Versal ACAP Transceivers Wizard v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG331 (v1.0) July 14, 2020

Table of Contents

Chapter 1

Introduction

The Versal™ ACAP GTY Transceivers Wizard IP solution helps configure one or more serial transceivers. You can start from scratch, input your requirements, and generate valid configurations. In addition, this wizard can also produce an example design for simple simulation.

Features

- Simple and intuitive feature selection flow.
- Onetime GUI entry for multiple line rate options to enable seamless dynamic rate switching without re-programming GT registers.
- Design entry is through IP integrator (IPI) for custom IP.
- Synthesizable example design with configurable pseudo-random binary sequence (PRBS) data generator, checker, and link status indicator logic to quickly demonstrate core and transceiver functionality in simulation.

Overview

Navigating Content by Design Process

Xilinx® documentation is organized around a set of standard design processes to help you find relevant content for your current development task. This document covers the following design processes:

- **Hardware, IP, and Platform Development:** Creating the PL IP blocks for the hardware platform, creating PL kernels, subsystem functional simulation, and evaluating the Vivado® timing, resource use, and power closure. Also involves developing the hardware platform for system integration. Topics in this document that apply to this design process include:
    - Customizing and Generating the Core
    - Chapter 5: Example Design

Licensing and Ordering

This Xilinx® LogiCORE™ IP module is provided at no additional cost with the Xilinx Vivado® Design Suite under the terms of the Xilinx End User License.

Note: To verify that you need a license, check the License column of the IP Catalog. Included means that a license is included with the Vivado® Design Suite; Purchase means that you have to purchase a license to use the core.

For more information about this core, visit the Versal ACAP Transceivers Wizard product web page.

Information about other Xilinx® LogiCORE™ IP modules is available at the Xilinx Intellectual Property page. For information about pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Chapter 3

Product Specification

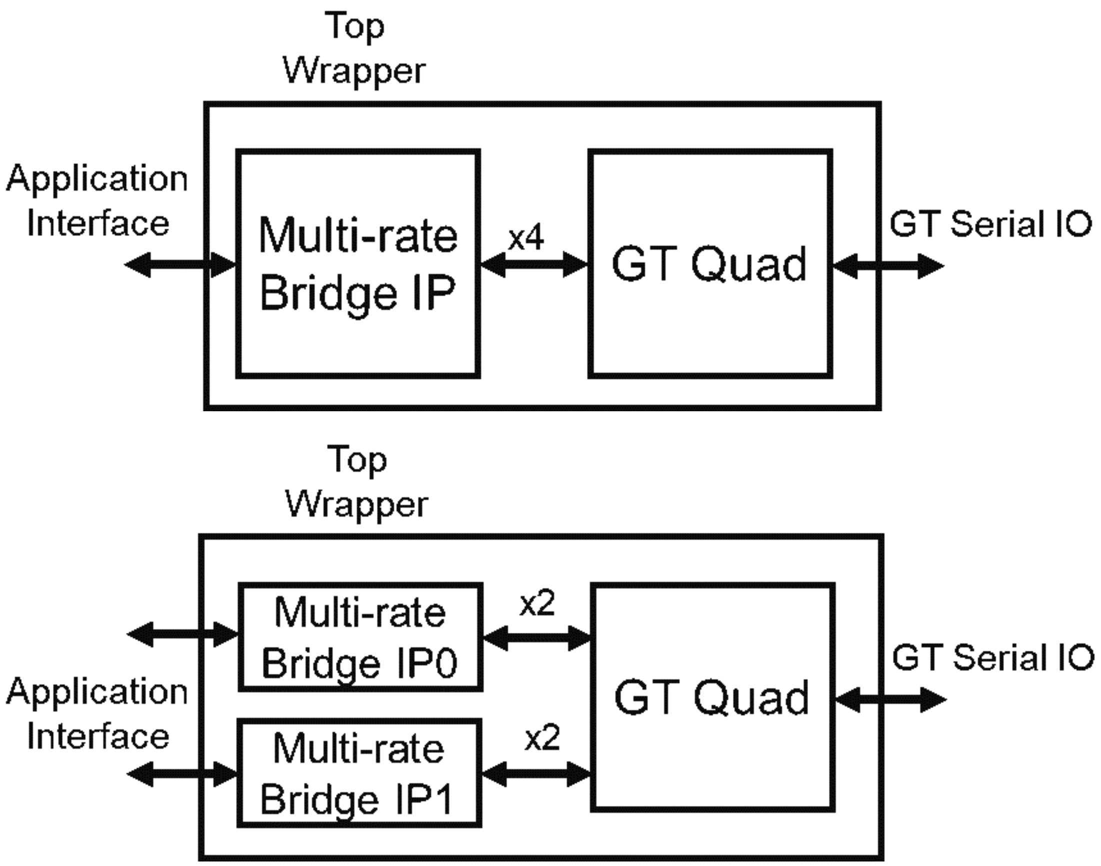
The Versal™ ACAP Transceivers Wizard solution contains two cores:

1. Versal ACAP Transceivers Bridge - A reference parent IP (Bridge IP) that configures transceivers wizard parameters.

   - **Bridge IP:** Versal ACAP transceivers bridge IP (`gt_bridge_ip_v1_0`) is available in the IP integrator canvas. A custom design entry is created through a Bridge IP only. It instantiates, configures, and connects single or multiple GT quad base IPs (`gt_quad_base_v1_0`).You can add Versal ACAP Transceiver Bridge IP (`gt_bridge_ip_v1_0`) in IPI and configure its parameters (`gt_quad_base_v1_0` parameters are programmable through `gt_bridge_ip_v1_0 GUI`) including number of lanes and click **Block Automation**. It instantiates multiple `gt_quad_base_v1_0` based on the number of lanes configured in `gt_bridge_ip_v1_0` and makes all required connections. For more information, see IP Integrator (IPI) Design Entry for Custom IPThe supported features in the Bridge IP are:

     - Simplex and Duplex configurations.
     - Multiple line rate configurations can be natively captured as part of IP customization, not necessarily requiring APB3 register updates to realize dynamic rate change. Up to 16 line rates can be configured per channel per direction. It will be enhanced to 256 line rates support in future version.

Figure 1: **Single and Multiple Bridge IPs - GT Quad Connections**



2. Versal ACAP Transceivers Wizard - A wrapper around `GTYE5_QUAD`primitive. It consists of single GTYE5 quad (GT quad base IP). Multiple transceivers wizards are instantiated for multi-lane (>4 lanes) designs.

    - **GT quad base IP:** GT quad base IP is available as Versal ACAP transceivers wizard (`gt_quad_base_v1_0`) in the Vivado® IP catalog. It instantiates, configures, and connects a `GTYE5_QUAD` primitive.The objective of GT quad base IP is to showcase various GT quad features through simulations. Simulation support is provided for one GT quad. You can configure GT quad base IP to share multiple Protocol IPs each supporting multiple line rates. You can create example design and simulate to understand dynamic line rate switching and GT quad sharing across multiple Protocol IPs.

Design Flow Steps

This section describes customizing and generating the core, constraining the core, and the simulation, synthesis, and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
- *Vivado Design Suite User Guide: Designing with IP* (UG896)
- *Vivado Design Suite User Guide: Getting Started* (UG910)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado® Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

As mentioned in Chapter 3: Product Specification, Versal™ ACAP Transceivers Wizard solution contains two cores:

1. Versal ACAP Transceivers Bridge - A reference parent IP (Bridge IP) that configures Transceivers Wizard parameters. For more information, see IP Integrator (IPI) Design Entry for Custom IP
2. Versal ACAP Transceivers Wizard - A wrapper around GTYE5_QUAD primitive. It consists of single GTYE5 quad (GT quad base IP). Multiple Transceivers Wizards are instantiated for multi-lane (>4 lanes) designs

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) and the *Vivado Design Suite User Guide: Getting Started* (UG910).

Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

IP Integrator (IPI) Design Entry for Custom IP

The design entry for Custom IP is through Bridge IP(`gt_bridge_ip_v1_0`). Custom IPs are not required to be packaged in IPI. You need to generate single or multi Quad design for your requirement using the Bridge IP. Generate the top file and hook it up with the custom IP. The Bridge IP GUI entry is replica of GT Wizard (`gt_quad_base_v1_0`).

