*GT Common Shared 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](#_Toc35467922)

[Introduction 4](#_Toc35467923)

[Tutorial Design Description 4](#_Toc35467924)

[Hardware and Software Requirements 4](#_Toc35467925)

[Locating Tutorial Design Files 4](#_Toc35467926)

[Step 1: Creating a New Project 5](#_Toc35467927)

[Step 2: Creating IP\_1 and example design 9](#_Toc35467928)

[Step 3: Creating IP\_2 and example design 12](#_Toc35467929)

[Step 4: Create the new file for common 15](#_Toc35467930)

[Step 5: Create the new file for top 16](#_Toc35467931)

[Step 6: Modify the related files 16](#_Toc35467932)

[Step 7: Create the new file for simulation 16](#_Toc35467933)

[Step 8: Simulate the Design 17](#_Toc35467934)

[Step 9: Synthesis the Design 17](#_Toc35467935)

[Step 10: Implement the Design 18](#_Toc35467936)

[Step 11: Test the design on Board 18](#_Toc35467937)

[Step 12: Adapt to customer project 18](#_Toc35467938)

[Conclusion 18](#_Toc35467939)

# Introduction

This tutorial introduces the GT common shared reference design method use the Xilinx® Vivado® Integrated Design Environment (IDE).

The tutorial describes the basic steps involved in taking a two small IP example design from RTL to implementation, and simulation. This design can fully verify the usability of the prototype before the design of the board. And it can extend other similar application scenarios.

# Tutorial Design Description

This tutorial is based on two GT transceiver IP integrator examples. It creates a top-level file, which integrates two example projects with different rates, constrains them to the same Quad, and shares the same common. And use two independent QPLL which are configured different parameters in this common.

In this reference design, you will learn about the GT common shared mathod in the Vivado® IDE. It will take you through the steps of project creation and the method how to modify the relation code to achieve common share between the two channels. It will also demonstrate using the behavioral simulation and test on board.

If you want to use multi-channel and multi rate sharing multi commons, you can refer to similar methods and use common code for implementation.

# 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 Kintex UltraScale Plus devices xcku3p-ffvb676-2-i, it needs to modify the device or constraint then run on the Evaluation Board of Kintex® 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:

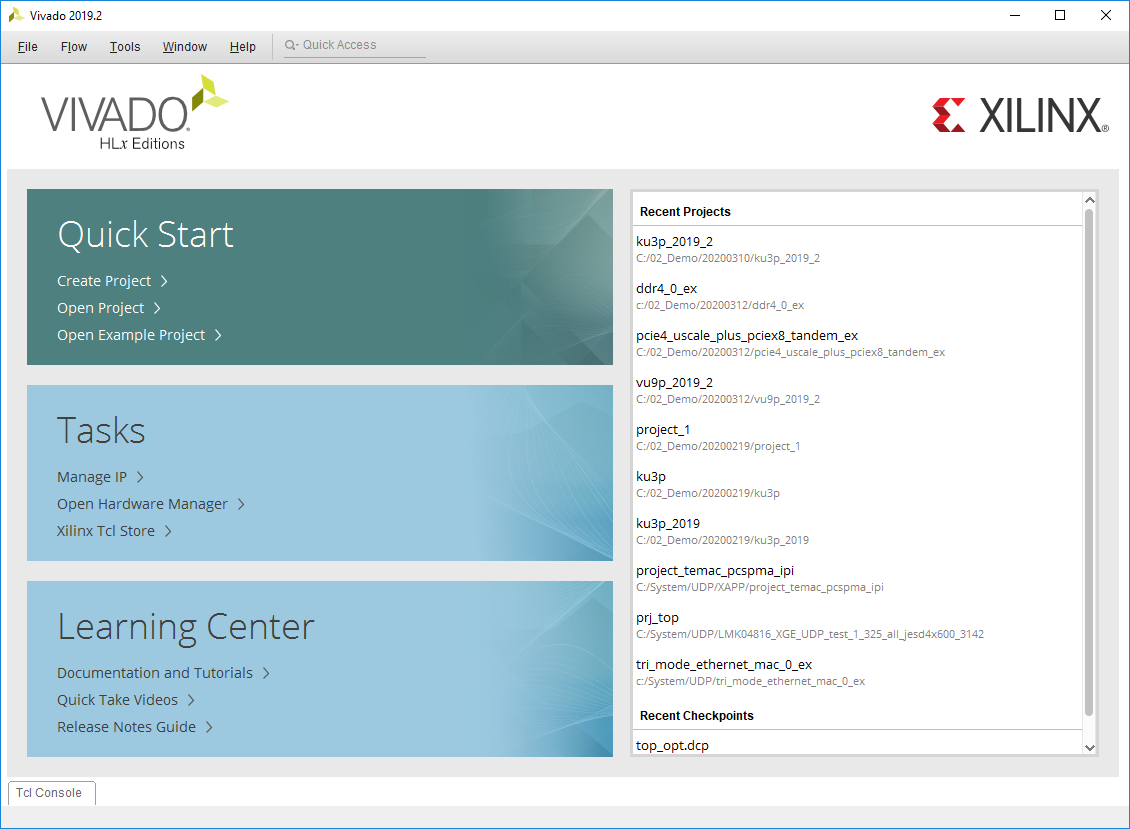


Figure 1: Vivado IDE – Getting Started Page

1. Create New Project to start the New Project wizard

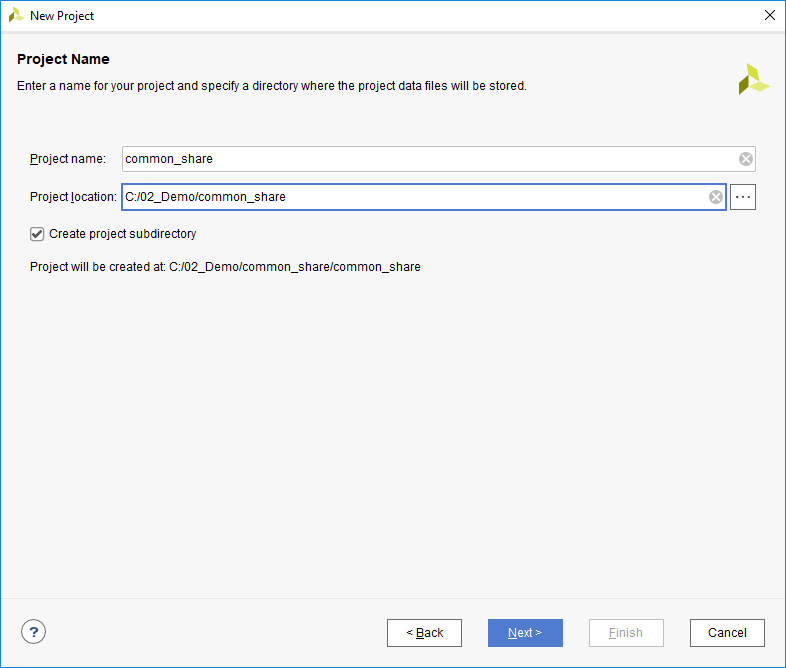


Figure 2: Creating a New Project

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

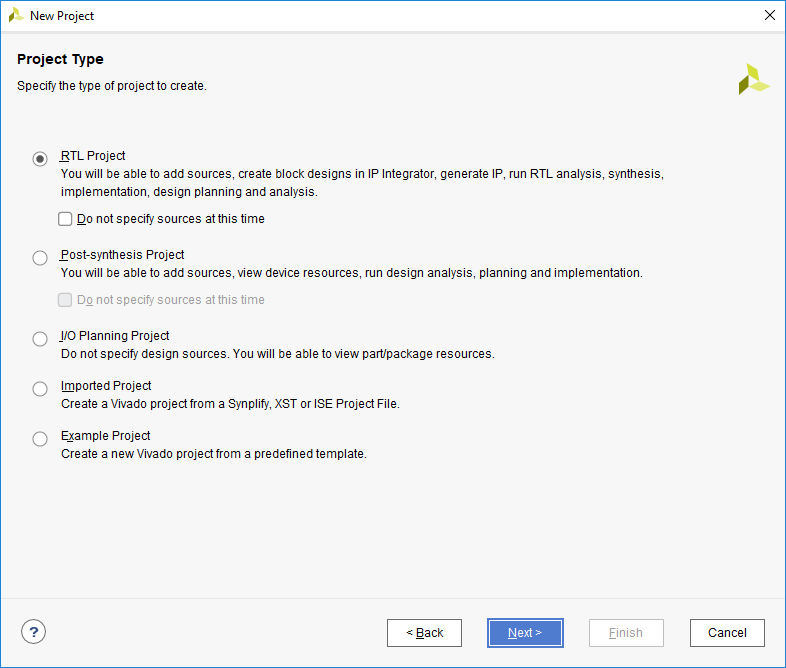


Figure 3: Setting Project Type

1. Don’t add any sources

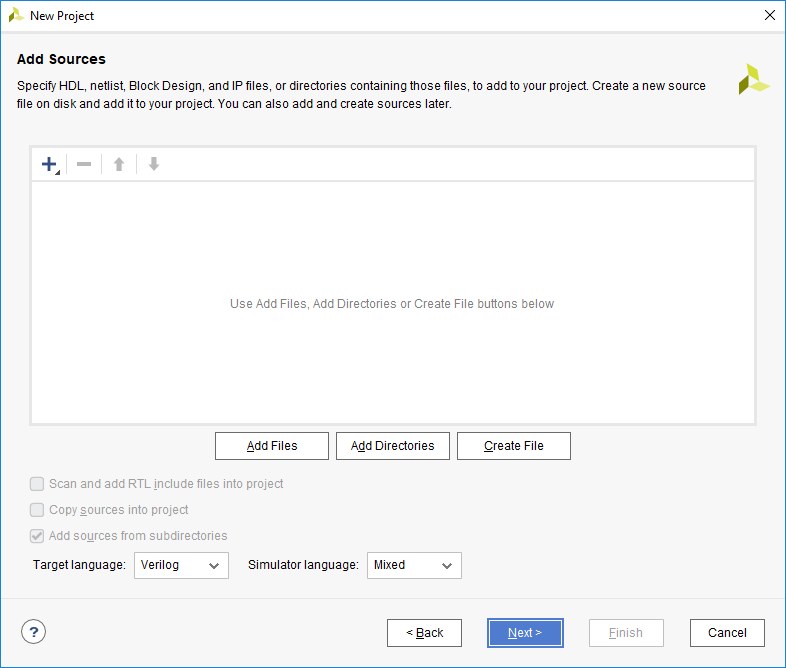


Figure 4: Setting HDL Source Type

1. In the Default Part dialog box, select the devices of xcku3p-ffvb676-2-i.

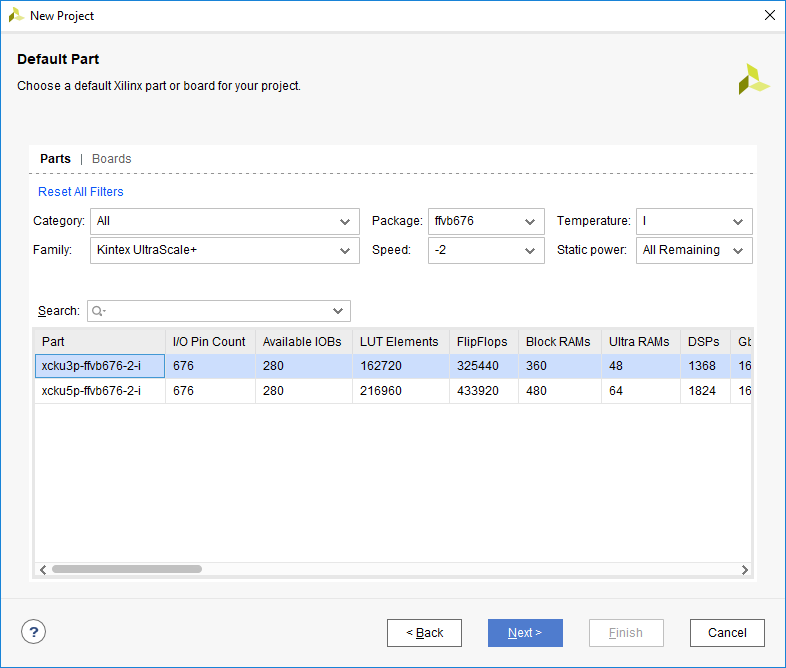


Figure 5: Setting Default Part

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

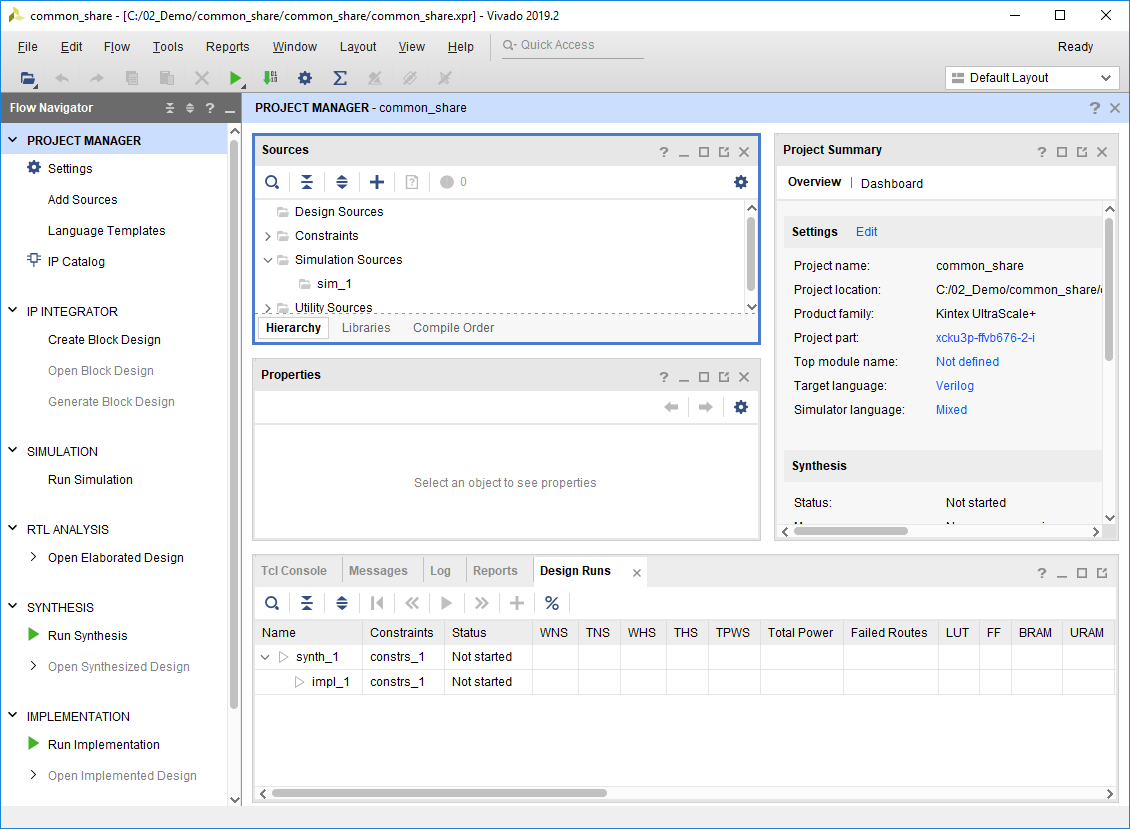


Figure 6: Project in Vivado IDE

# Step 2: Creating IP\_1 and example design

To create the first IP, it’s named gtwizard\_ultrascale\_10G is configured the line rate 9.8304 Gb/s used QPLL0, the channel used x0y15 in Quad x0y3.The steps as below.

