# IBERT for UltraScale GTY Transceivers v1.2

## *LogiCORE IP Product Guide*

**Vivado Design Suite**

**PG196 April 5, 2017**

XILINX
ALL PROGRAMMABLE™

# Table of Contents

## Appendix C: Additional Resources and Legal Notices

# Introduction

The LogiCORE™ IP Integrated Bit Error Ratio Test (IBERT) core for UltraScale™/UltraScale+™ architecture GTY transceivers is designed for evaluating and monitoring the GTY transceivers. This core includes pattern generators and checkers that are implemented in FPGA logic, and access to ports and the dynamic reconfiguration port attributes of the GTY transceivers. Communication logic is also included to allow the design to be runtime accessible through JTAG. This core can be used as a self-contained or open design, based on customer configuration and as described in this document.

# Features

- Provides a communication path to the Vivado® Serial I/O Analyzer feature.
- Provides a user-selectable number of UltraScale architecture GTY transceivers.
- Transceivers can be customized for the desired line rate, reference clock rate, and reference clock source.
- Requires a system clock that can be sourced from a pin or one of the enabled GTY transceivers.

| LogiCORE IP Facts Table | |
|---|---|
| **Core Specifics** | |
| Supported Device Family(1) | UltraScale<br>UltraScale+ |
| Supported User Interfaces | N/A |
| Resources | Performance and Resource Utilization web page |
| **Provided with Core** | |
| Design Files | register transfer level (RTL) |
| Example Design | Verilog |
| Test Bench | Not Provided |
| Constraints File | Xilinx Design Constraints (XDC) |
| Simulation Model | Not Provided |
| Supported S/W Driver | N/A |
| **Tested Design Flows(2)** | |
| Design Entry | Vivado Design Suite |
| Simulation | Not Provided |
| Synthesis | Vivado Synthesis |
| **Support** | |
| Provided by Xilinx at the Xilinx Support web page | |

**Notes:**

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

# Overview

The IBERT for UltraScale™ Architecture GTY Transceivers core provides a broad-based physical medium attachment (PMA) evaluation and demonstration platform for UltraScale/ UltraScale+™ architecture GTY transceivers. Parameterizable to use different GTY transceivers and clocking topologies, the core can also be customized to use different line rates, reference clock rates, and logic widths. Data pattern generators and checkers are included for each GTY transceiver, giving several different pseudo-random binary sequence (PRBS) and clock patterns to be sent over the channels.

In addition, the configuration and tuning of the GTY transceivers is accessible through logic that communicates to the dynamic reconfiguration port (DRP) of the GTY transceiver to change attribute settings as well as registers that control the values on the ports. At runtime, the Vivado® Serial I/O Analyzer communicates with the core through JTAG, using the Xilinx cables and proprietary logic that is part of the core.

## Feature Summary

The core is designed for PMA evaluation and demonstration. All the major PMA features of the GTY transceiver are supported and controllable, including:

- TX pre-emphasis and post-emphasis
- TX differential swing
- RX equalization
- Decision feedback equalizer (DFE)
- Phase-locked loop (PLL) divider settings

Some of the physical coding sublayer (PCS) features offered by the transceiver are outside the scope of IBERT, including:

- Clock correction
- Channel bonding
- 8B/10B, 64B/66B, or 64B/67B encoding
- TX or RX buffer bypass

## PLL Configuration

For each serial transceiver channel, there is a ring PLL called channel PLL (CPLL). The UltraScale GTY architecture has two additional shared PLLs per quad, QPLL0 and QPLL1. These PLLs are shared to support high-speed, high-performance, and low-power multi-lane applications.

Figure 1-1 shows a Quad in the UltraScale architecture. The GTYE3_CHANNEL component has the serial transceiver and CPLL units and the GTYE3_COMMON has the quad PLL (QPLL) unit.