You can add `gt_bridge_ip_v1_0` in IPI. Double-click on the IP symbol to open the Bridge IP GUI. Enable **Pass Through Mode** by selecting this option in the GUI. This will bring out essential GT parallel interface to the user interface. Configure other parameters (`gt_quad_base_v1_0` parameters are programmable through `gt_bridge_ip_v1_0` GUI) including number of lanes.

- **Main GUI Configuration Tab:** The Main GUI Configuration tab provides customization options for transceiver preset selection, direction, master clock source, and number of lanes. The Main GUI tab is shown in the following figure:

Figure 2: **GT Bridge IP**

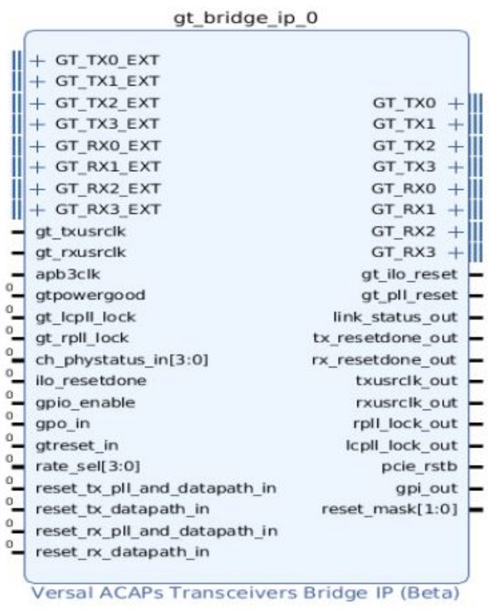
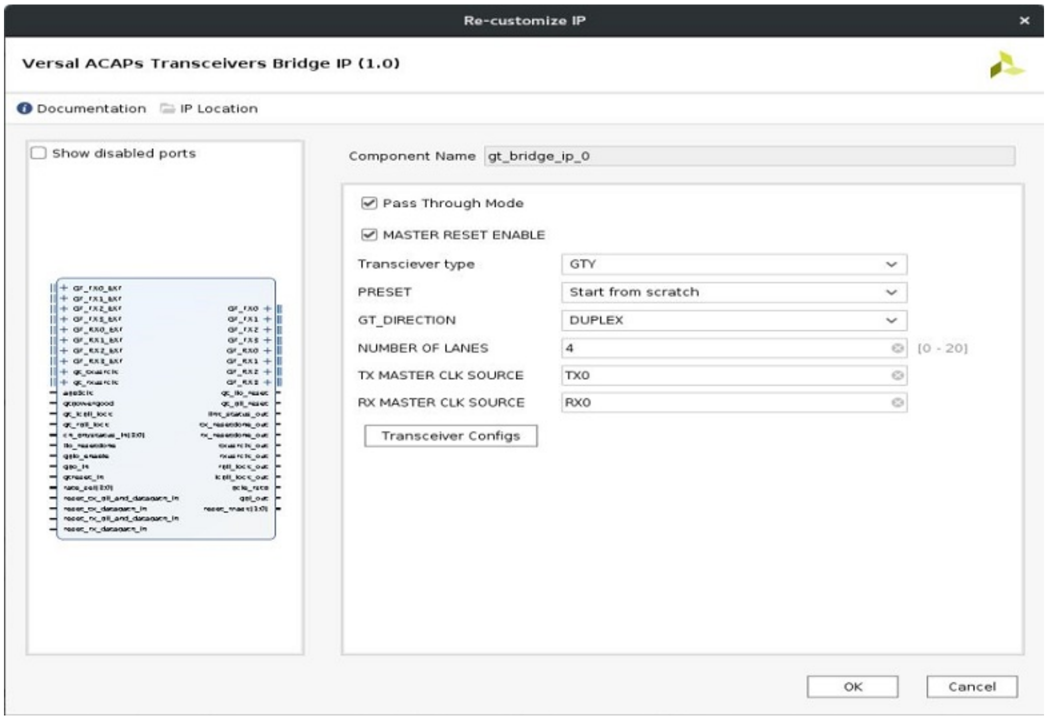


Figure 3: **Bridge IP GUI**



Component Name: gt_bridge_ip_0

☑ Pass Through Mode

☑ MASTER RESET ENABLE

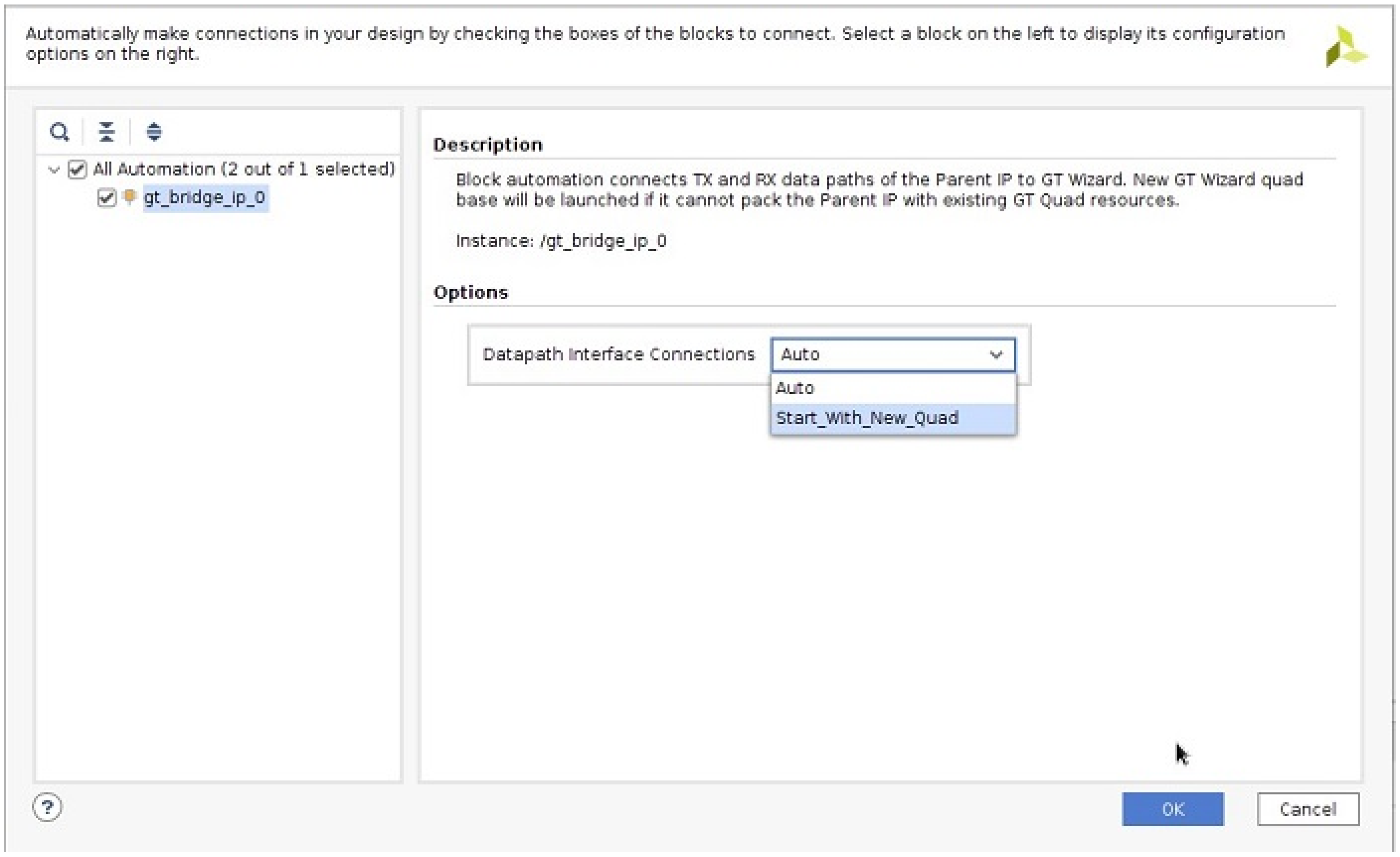
Transciever type	GTY
PRESET	Start from scratch
GT_DIRECTION	DUPLEX
NUMBER OF LANES	4 [0 - 20]
TX MASTER CLK SOURCE	TX0
RX MASTER CLK SOURCE	RX0

