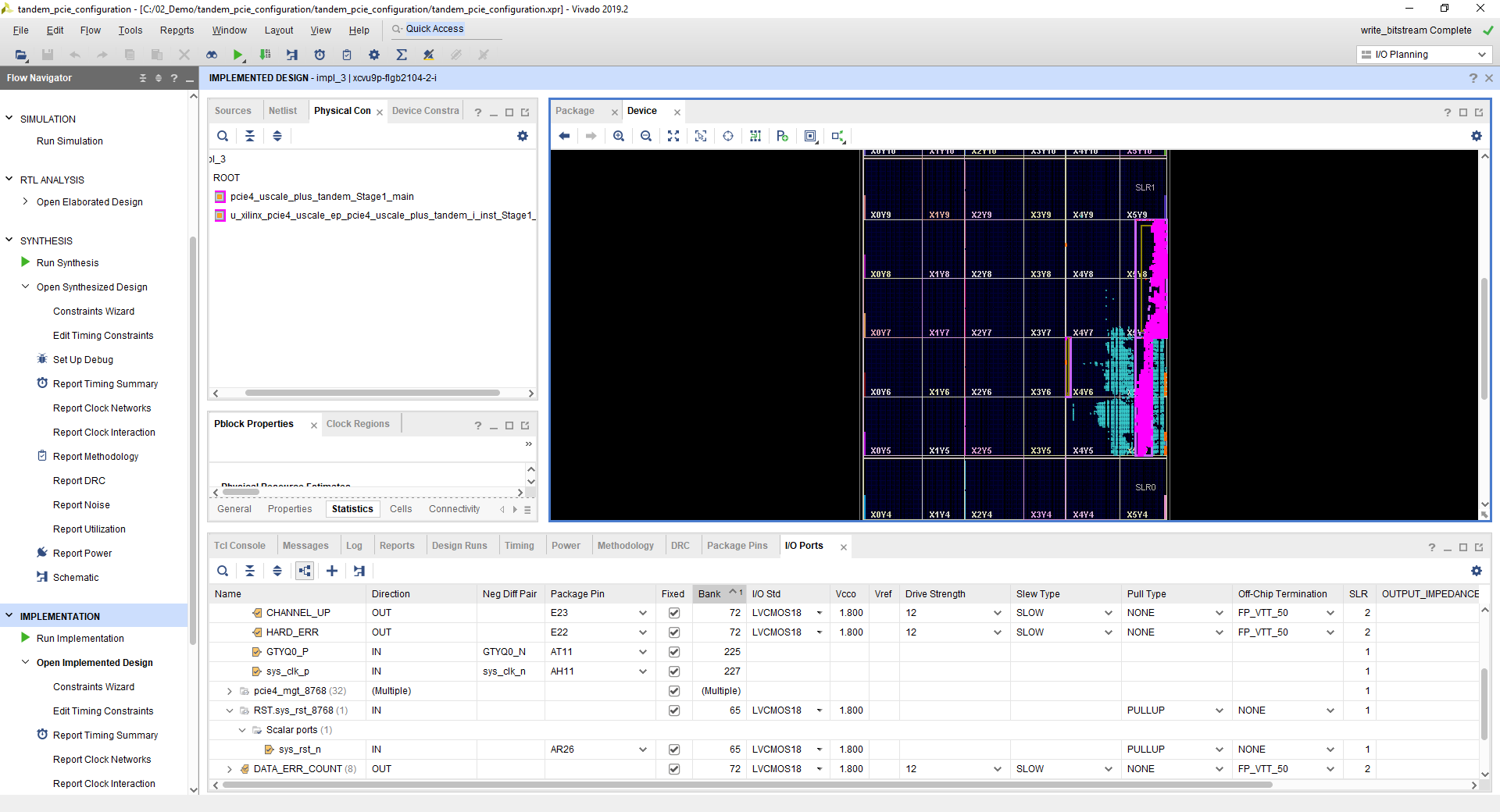


Figure 21: The Implemented Design Stage 1 Leave cells with pink color