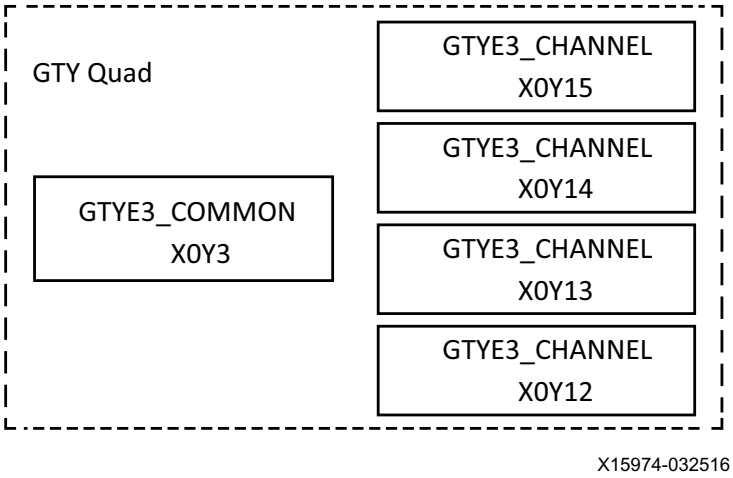


*Figure 1-1:* **Quad in UltraScale Architecture**

The serial transceiver REFCLK can be sourced from either CPLL or QPLL based on multiplexers as shown in Figure 1-2.