1. Search the GT transceiver used IP catalog:

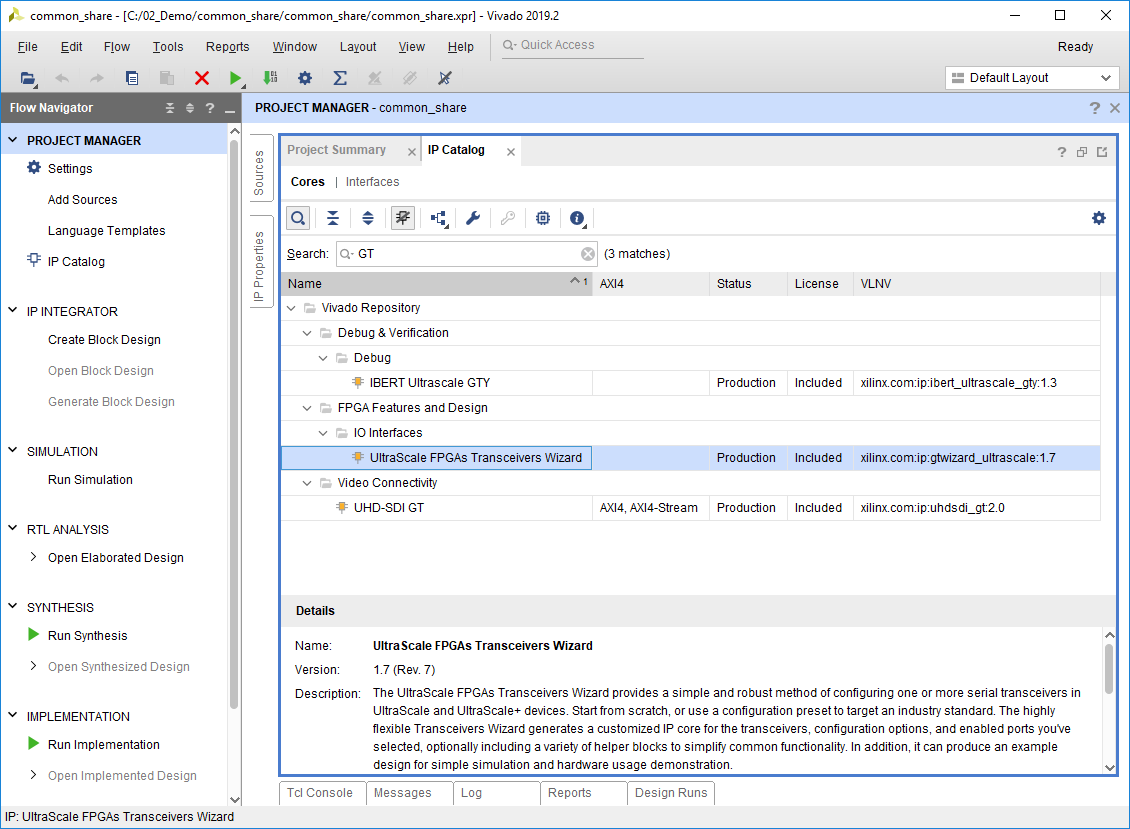


Figure 7: Gt wizard ultrascale in IP catalog

1. Customize the IP gtwizard\_ultrascale\_10G basic:

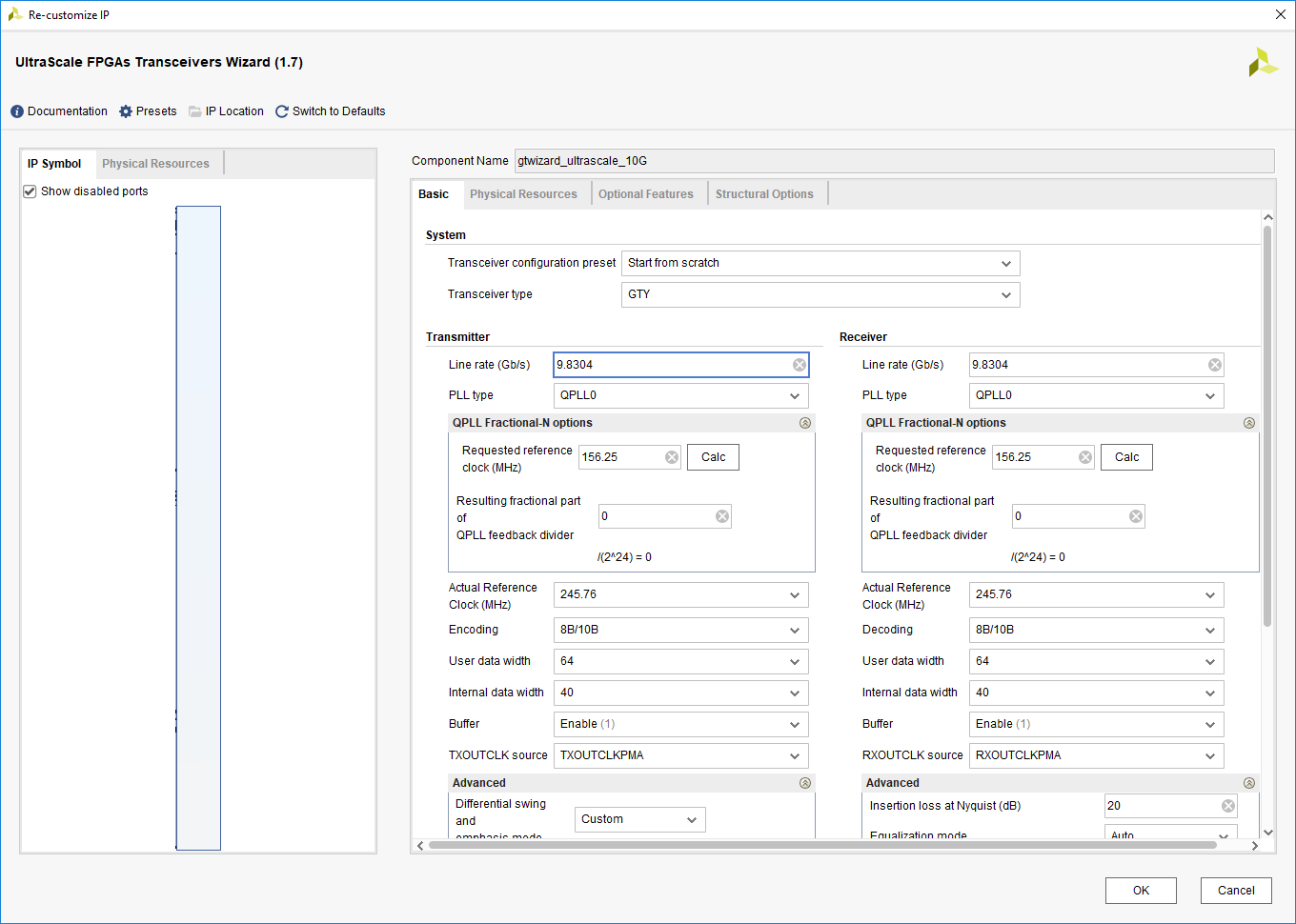


Figure 8: Setting the basic parameters

1. Customize the IP gtwizard\_ultrascale\_10G location:

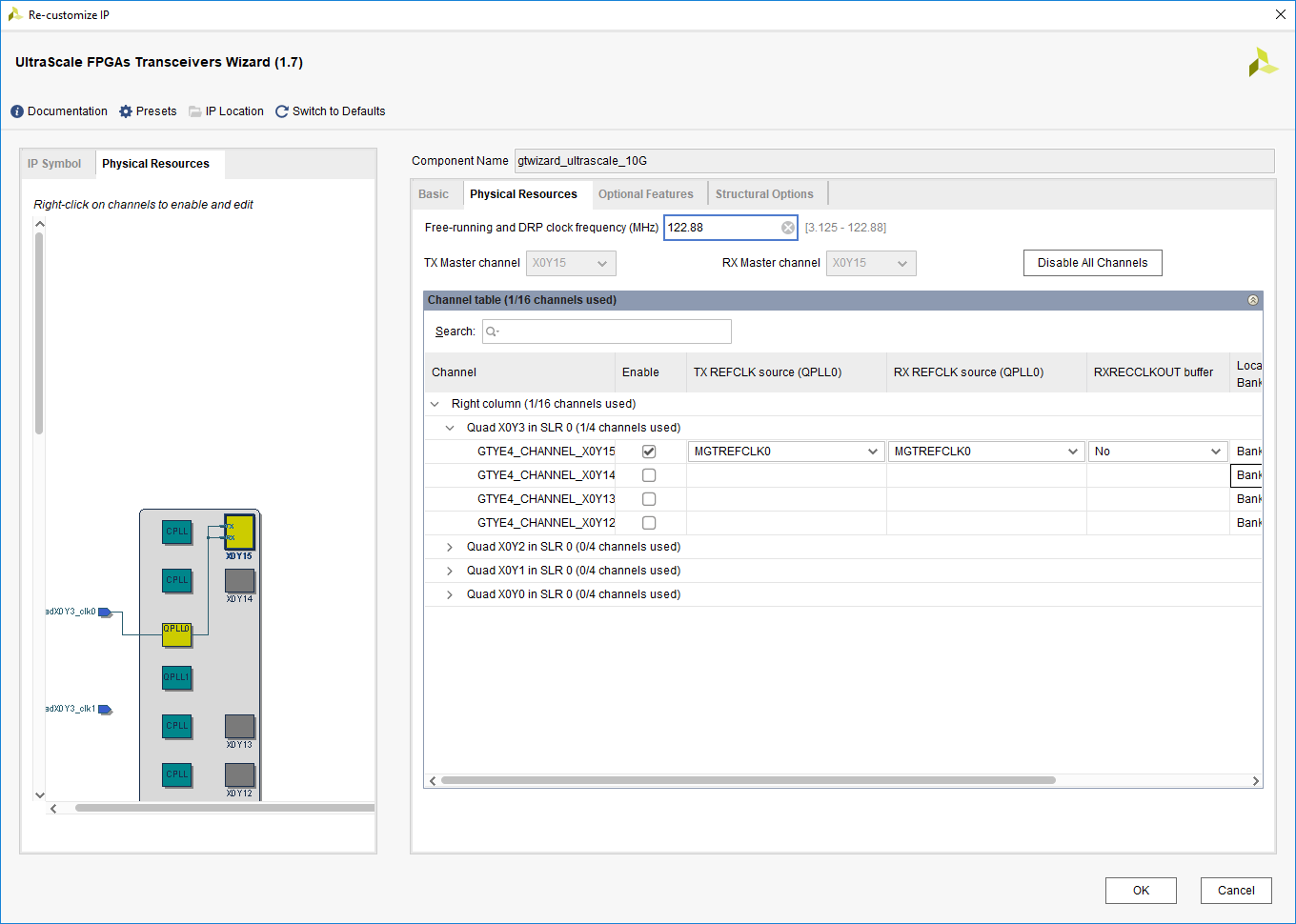


Figure 9: Setting the Physical Resources

1. Customize the IP gtwizard\_ultrascale\_10G location:

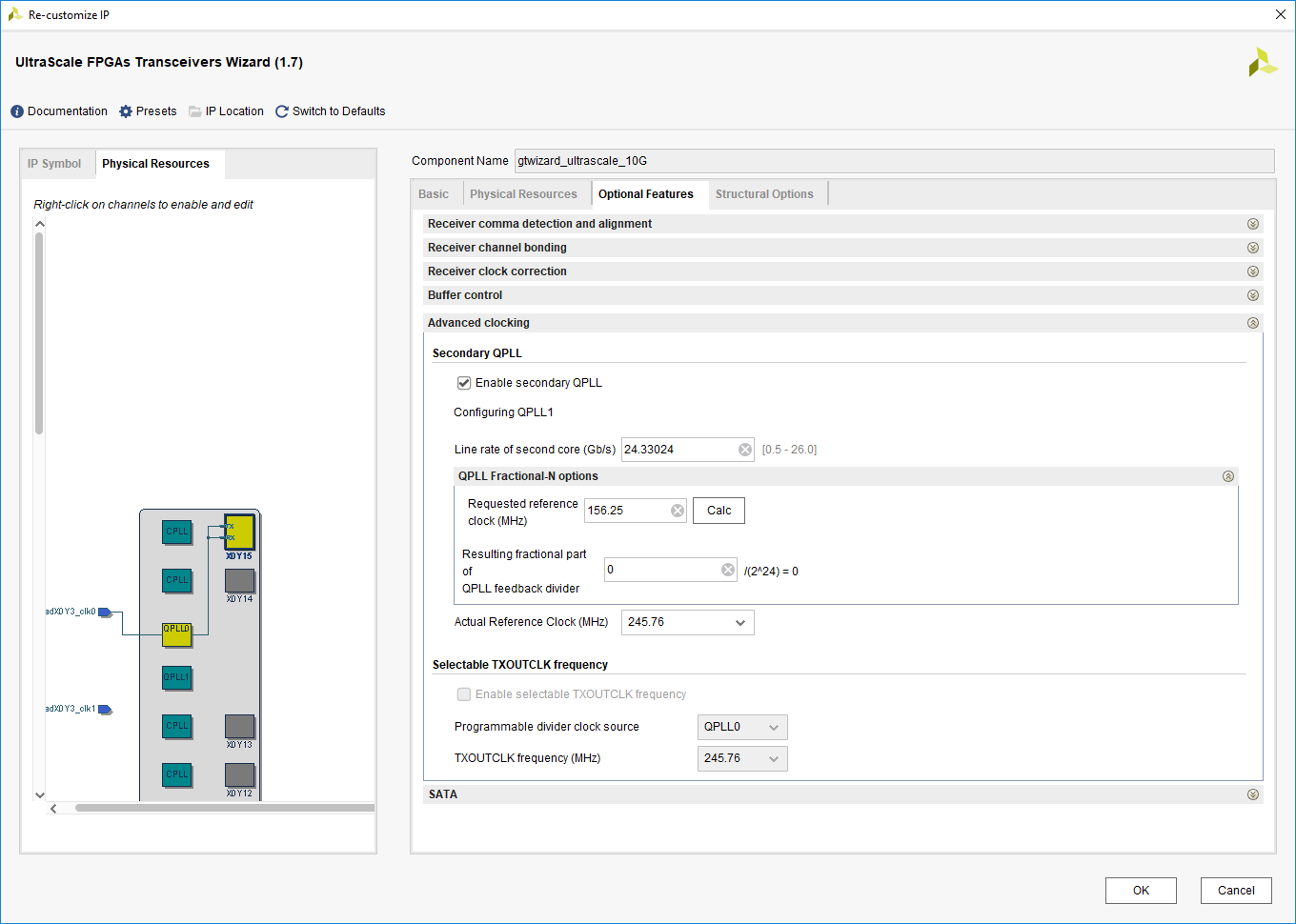


Figure 10: Setting the Optional Feathers

1. Ensure that the COMMON in the Example Design:

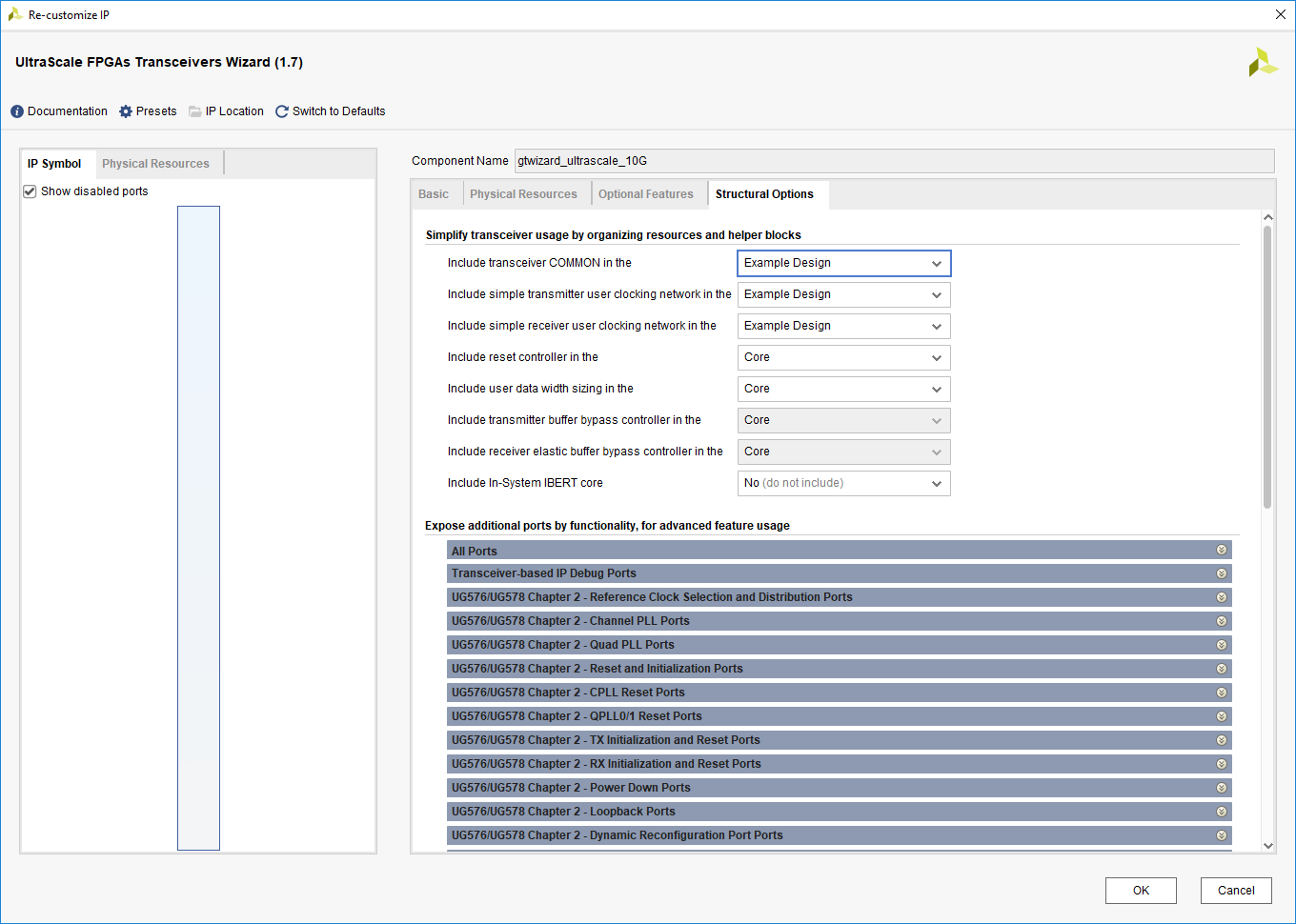


Figure 11: Setting the Structural Options

1. Generate the IP gtwizard\_ultrascale\_10G:

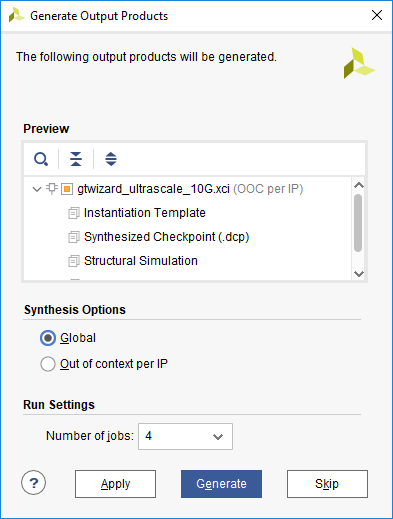


Figure 12: Setting synthesis Options and Generate

1. Open IP Example Design of gtwizard\_ultrascale\_10G:

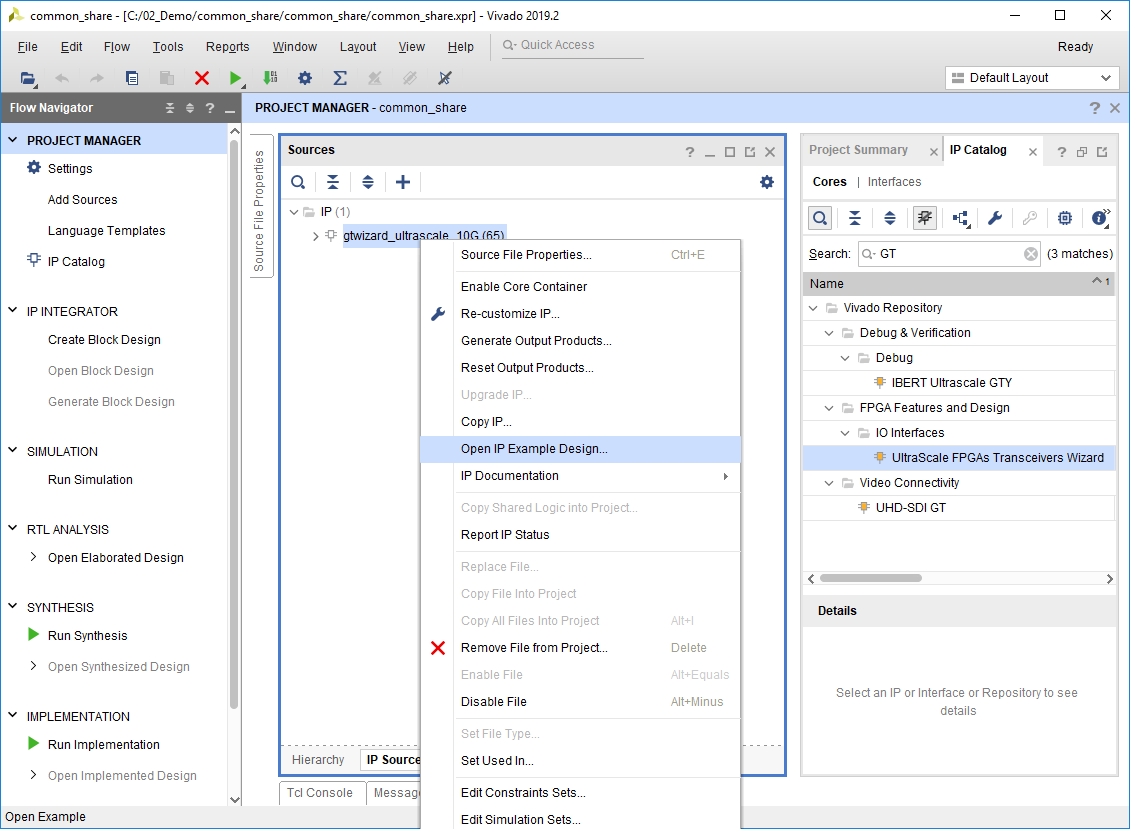


Figure 13: Open IP Example Design

# Step 3: Creating IP\_2 and example design

To create the second IP, it’s named gtwizard\_ultrascale\_25G is configured the line rate 24.33024 Gb/s used QPLL1, the channel used x0y14 in Quad x0y3.The steps as below.