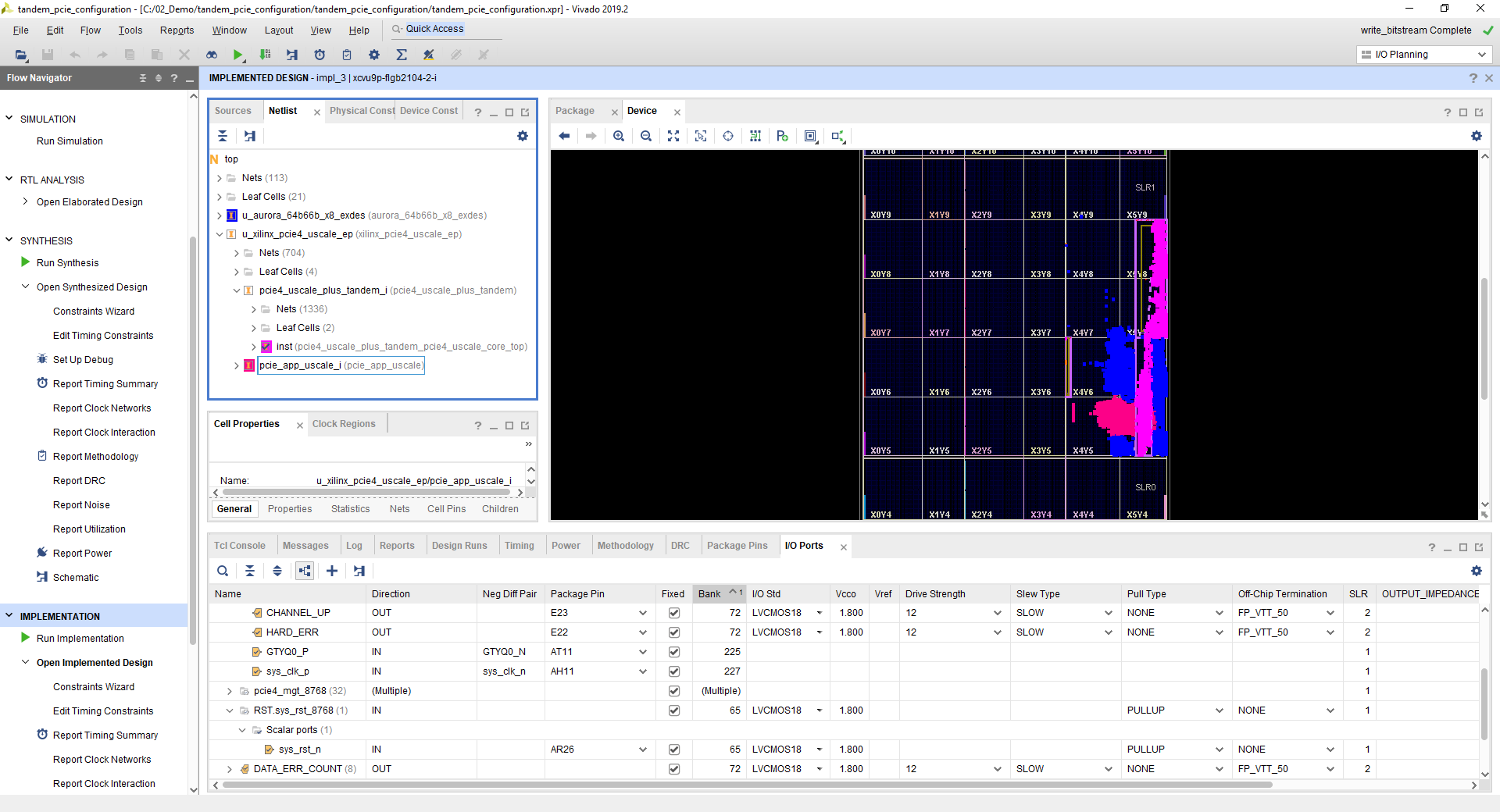


Figure 22: The Implemented Design Aurora (Blue) and PCIE App (Red)