- **Component Name:** The name of the generated IP is set in the Component Name field. The default name is `gt_bridge_ip_0`.
- **Pass Through Mode:** Selection of this option brings out essential GT parallel interface to the user interface.
- **MASTER RESET ENABLE:** Enables Master reset looper logic in GT Quad. It's enabled by default. You should not to disable this option.
- **PRESET:** Standard protocol presets are supported. Selecting a preset loads corresponding transceiver settings in sub GUI. multi line rate presets can be loaded in this Preset window. It would populate multiple single line rate presets in corresponding sub GUI. For example multi line rate presets with suffix MLR are added in preset lists.
- **GT_DIRECTION:** Select GT direction options given in the drop-down menu.
- **NUMBER OF LANES:** Select the required number of GT lanes.
- **TX RX Master Clock Source:** Select desired master clock source options given in the dropdown menu.
- **Transceiver Configs:** This option opens a pop up Sub GUI in which transceiver transmitter and receiver settings can be configured.

See Sub GUI Configuration Tab for more information.

Upon clicking Block Automation, the Block Automation GUI opens up. By selecting **Auto**, you let the Block Automation choose the optimal usage of the GT Quad resources. By selecting **Start_With_New_Quad**, the Block Automation instantiates the New GT Quad and makes the data path, clocks, and reset connections. On a fresh IPI canvas, both **Auto** and **Start_With_New_Quad** options behave similarly. It instantiates multiple `gt_quad_base_v1_0` based on the number of lanes configured in `gt_bridge_ip_v1_0` and makes data path, clocks and reset connections as shown below. GT Quads are wrapped inside a level of hierarchy in this case to simplify view

Figure 4: **Bridge IP Block Automation GUI**



Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.

All Automation (2 out of 1 selected)
gt_bridge_ip_0

Description

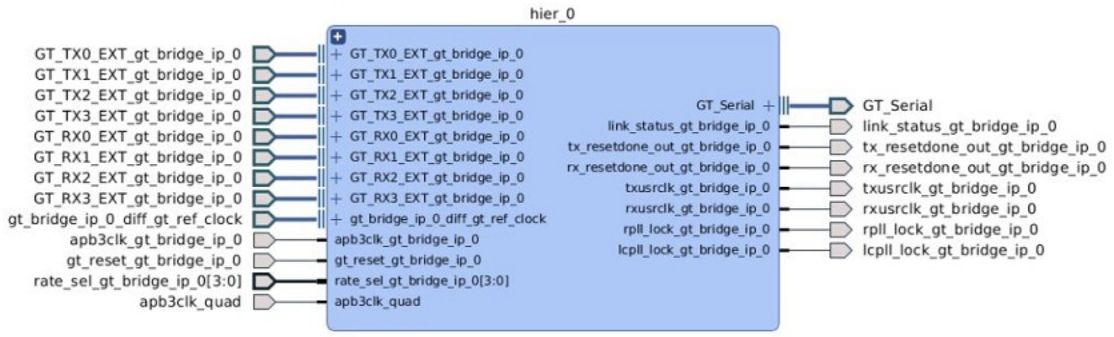
Block automation connects TX and RX data paths of the Parent IP to GT Wizard. New GT Wizard quad base will be launched if it cannot pack the Parent IP with existing GT Quad resources.

Instance: /gt_bridge_ip_0

Options

Datapath Interface Connections: Auto

Figure 5: **Bridge IP with Multi-Quad Design**


hier_0
GT_TX0_EXT_gt_bridge_ip_0
GT_TX1_EXT_gt_bridge_ip_0
GT_TX2_EXT_gt_bridge_ip_0
GT_TX3_EXT_gt_bridge_ip_0
GT_RX0_EXT_gt_bridge_ip_0
GT_RX1_EXT_gt_bridge_ip_0
GT_RX2_EXT_gt_bridge_ip_0
GT_RX3_EXT_gt_bridge_ip_0
gt_bridge_ip_0_diff_gt_ref_clock
apb3clk_gt_bridge_ip_0
gt_reset_gt_bridge_ip_0
rate_sel_gt_bridge_ip_0[3:0]
apb3clk_quad
GT_Serial
link_status_gt_bridge_ip_0
tx_resetdone_out_gt_bridge_ip_0
rx_resetdone_out_gt_bridge_ip_0
txusrclk_gt_bridge_ip_0
rxusrclk_gt_bridge_ip_0
rpll_lock_gt_bridge_ip_0
lcpll_lock_gt_bridge_ip_0

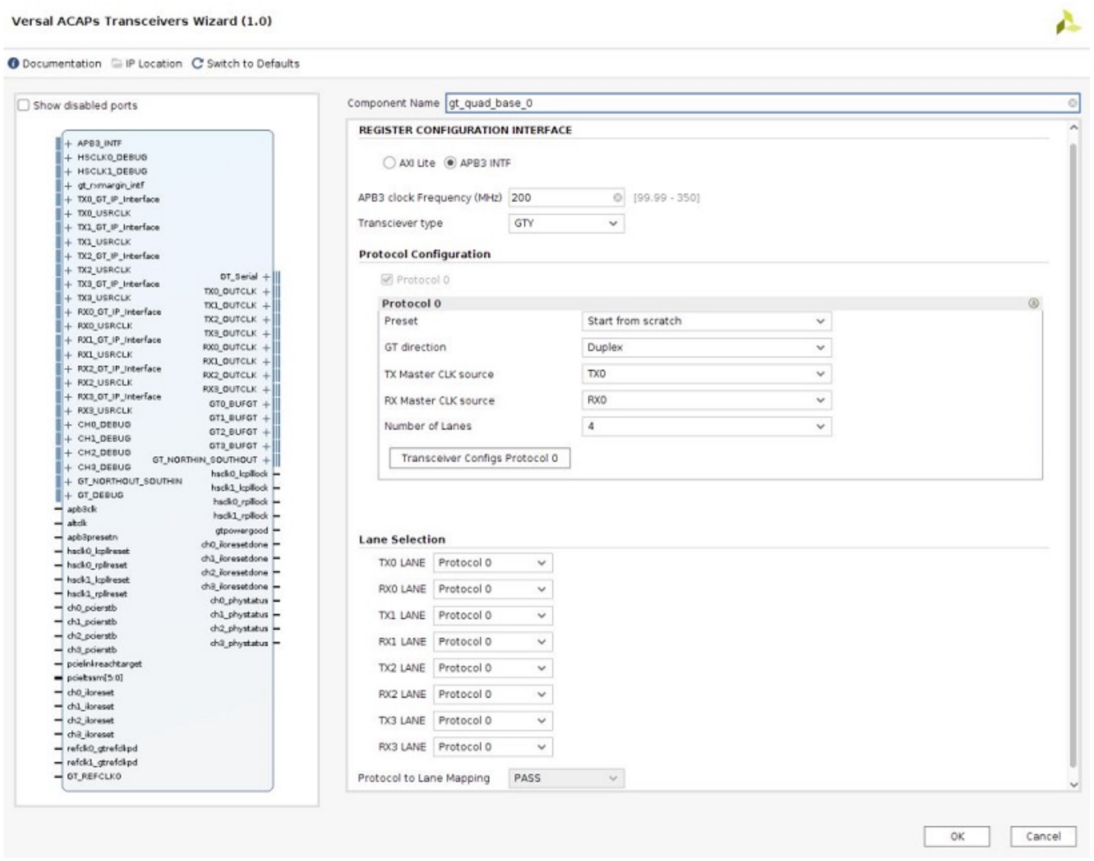
Versal ACAP Transceivers Wizard

The objective of GT quad base IP is to showcase various GT quad features through simulations. Simulation support is provided for one GT quad. You can configure GT quad base IP to share multiple protocol IPs each supporting multiple line rates. You can create example design and simulate to understand dynamic line rate switching and GT quad sharing across multiple Protocol IPs.