1. Search the GT transceiver used IP catalog:

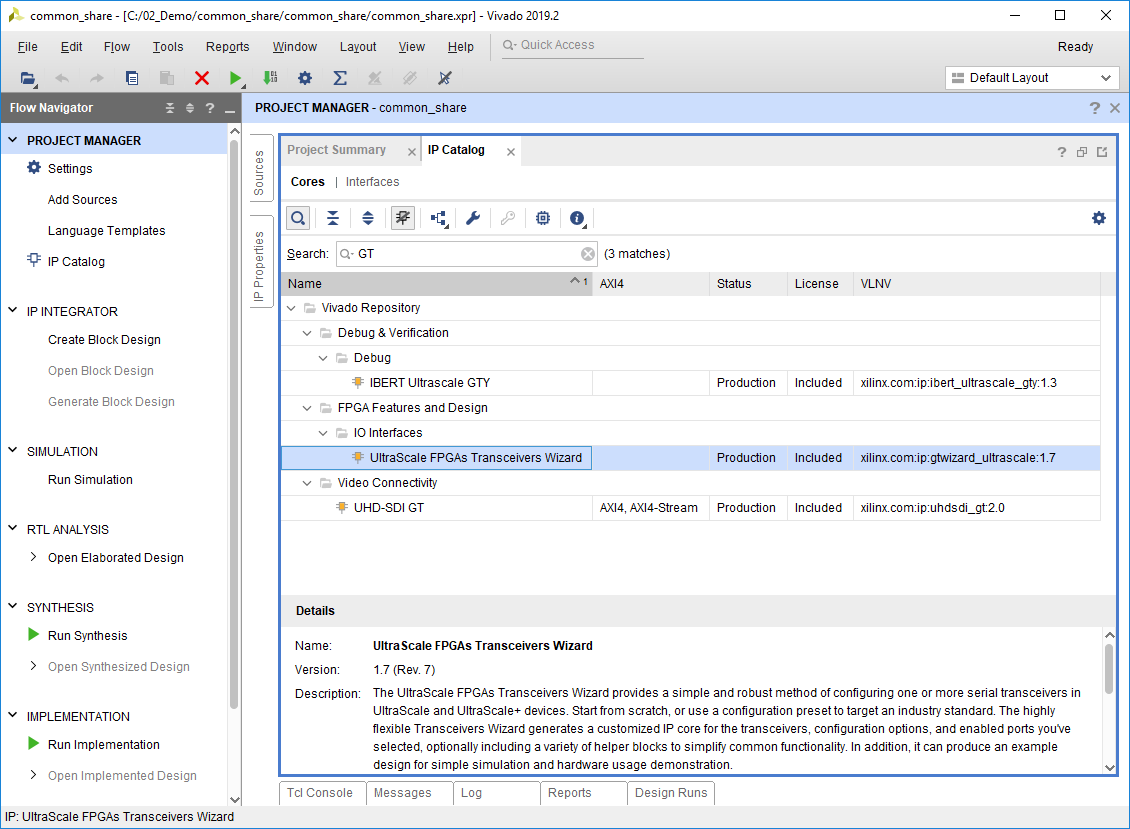


Figure 14: Gt wizard ultrascale in IP catalog

1. Customize the IP gtwizard\_ultrascale\_25G basic:

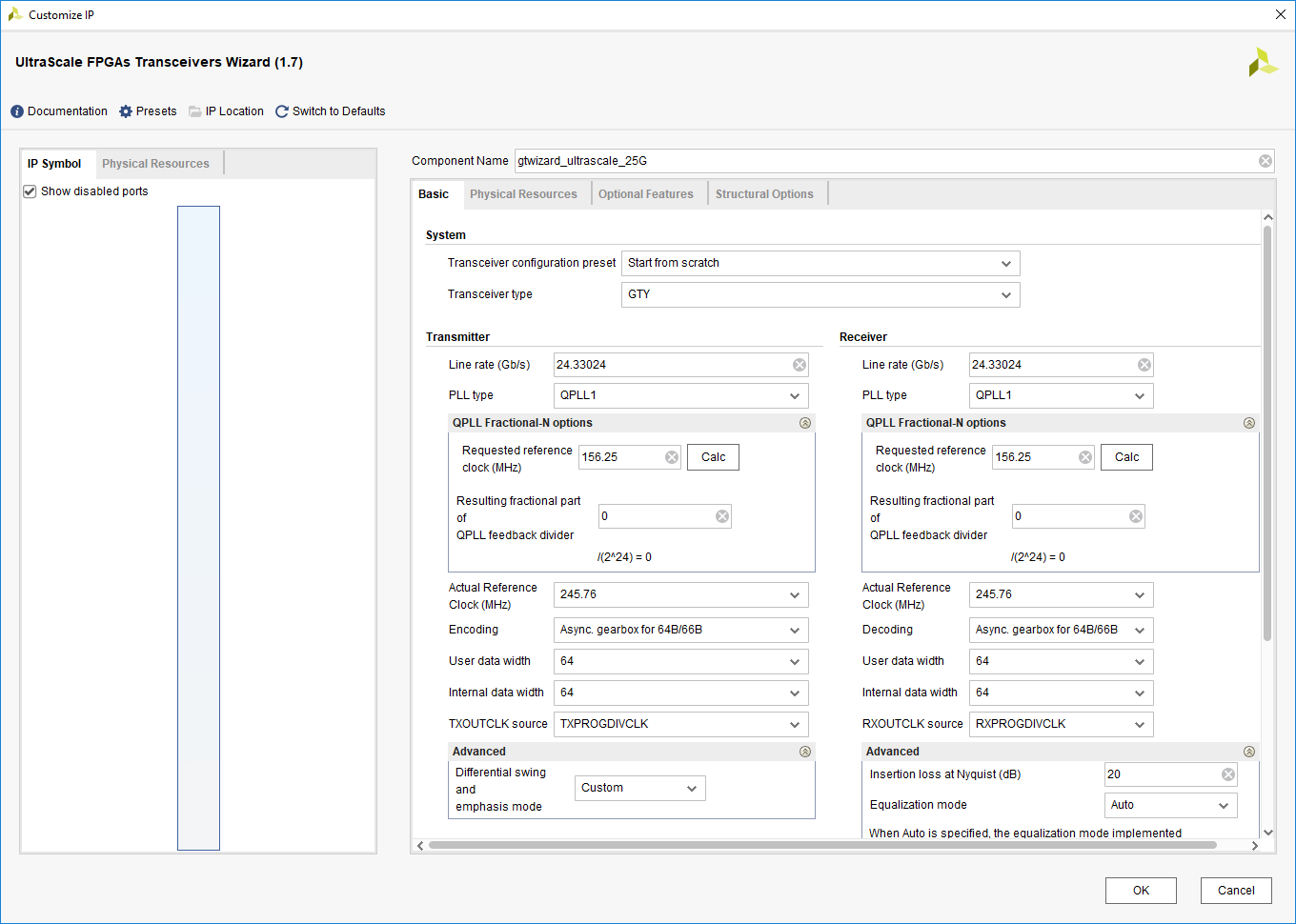


Figure 15: Setting the basic parameters

1. Customize the IP gtwizard\_ultrascale\_25G location:

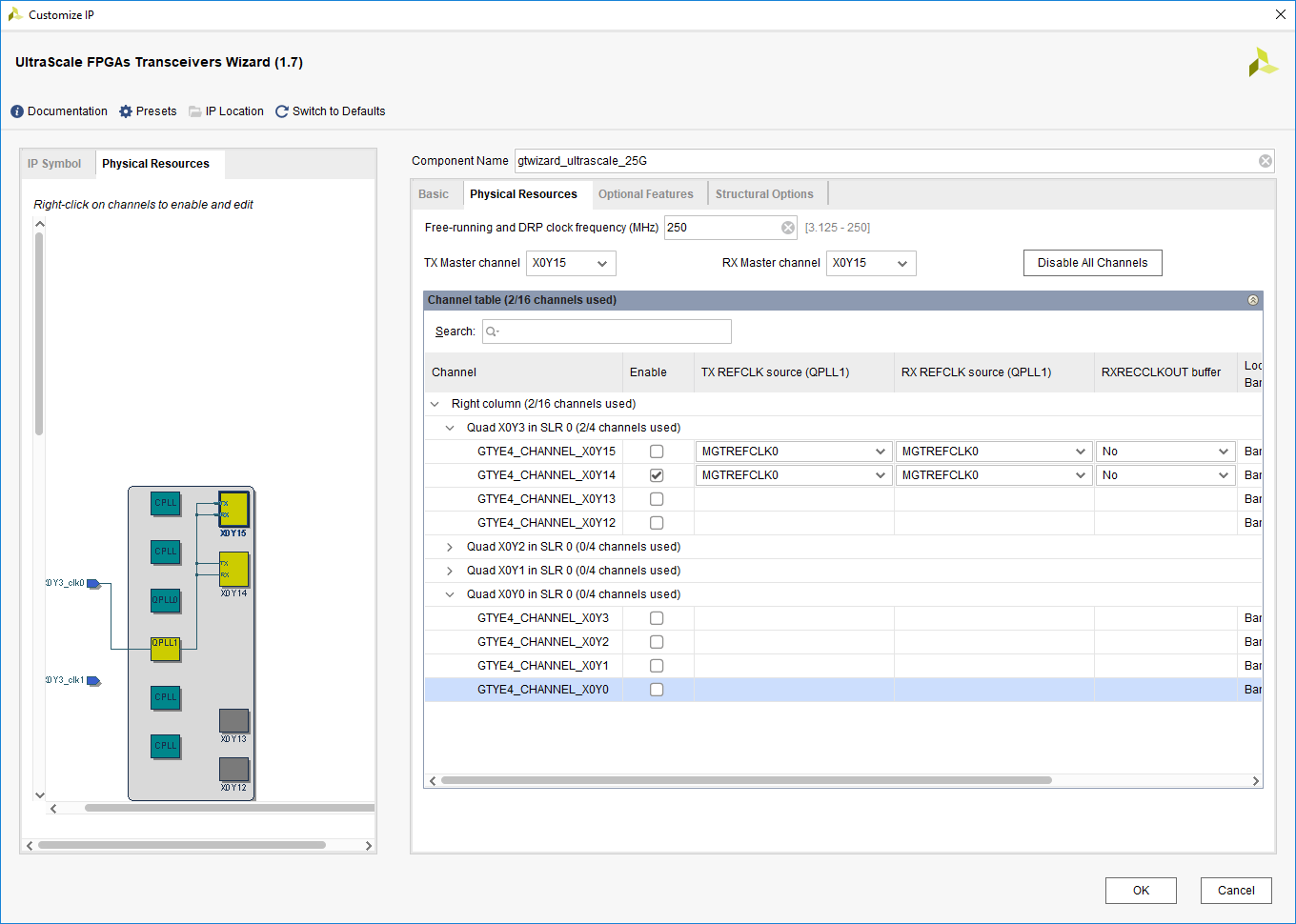


Figure 16: Setting the Physical Resources

1. Customize the IP gtwizard\_ultrascale\_25G location:

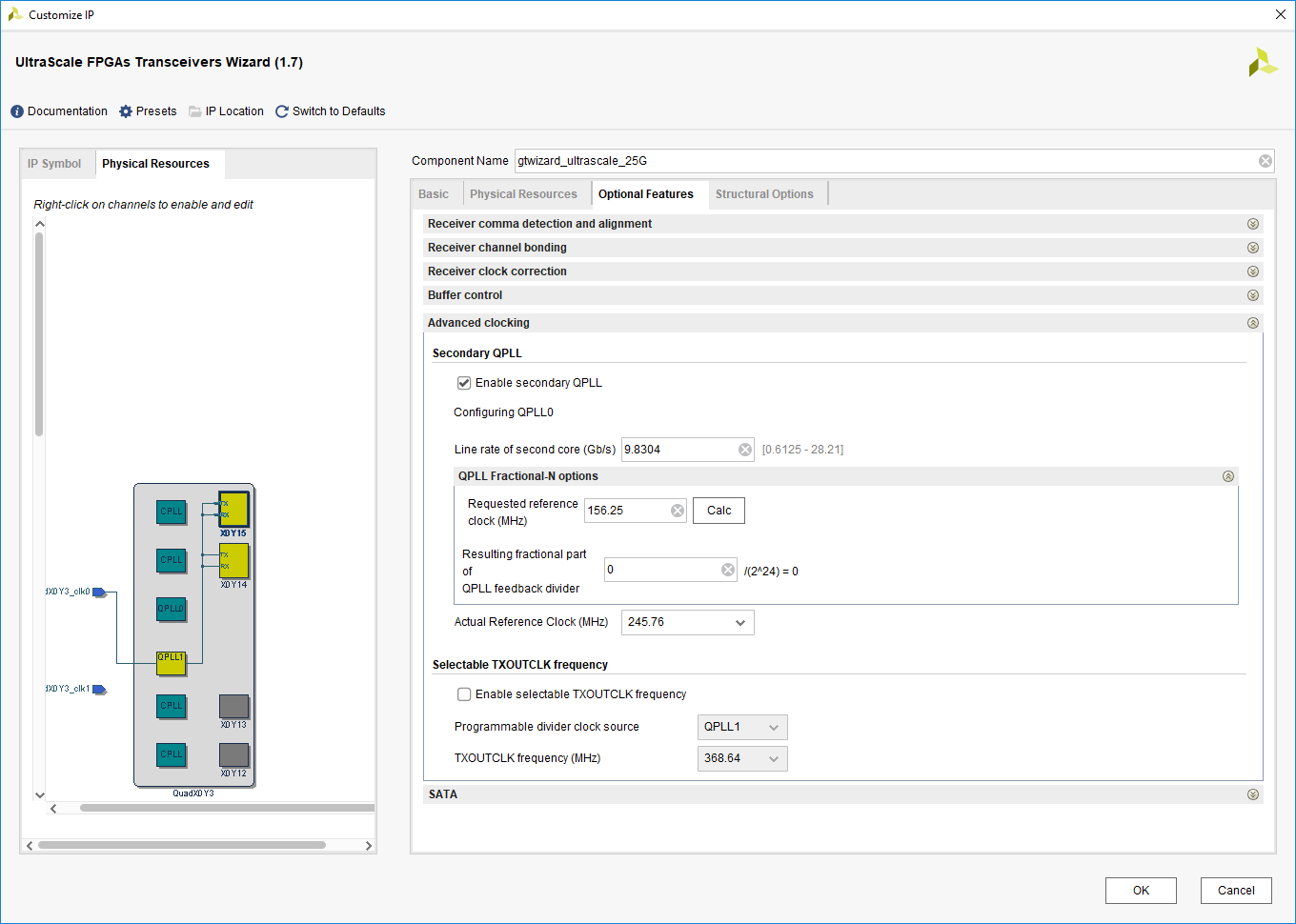


Figure 17: Setting the Optional Feathers

1. Ensure that the COMMON in the Example Design:

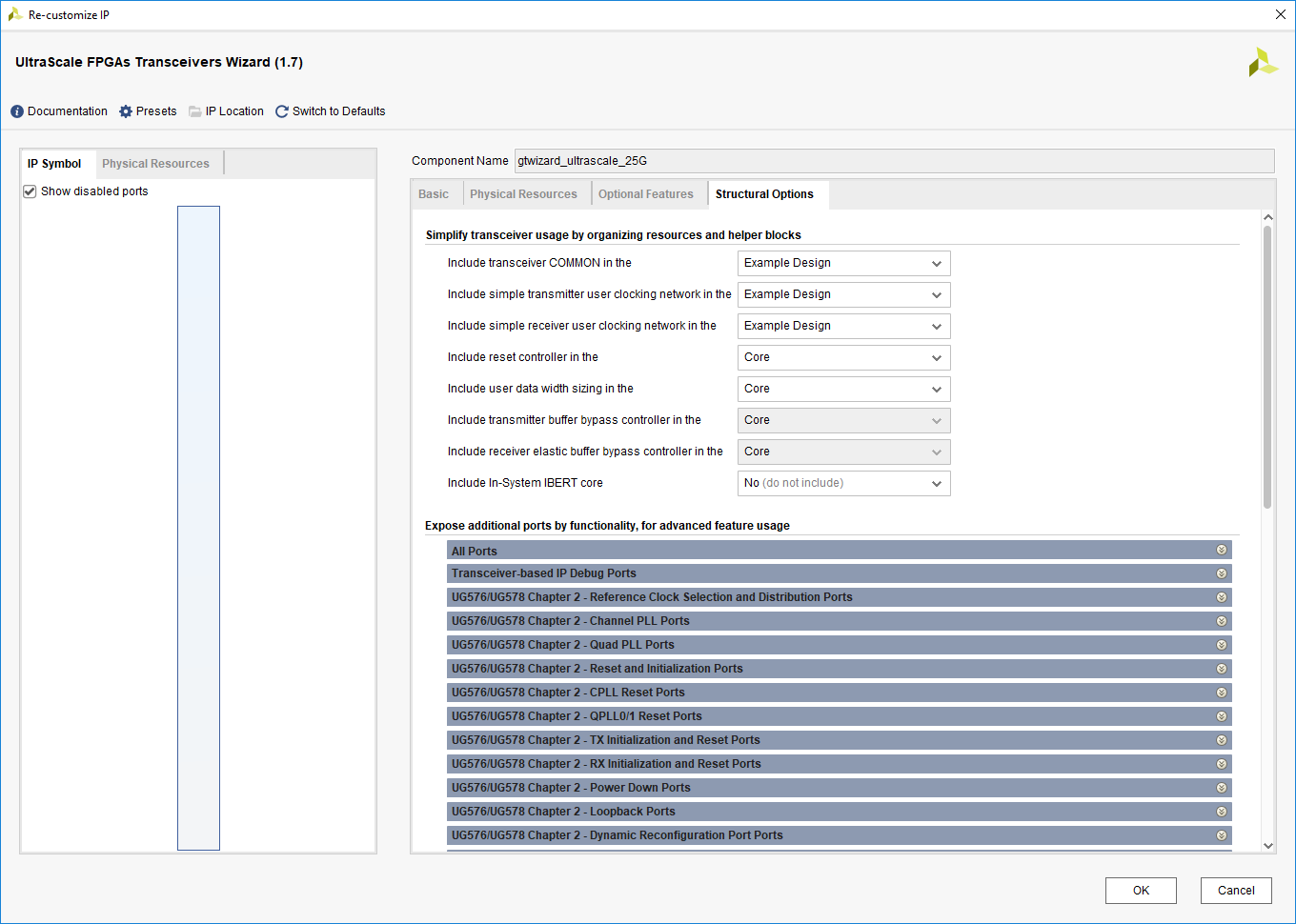


Figure 18: Setting the Optional Feathers

1. Generate the IP gtwizard\_ultrascale\_25G:

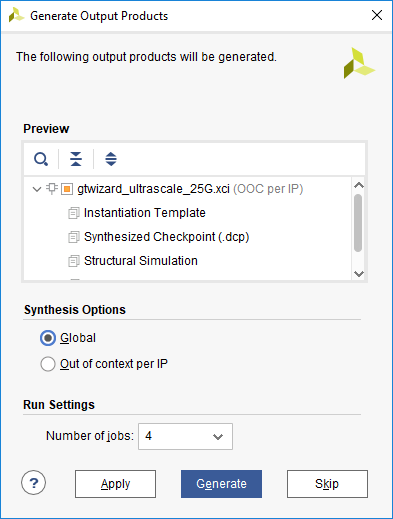


Figure 19: Setting synthesis Options and Generate

1. Open IP Example Design of gtwizard\_ultrascale\_25G:

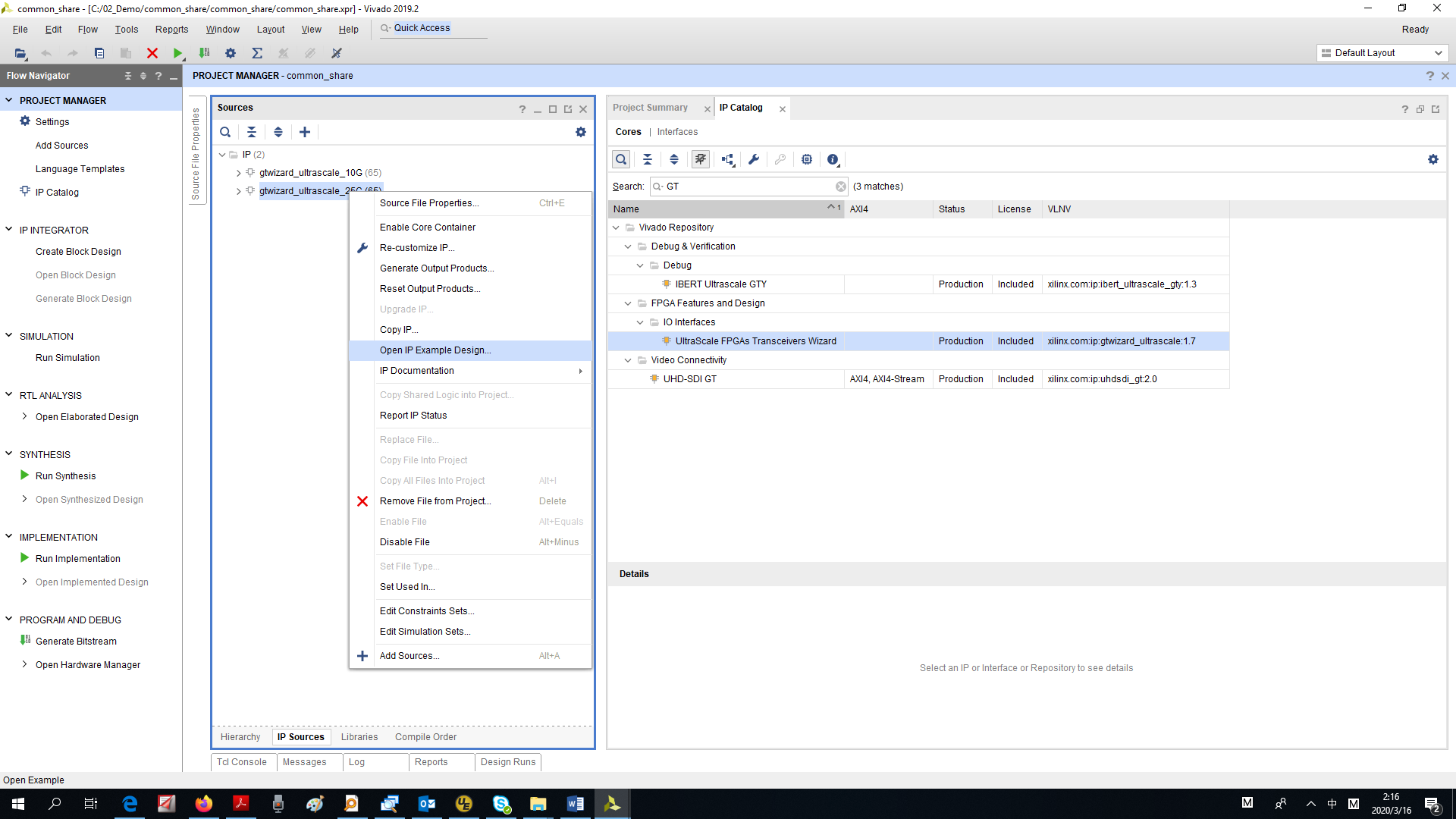


Figure 20: Open IP Example Design

# Step 4: Create the new file for common

To create a new file called gt\_commons.v,which used the parameters of the common either from the gtwizard\_ultrascale\_10G or gtwizard\_ultrascale\_25G example design. This case used the gtwizard\_ultrascale\_10G\_gtye4\_common\_wrapper module from the IP of gtwizard\_ultrascale\_10G.

****

# Step 5: 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 three modules, gtwizard\_ultrascale\_10G\_example\_top, gtwizard\_ultrascale\_25G\_example\_top and gt\_commons. The common control signal reset IBUF and IBUFDS\_GTE4 for common both need to place on the top.



# Step 6: Modify the related files

To modify the related files to ensure that the key signals of the common module can be driven normally. Delete the common in the IP examples and replaced by the common added in the top file. The related files as below.

# Step 7: Create the new file for simulation

To create a simulation top file called top\_sim.v, which used to integrate these two IP examples simulation into one project. And it will include the modules of top.



# Step 8: Simulate the Design

Simulate the design focus on the common functions, and the status of both separated IP.