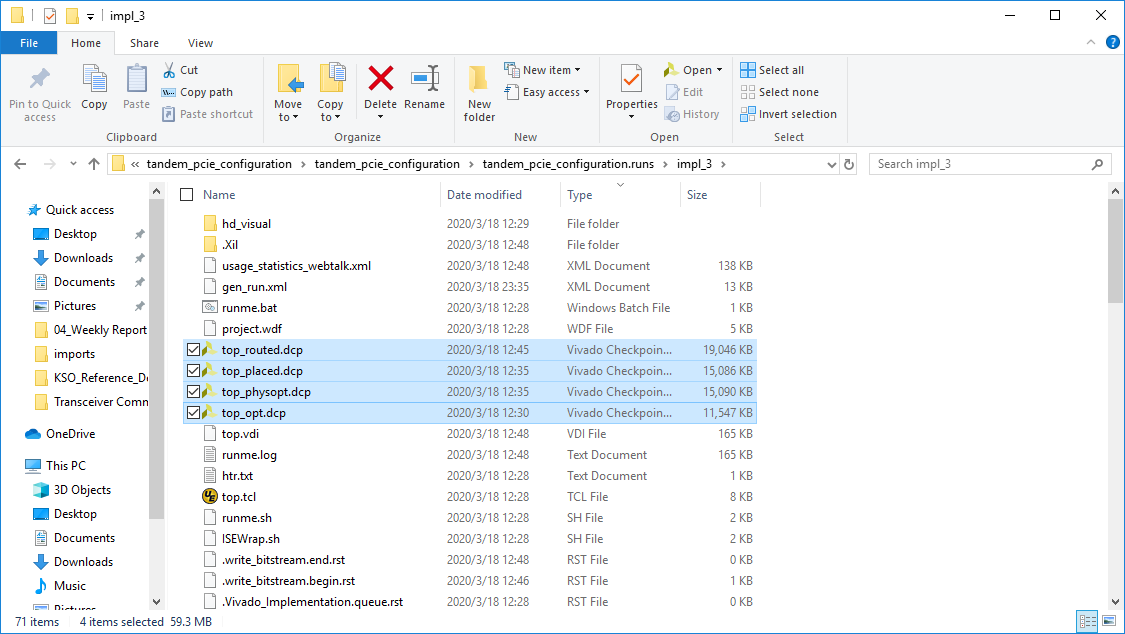


Figure 23: The Implemented Design DCP files

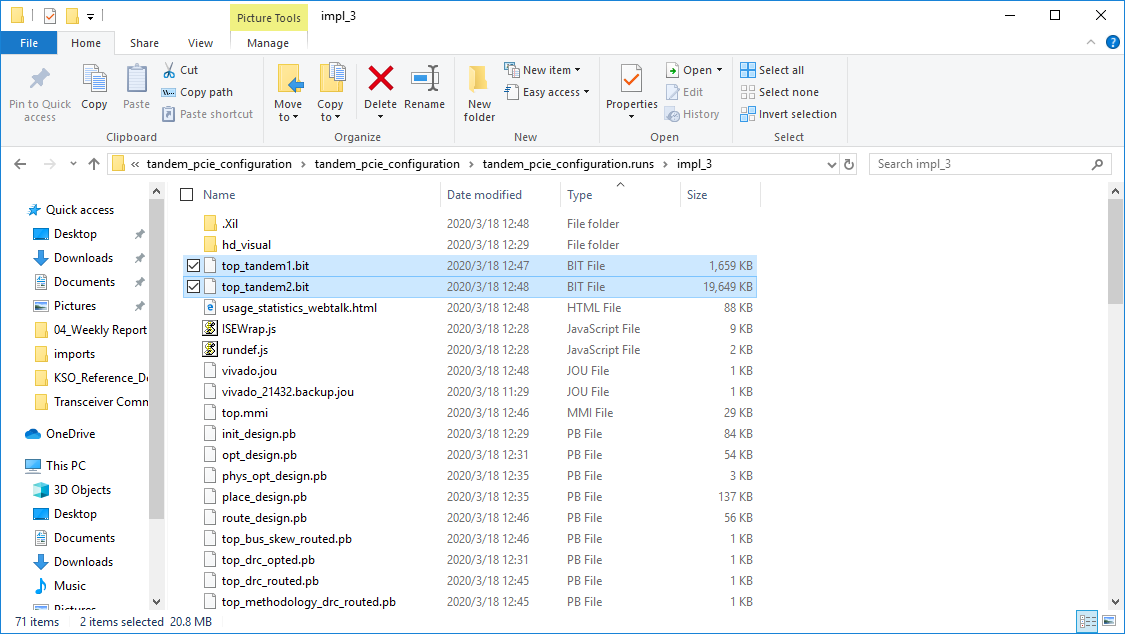


Figure 24: The Implemented Design bitstream files

# Step 8: Adapt to customer project

After the function verification on the board is normal, part of the code will be modified, only the wrapper part will be reserved, and it will be adapted to the customer's code. Through joint simulation and board verification, the function is normal.

# Conclusion

In this reference design, It takes the customer's actual application scenario as a reference. By modifying the example project of IP, it can output a set of test project for the customer faster for the rapid verification of the actual project. Manual modification of code is less, and verification of speed block can effectively help the migration of customer code.