Main GUI Configuration Tab

The Main GUI Configuration tab provides customization options for transceiver preset selection, direction, master clock source, and number of lanes. The Main GUI tab is shown in the following figure:

Figure 6: **Main GUI Configuration Tab**



Component Name: gt_quad_base_0

REGISTER CONFIGURATION INTERFACE

AXI Lite [✓ APB3 INTF]

APB3 clock Frequency (MHz): 200 [99.99 - 350]

Transciever type: GTY

Protocol Configuration

[✓ Protocol 0]

Protocol 0

Preset	Start from scratch
GT direction	Duplex
TX Master CLK source	TX0
RX Master CLK source	RX0
Number of Lanes	4

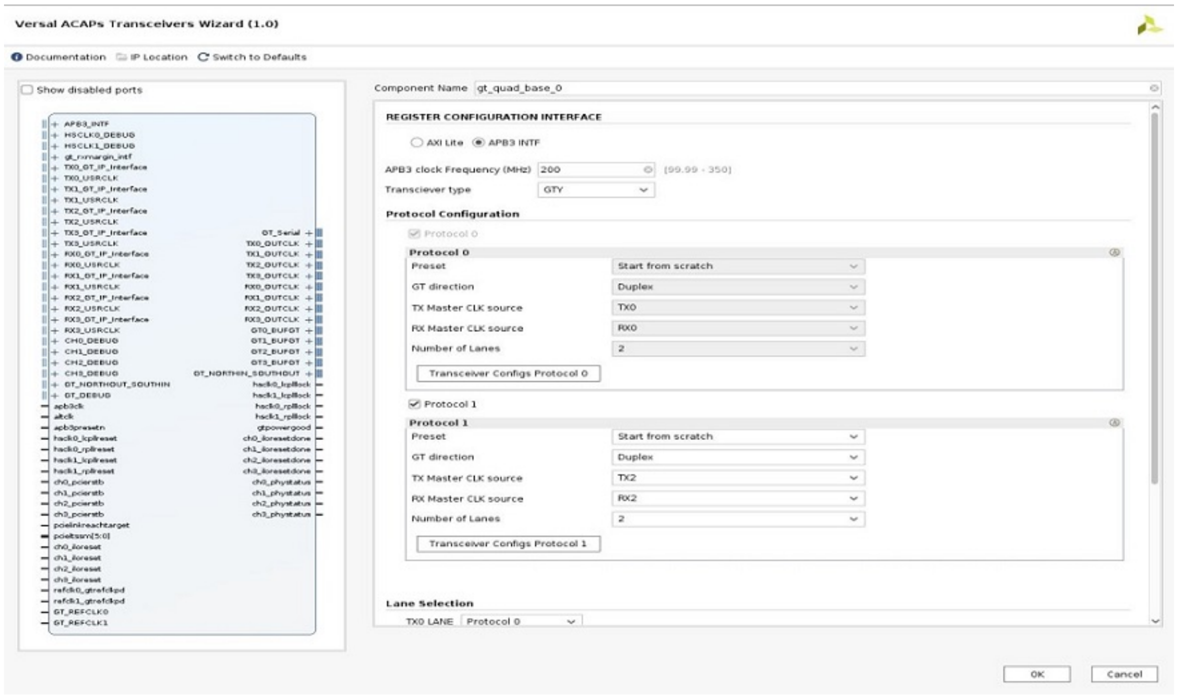
Lane Selection

TX0 LANE	Protocol 0
RX0 LANE	Protocol 0
TX1 LANE	Protocol 0
RX1 LANE	Protocol 0
TX2 LANE	Protocol 0
RX2 LANE	Protocol 0
TX3 LANE	Protocol 0
RX3 LANE	Protocol 0

Protocol to Lane Mapping: PASS

- **Component Name:** The name of the generated IP is set in the Component Name field. The default name is `gt_quad_base_v1_0`.
- **Register Configuration Interface:** Select AxiAXI4-Lite or APB3 Interface for registers configuration. Also enter the frequency of operation of the selected interface.
- **Protocol 0:** This option is enabled by default as four lane duplex design. Based on the number of unused lanes, rest of the protocol options (1, 2, 3 etc.) are enabled. Two protocol options are shown in the following figure, both the protocol options are duplex and 2 lanes each are programmed:

Figure 7: **Multi-Protocol Programming Option**



Component Name gt_quad_base_0

REGISTER CONFIGURATION INTERFACE

AXI Lite [✓ APB3 INTF]

APB3 clock Frequency (MHz) 200 [99.99 - 350]

Transciever type GTY

Protocol Configuration

☑ Protocol 0

Protocol 0

Preset	Start from scratch
GT direction	Duplex
TX Master CLK source	TX0
RX Master CLK source	RX0
Number of Lanes	2

Transceiver Configs Protocol 0

☑ Protocol 1

Protocol 1

Preset	Start from scratch
GT direction	Duplex
TX Master CLK source	TX2
RX Master CLK source	RX2
Number of Lanes	2

Transceiver Configs Protocol 1

Lane Selection

TX0 LANE Protocol 0

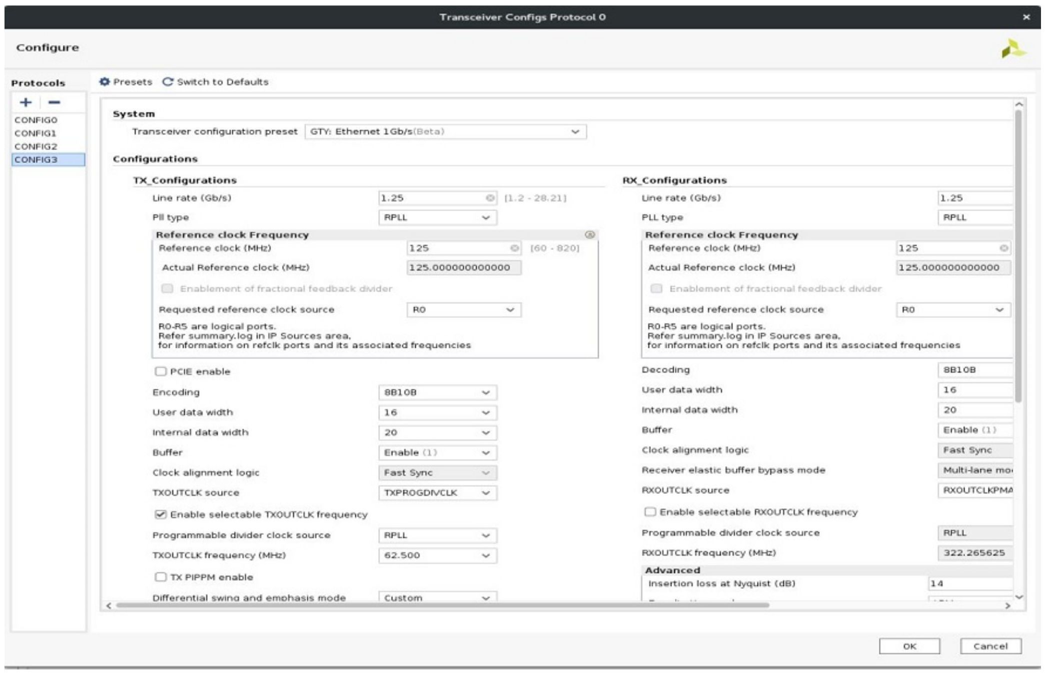
- **Preset:** Standard protocol presets are supported. Selecting a preset will load corresponding transceiver settings in Sub GUI. Multi line rate presets can be loaded in this Preset window. It will populate multiple single line rate presets in corresponding Sub GUI. Example multi line rate presets with suffix MLR are added in preset lists.
- **GT Direction:** Select GT direction options given in the drop-down menu.
- **TX RX Master Clock Source:** Select desired master clock source options given in the drop-down menu.

- **Number of Lanes:** Select number of lanes. Maximum 4 lanes are supported as it's a wrapper of a GT QUAD
- **Transceiver Configs Protocol 0:** This option opens a pop up Sub GUI in which transceiver transmitter and receiver settings can be configured.
- **Lane Selection:** Select physical lane mapping based on "Number of Lanes" value.