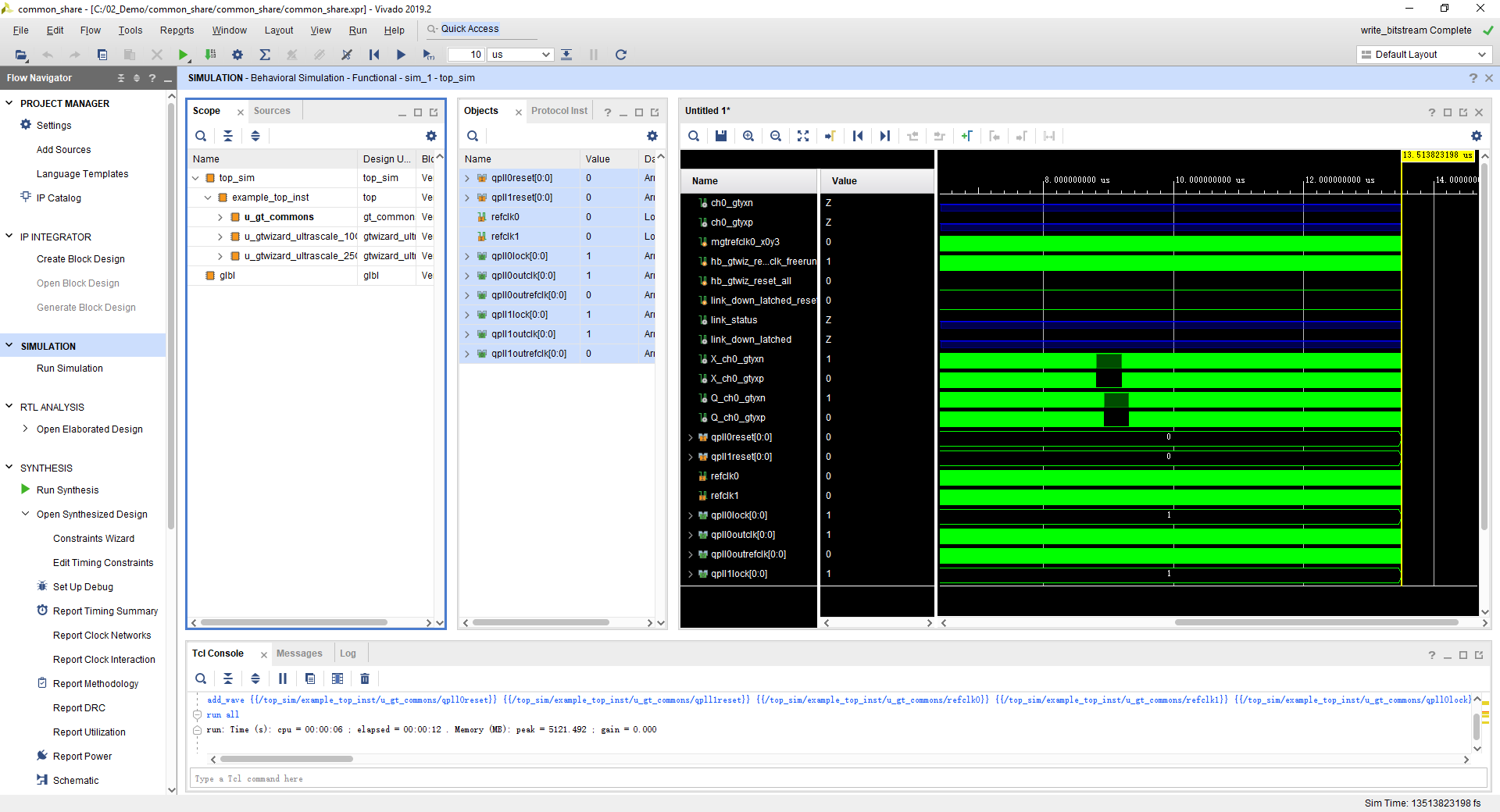


Figure 21: The Behavior Simulation

# Step 9: Synthesis the Design

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

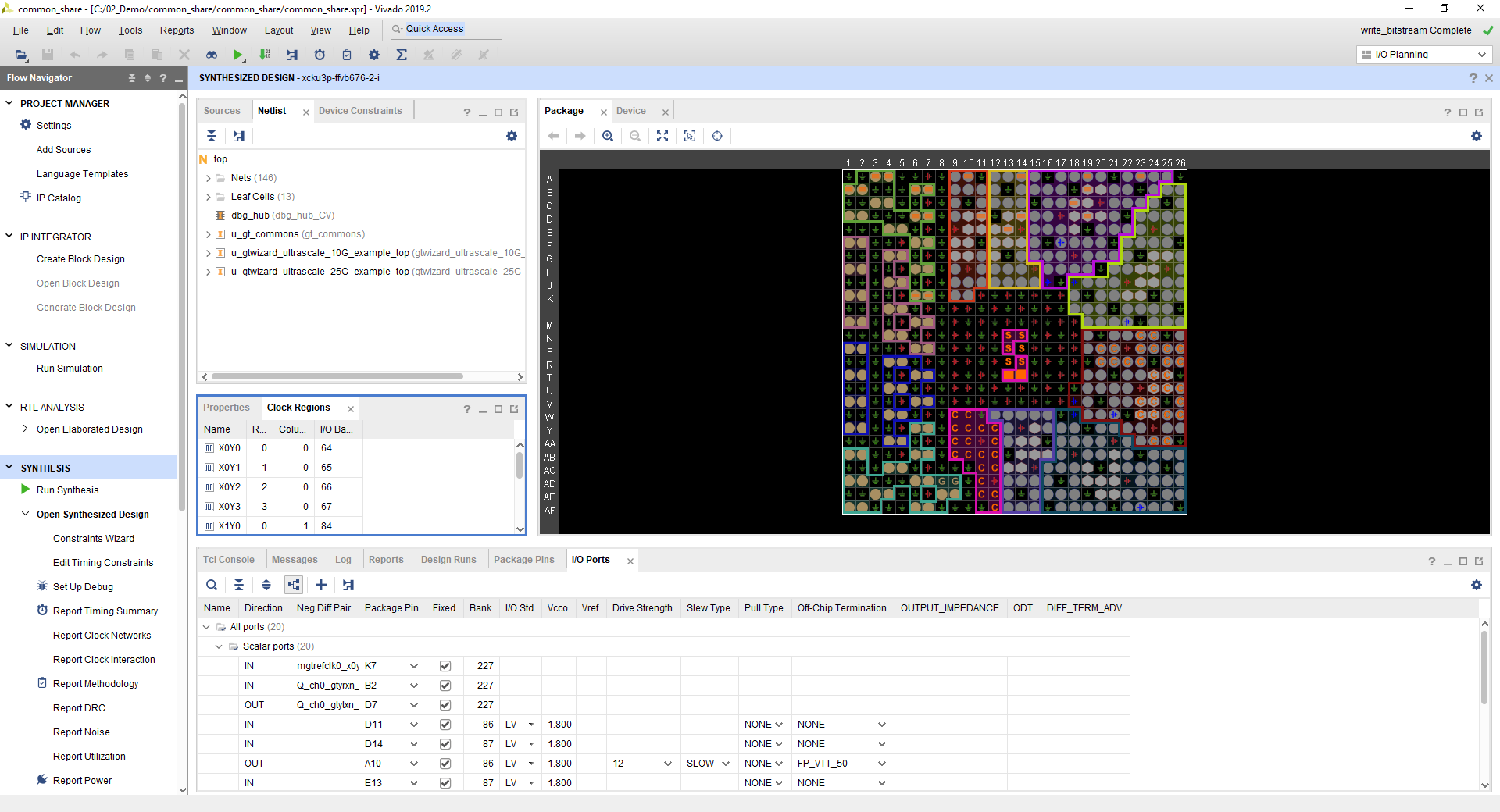


Figure 22: The Synthesized Design

# Step 10: 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.

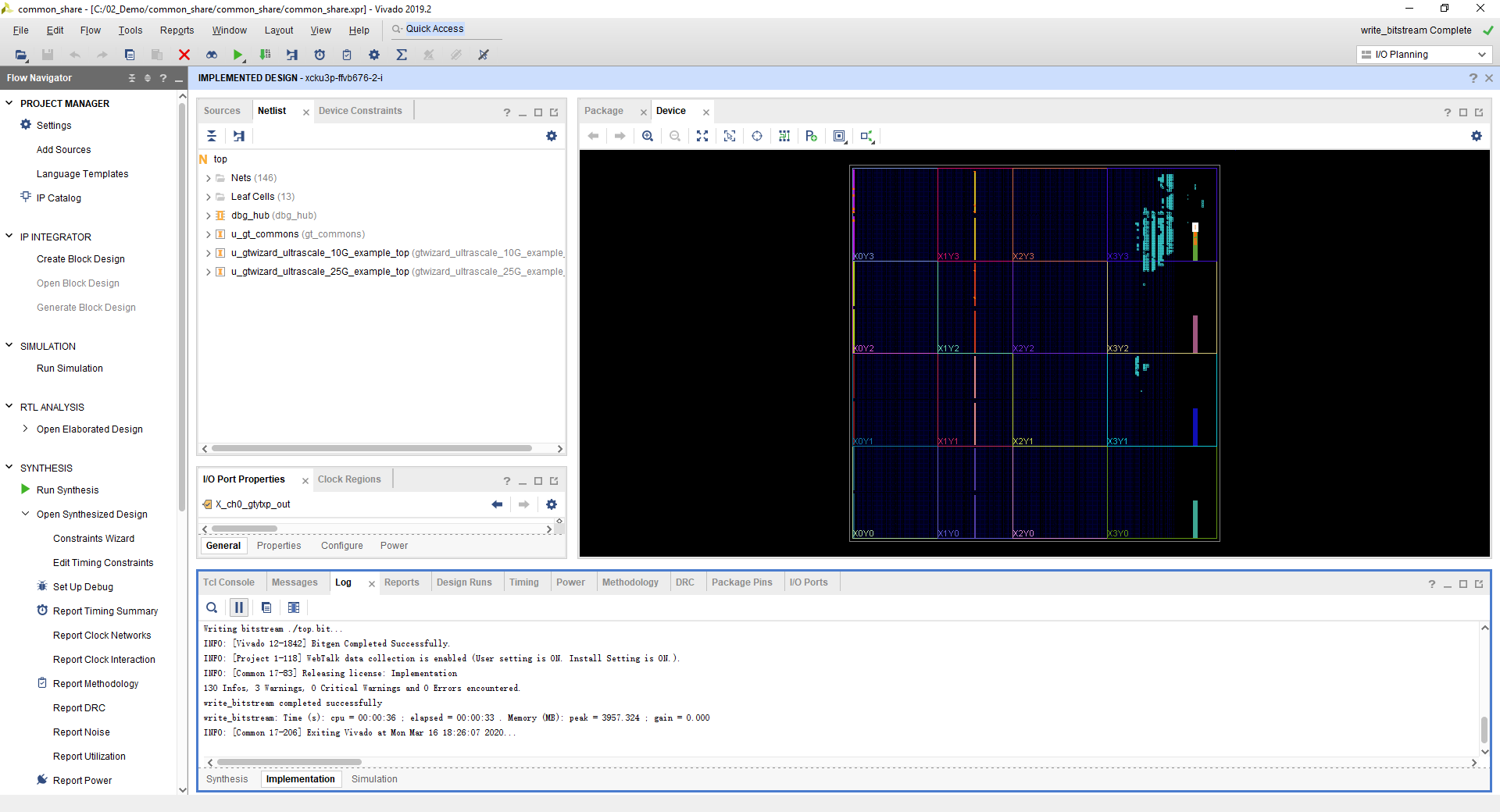


Figure 23: The Implemented Design

# Step 11: Test the design on Board

Test on the board of customer, used the fiber nearby loopback mode. The result judgment is normal, by compared the actual test and simulation results.

# Step 12: 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.

In addition, for the design of multi-channel and multi-protocol types, you can refer to the same method to modify.