Sub GUI Configuration Tab

The Sub GUI configuration tab provides customization options for fundamental transceiver features, including transceiver type, and transmitter and receiver settings. Each Sub GUI configuration corresponds to a target transceiver configuration referred to as CONFIG<0/1...> in the Protocols panel. The target configuration could be achieved by controlling the `ch*_txrate[7:0]` or `ch*_rxrate[7:0]` ports on the wizard. Multiple line rate settings require multiple Sub GUI configurations. The Sub GUI tab is shown in the following figure:

Figure 8: **Sub GUI Configuration Tab**



System

Transceiver configuration preset: GTY: Ethernet 1Gb/s(Beta)

Configurations

TX_Configurations

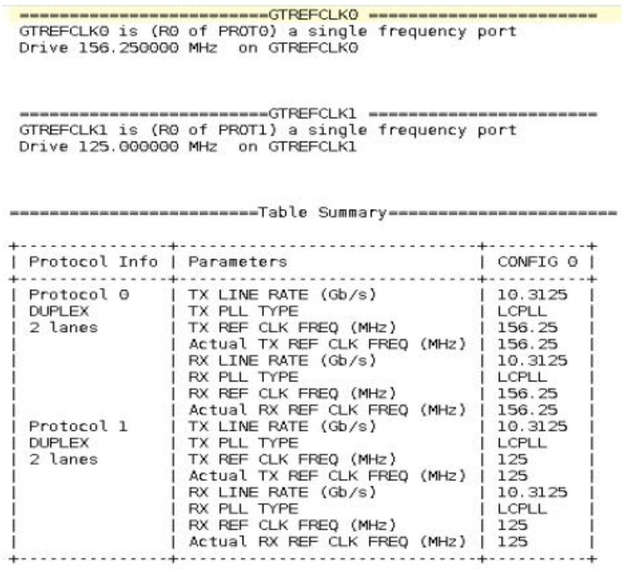
- Line rate (Gb/s): 1.25 [1.2 - 28.21]
- Pll type: RPLL
- Reference clock Frequency
  - Reference clock (MHz): 125 [60 - 820]
  - Actual Reference clock (MHz): 125.000000000000
  - ☐ Enablement of fractional feedback divider
  - Requested reference clock source: R0
  - R0-R5 are logical ports.
    Refer summary.log in IP Sources area,
    for information on refclk ports and its associated frequencies
- ☐ PCIE enable
- Encoding: 8B10B
- User data width: 16
- Internal data width: 20
- Buffer: Enable (1)
- Clock alignment logic: Fast Sync
- TXOUTCLK source: TXPROGDIVCLK
- ☑ Enable selectable TXOUTCLK frequency
- Programmable divider clock source: RPLL
- TXOUTCLK frequency (MHz): 62.500
- ☐ TX PIPPM enable
- Differential swing and emphasis mode: Custom

RX_Configurations

- Line rate (Gb/s): 1.25
- PLL type: RPLL
- Reference clock Frequency
  - Reference clock (MHz): 125
  - Actual Reference clock (MHz): 125.000000000000
  - ☐ Enablement of fractional feedback divider
  - Requested reference clock source: R0
  - R0-R5 are logical ports.
    Refer summary.log in IP Sources area,
    for information on refclk ports and its associated frequencies
- Decoding: 8B10B
- User data width: 16
- Internal data width: 20
- Buffer: Enable (1)
- Clock alignment logic: Fast Sync
- Receiver elastic buffer bypass mode: Multi-lane mo
- RXOUTCLK source: RXOUTCLKPMA
- ☐ Enable selectable RXOUTCLK frequency
- Programmable divider clock source: RPLL
- RXOUTCLK frequency (MHz): 322.265625
- Advanced
  - Insertion loss at Nyquist (dB): 14

- **Protocols:** By clicking "+" button new CONFIGx will be added. Each CONFIG corresponds to a Sub GUI.
- **Transceiver configuration preset:** Standard protocol preset can be loaded from list of presets.
- **Configurations:** GT transceiver and receiver settings like line rate, GT Type, Reference clock frequency can be selected.
- **Reference clock Frequency:** It has user reference clock frequency input. Actual reference clock frequency is calculated based on user reference clock input.
- **Requested reference clock source:** Six logical reference clock inputs from R0 – R5 are given. Based on user selection across various line rates, wizard assigns the sorted-out reference clock ports (GTREFCLK0-5) and frequency to be driven on these clock ports. Summary.log in IP Sources area provides clock port and frequency assignment and Table Summary of CONFIG information. Sample summary.log is given in the following figure:

Figure 9: **Sample Summary Log**

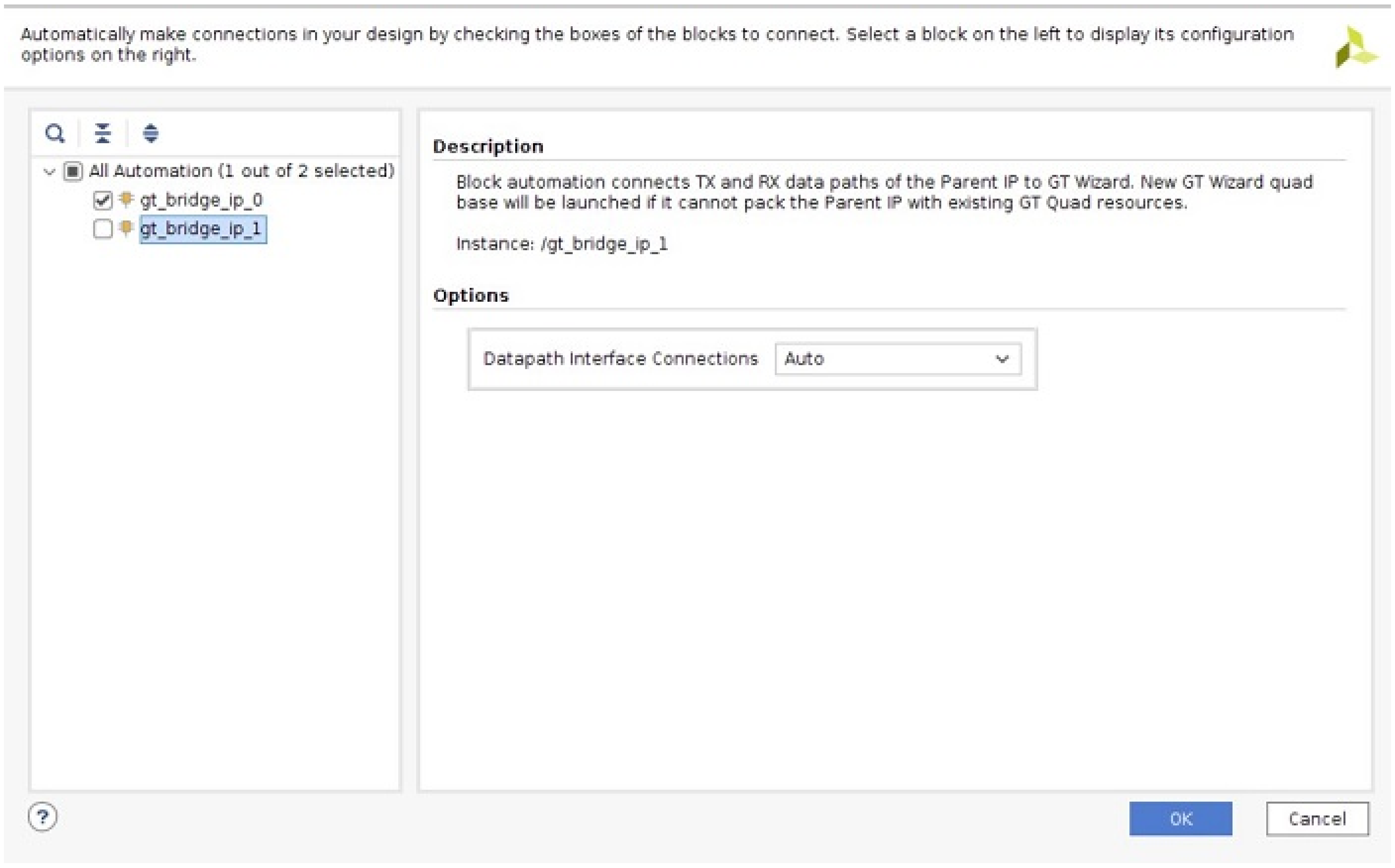


```
=========================GTREFCLK0 ========================
GTREFCLK0 is (R0 of PROT0) a single frequency port
Drive 156.250000 MHz  on GTREFCLK0


=========================GTREFCLK1 ========================
GTREFCLK1 is (R0 of PROT1) a single frequency port
Drive 125.000000 MHz  on GTREFCLK1


=========================Table Summary=======================

+---------------+-------------------------------+----------+
| Protocol Info | Parameters                    | CONFIG 0 |
+---------------+-------------------------------+----------+
| Protocol 0    | TX LINE RATE (Gb/s)           | 10.3125  |
| DUPLEX        | TX PLL TYPE                   | LCPLL    |
| 2 lanes       | TX REF CLK FREQ (MHz)         | 156.25   |
|               | Actual TX REF CLK FREQ (MHz)  | 156.25   |
|               | RX LINE RATE (Gb/s)           | 10.3125  |
|               | RX PLL TYPE                   | LCPLL    |
|               | RX REF CLK FREQ (MHz)         | 156.25   |
|               | Actual RX REF CLK FREQ (MHz)  | 156.25   |
| Protocol 1    | TX LINE RATE (Gb/s)           | 10.3125  |
| DUPLEX        | TX PLL TYPE                   | LCPLL    |
| 2 lanes       | TX REF CLK FREQ (MHz)         | 125      |
|               | Actual TX REF CLK FREQ (MHz)  | 125      |
|               | RX LINE RATE (Gb/s)           | 10.3125  |
|               | RX PLL TYPE                   | LCPLL    |
|               | RX REF CLK FREQ (MHz)         | 125      |
|               | Actual RX REF CLK FREQ (MHz)  | 125      |
+---------------+-------------------------------+----------+
```

GT Quad Sharing Multiple Bridge IPs

It is possible to share a GT Quad with more than one Bridge IP based on their number of lanes, PLL requirements, and other configurations. For example, a GT Quad can be shared by two x2 lane Bridge IPs. To achieve this, you can drop in two Bridge IPs in the IPI canvas, configure Bridge IP GUI, and click **Block Automation**. The Block Automation GUI opens, as shown in the following figure. This figure shows the Block Automation GUI for two Bridge IPs. You can either choose to do Block Automation for both the Bridge IPs simultaneously, or one after another.

Figure 10: **Block Automation for Two Bridge IPs**



Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.

All Automation (1 out of 2 selected)
- [✓] gt_bridge_ip_0
- [ ] gt_bridge_ip_1

Description

Block automation connects TX and RX data paths of the Parent IP to GT Wizard. New GT Wizard quad base will be launched if it cannot pack the Parent IP with existing GT Quad resources.

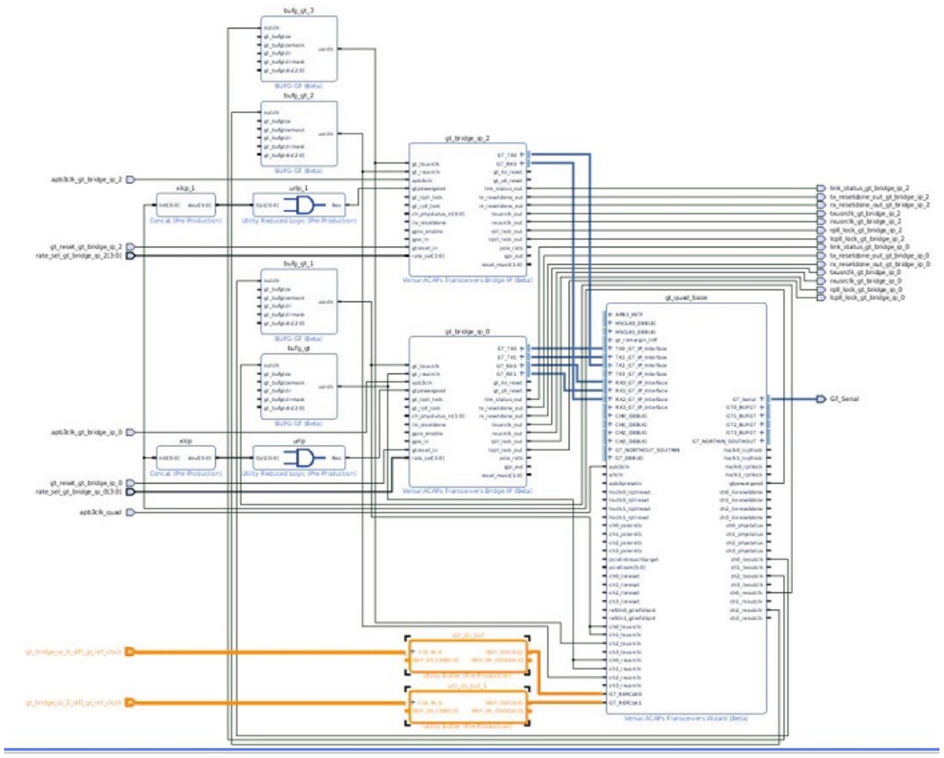
Instance: /gt_bridge_ip_1

Options

Datapath Interface Connections: Auto

The following figure shows the fully connected Bridge IPs shared with a single GT Quad.

Figure 11: **Fully Connected Bridge IP**

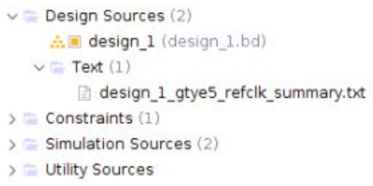


GT Reference Clock Summary and Optimization

As part of block automation, GT Reference Clocks(GTREFCLK) are shorted across quads for simple designs.

- **Example 1:** For a x8 design that use two Quads, GTREFCLKs from both Quads are shorted and connected to single `IBUFDS_GT`. For complex designs using Mulitple Bridge IPs with Multiple Quads, System designers can short GTREFCLKs based on its frequency information, Quad placement and Clock availability on the board. To help System designers to make an informed decision, GTREFCLK summary is provided for the whole system in the canvas. This table can be obtained by entering the following command in Tcl console. `xilinx::designutils::report_gtye5_refclk_summary`Upon executing the command, `gtye5_refclk_summary.txt` is generated in below path

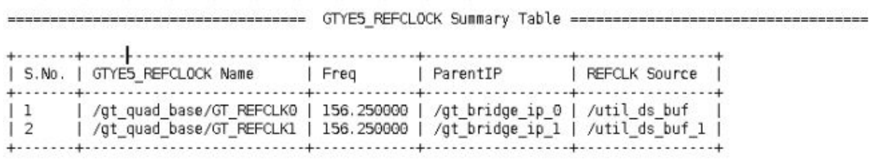
Figure 12: **GTYE5 Reference clock summary file location**



```
Design Sources (2)
    design_1 (design_1.bd)
  Text (1)
      design_1_gtye5_refclk_summary.txt
Constraints (1)
Simulation Sources (2)
Utility Sources
```

It reports GT Reference Clocks, its frequencies and its source in this design shown in figureFigure 11: Fully Connected Bridge IP .

Figure 13: **Example 1 GTYE5 Reference Clock Summary Table**



```
===================================  GTYE5_REFCLOCK Summary Table  ===================================

+-------+-------------------------+------------+----------------+---------------+
| S.No. | GTYE5_REFCLOCK Name     | Freq       | ParentIP       | REFCLK Source |
+-------+-------------------------+------------+----------------+---------------+
| 1     | /gt_quad_base/GT_REFCLK0 | 156.250000 | /gt_bridge_ip_0 | /util_ds_buf   |
| 2     | /gt_quad_base/GT_REFCLK1 | 156.250000 | /gt_bridge_ip_1 | /util_ds_buf_1 |
+-------+-------------------------+------------+----------------+---------------+
```

- **Example 2:** Figure 13 shows four instances of x2 IPs sharing two Quads. In this case, four GTREFCLKs are available in the design. Since these four instances of same IP, GTREFCLK frequencies are same for all IPs. System designer can potentially short these GTREFCLKs if Quads are placed adjacent to each other and use single input pin from board. Figure 14 shows the GTREFCLKs summary of this design

Figure 14: **Multiple Bridge IPs sharing GT Quads**

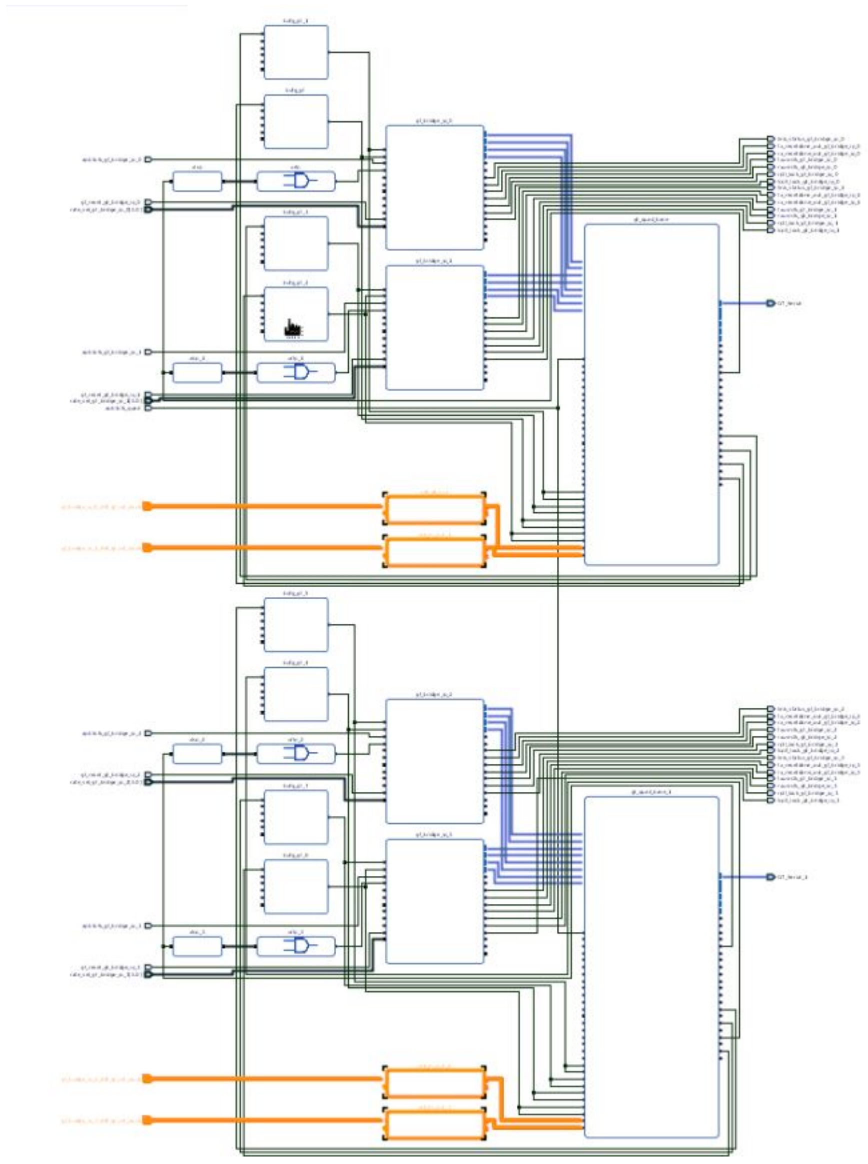
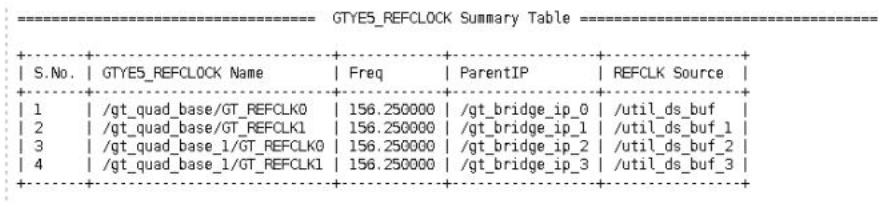


Figure 15: **Example 2 GTYE5 Reference Clock Summary Table**



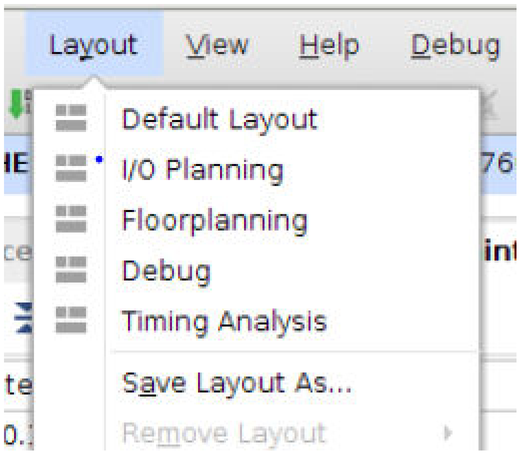
```
==================================  GTYE5_REFCLOCK Summary Table ==================================

+-------+-----------------------------+------------+------------------+-----------------+
| S.No. | GTYE5_REFCLOCK Name         | Freq       | ParentIP         | REFCLK Source   |
+-------+-----------------------------+------------+------------------+-----------------+
| 1     | /gt_quad_base/GT_REFCLK0    | 156.250000 | /gt_bridge_ip_0  | /util_ds_buf    |
| 2     | /gt_quad_base/GT_REFCLK1    | 156.250000 | /gt_bridge_ip_1  | /util_ds_buf_1  |
| 3     | /gt_quad_base_1/GT_REFCLK0  | 156.250000 | /gt_bridge_ip_2  | /util_ds_buf_2  |
| 4     | /gt_quad_base_1/GT_REFCLK1  | 156.250000 | /gt_bridge_ip_3  | /util_ds_buf_3  |
+-------+-----------------------------+------------+------------------+-----------------+
```

Synthesizing and Implementing the Design

Use the following steps once the system is created:

1. Validate the design for correctness. Right-click in the IPI canvas and choose **Validate Design** or press function key F6.
2. Once the design is validated and the top-level file is generated, you can synthesize the design by clicking **Run Synthesis** in Vivado.
3. Open **synthesized design → Layout → I/O Planning for GT** and refclk pin placements as shown in figure below:

   IO Planning

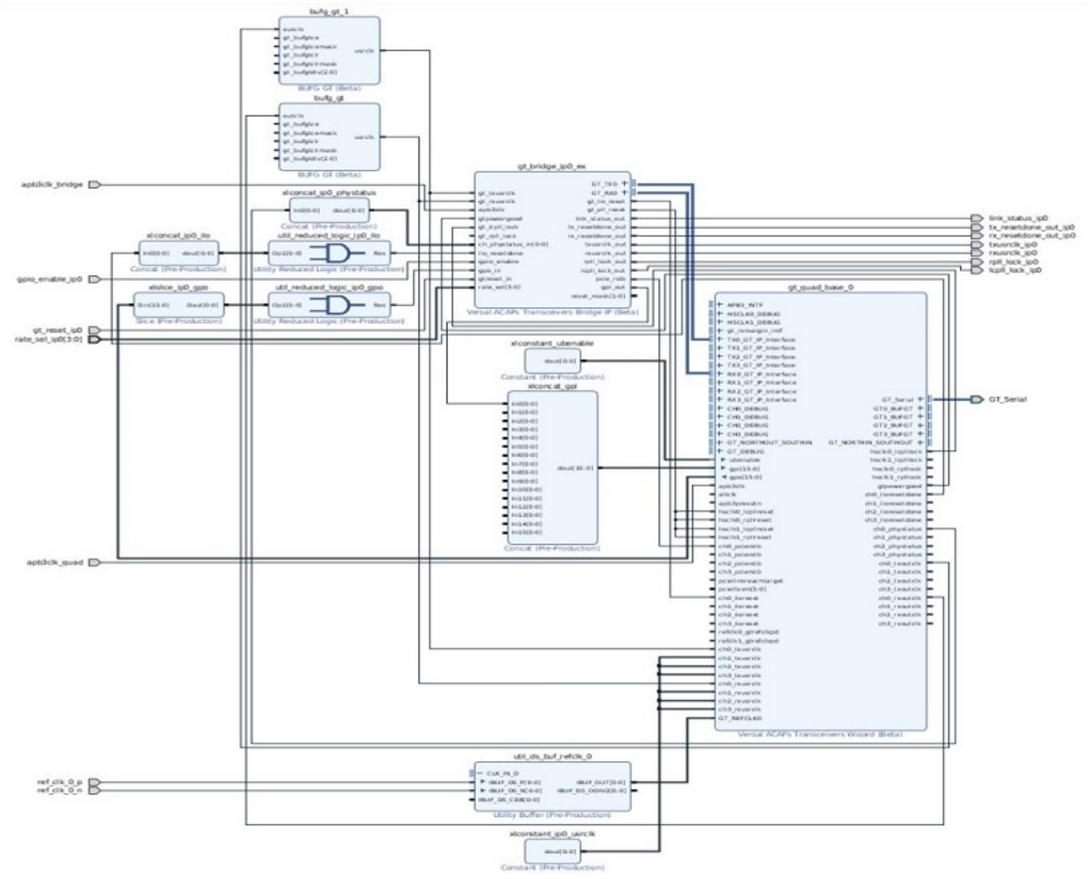
   

4. Open Package Pins tab and provide GT and GT reference clock locations in corresponding MGT banks.
5. Once all the I/O Ports are assigned, click **Run Implementation** to implement the design.

Example Design

An example design can be generated for any customization of the Transceivers Wizard IP core. After you customize and generate a core instance, choose the Open IP Example Design Vivado Integrated Design Environment (IDE) option for that instance. A separate Vivado project opens with the Wizard example design as the top-level module. The example design instantiates the customized core. Example design invokes IPI in the background, adds `gt_bridge_ip_v1_0` and generates quad design for a given configuration as given in the following figure:

Figure 16: **Core Example Design**



The purpose of the Wizard IP example design is to:

- Provide a simple demonstration of the customized core instance operating in simulation through the use of a link status indicator based on PRBS generators and checkers.

- Provide a starting point for integrating the customized core into your system, including reference clock buffers.

The gt_bridge_ip_v1_0 contains configurable PRBS generator and checker modules per transceiver channel that enable simple data integrity testing, and resulting link status reporting. The example design is also synthesizable.

Note**:** You need to provide appropriate GT Quad and GTREFCLK location in IP xdc for a selected device if not using IO Planner for locations selection.

Limitations of the Example Design

The example design is the recommended means of simulating or implementing an instance of the Wizard IP core outside the context of your own system. It is quite simplistic, and the following limitations should be understood:

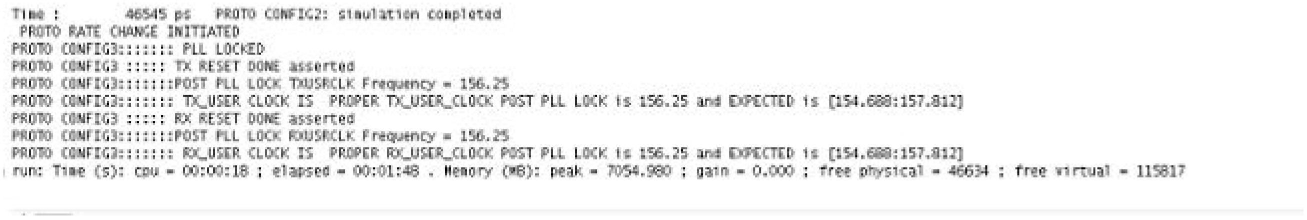
- The example design does not implement specific protocols to generate or check data. Fundamentally, raw PRBS data is generated and checked.
- When the example design is simulated using the provided test bench, each transceiver channel is looped back from the serial data transmitter to the receiver. As such, data integrity can only be properly checked if the transmitter and receiver are configured for the same line rate and to use the same data coding. No rate adjustment schemes are used. If the transmitter and receiver line rates or data coding are configured differently from one another in your system, you might wish to cross-couple two appropriately-customized core instances and check for data integrity in hardware or in your own test bench. In such a setup, the transmitter of core instance A is rate and coding-matched to the receiver of core instance B, and vice versa.
- Example designs are not provided for Multi-Quad designs.

Simulating the Example Design

To simulate an instance of the Wizard IP core, first open its example design.

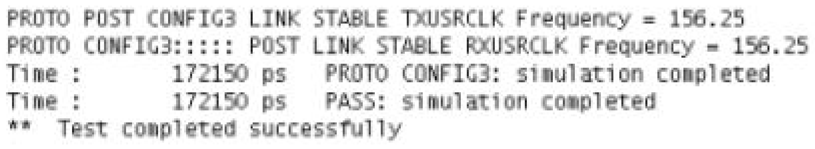
In the example project, start a behavioral simulation by clicking **Run → Simulation → Run Behavioral Simulation** in the Vivado Integrated Design Environment (IDE).

The example design simulation test bench provides the requisite free-running clock and transceiver reference clock signals, as well as a “reset all” pulse to the example design logic and reset controller helper block input ports. This stimulus is sufficient to allow the helper blocks to bring up the remainder of the system. After some time, the transceiver PLL(s) will achieve lock, allowing the reset controller helper block finite state machines to complete the full reset sequence. After the reset sequence is complete, you can begin to observe the example stimulus module transmitting data. A short time later, the example checking module begins to search for data alignment and checks for data integrity, which is in turn used by the link status logic to drive the link status indicator. In case of multi-line rate configurations, example design switches to next line rate and again tries to achieve lock and other Reset Done signals. Finally, it measures the expected user clk frequency to ensure expected line rate is achieved. After every rate change, it prints the following message:



```
Time :        46545 ps   PROTO CONFIG2: simulation completed
 PROTO RATE CHANGE INITIATED
PROTO CONFIG3::::::: PLL LOCKED
PROTO CONFIG3 ::::: TX RESET DONE asserted
PROTO CONFIG3:::::::POST PLL LOCK TXUSRCLK Frequency = 156.25
PROTO CONFIG3::::::: TX_USER CLOCK IS  PROPER TX_USER_CLOCK POST PLL LOCK is 156.25 and EXPECTED is [154.688:157.812]
PROTO CONFIG3 ::::: RX RESET DONE asserted
PROTO CONFIG3:::::::POST PLL LOCK RXUSRCLK Frequency = 156.25
PROTO CONFIG3::::::: RX_USER CLOCK IS  PROPER RX_USER_CLOCK POST PLL LOCK is 156.25 and EXPECTED is [154.688:157.812]
run: Time (s): cpu = 00:00:18 ; elapsed = 00:01:48 . Memory (MB): peak = 7054.980 ; gain = 0.000 ; free physical = 46634 ; free virtual = 115817
```

Finally, the following messages are printed to the transcript and the test is considered to have passed.



```
PROTO POST CONFIG3 LINK STABLE TXUSRCLK Frequency = 156.25
PROTO CONFIG3::::: POST LINK STABLE RXUSRCLK Frequency = 156.25
Time :       172150 ps   PROTO CONFIG3: simulation completed
Time :       172150 ps   PASS: simulation completed
**  Test completed successfully
```

Known Issues

The following issues are identified during this release and will be fixed in future releases:

- When two IPs - Simplex TX and Simplex RX with unequal number of lanes and unequal number of line rates then the configurations may not function properly.
- Few Buffer Bypass cases with more margin might fail with data integrity checks.
- 64b66b async mode with 64-bit user width and 32-bit int width fails in some configurations.
- Summary log has issues after six decimal values of Actual Refclk frequency for some configurations.
- Few Multi IP configurations in shared configurations may not work in simulation.
- Some combinations of line rate/reference clock frequencies could give improper user clock frequencies due to precision issues.
- Some of the configurations with different TX/RX data widths may fail in simulation with PRBS mismatch.

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

***Note*:** For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this guide:

1. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Getting Started* (UG910)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
07/14/2020 Version 1.0	
Initial release	N/A

Please Read: Important Legal Notices

Copyright