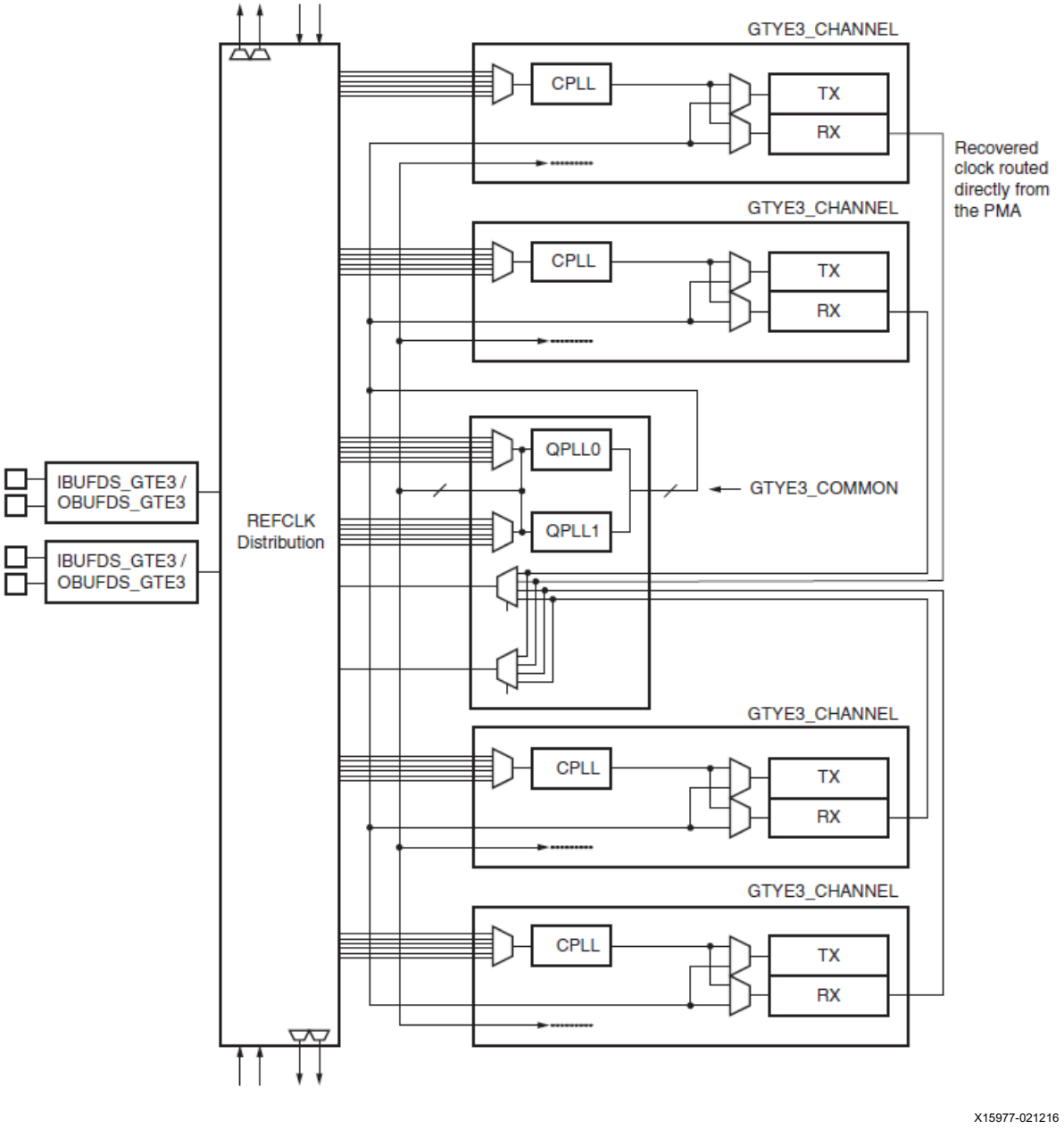


*Figure 1-2:* **Serial Transceiver REFCLK Sourcing**

### Pattern Generation and Checking

Each GTY transceiver enabled in the IBERT design has a pattern generator and a pattern checker. The pattern generator sends data out through the transmitter. The pattern checker accepts data through the receiver and checks it against an internally generated pattern. IBERT offers PRBS 7-bit, PRBS 15-bit, PRBS 23-bit, PRBS 31-bit, Clk 2x (101010...), and Clk 10x (11111111110000000000...) patterns. These patterns are optimized for the logic width that was selected at runtime. The TX and RX patterns are individually selected.

Using the pattern checker logic, the incoming data is compared against a pattern that is internally generated. When the checker receives five consecutive cycles of data with no errors, the LINK signal is asserted. If the LINK signal is asserted and the checker receives five consecutive cycles with data errors, the LINK signal is deasserted. Internal counters accumulate the number of words and errors received.

### DRP and Port Access

GTY transceiver ports and attributes can be changed. The DRP interface logic allows the runtime software to monitor and change any attribute of the GTY transceivers and the corresponding CPLL/QPLL. When applicable, readable and writable registers are also included that are connected to the various ports of the GTY transceiver. All are accessible at runtime using the Vivado Serial I/O Analyzer.

## Applications

The core is designed to be used in any application that requires verification or evaluation of UltraScale architecture GTY transceivers.

## Licensing and Ordering Information

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

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

# Product Specification

## Performance

The core can be configured to run any of the allowable line rates for the GTY transceivers.

### Maximum Frequencies

The core can operate at the maximum user clock frequencies for the FPGA logic width/ speed grade selected. The maximum system clock rate is 100 MHz and the generated design divides any incoming system clock to adhere to this constraint.

## Resource Utilization

For details about performance and resource utilization, visit Performance and Resource Utilization.

## Port Descriptions

The core ports are shown in Table 2-1.

*Table 2-1:* **IBERT I/O Signals**

| Signal Name | I/O | Description |
|---|---|---|
| IBERT_SYSCLOCK_I | I | Clock for all communication logic. This port is present only when an external clock is selected in the generator. |
| $X_iY_j$_TX_N_OPAD[n - 1:0](1) | O | TX differential pairs for each of the *n* GTY transceivers used. |
| $X_iY_j$_TX_P_OPAD[n - 1:0](1) | | |
| $X_iY_j$_RX_N_IPAD[n - 1:0](1) | I | RX differential pairs for each of the *n* GTY transceivers used. |
| $X_iY_j$_RX_P_IPAD[n - 1:0](1) | | |
| $Q_k$_CLK0_MGTREFCLK_I[m - 1:0](2) | I | GTY transceiver reference clocks used.<br>The number of MGTREFCLK ports can be equal to or less than the number of transmit and receive ports because some GTY transceivers can share clock inputs. |
| $Q_k$_CLK1_MGTREFCLK_I[m - 1:0](2) | | |

**Notes:**

1. $X_iY_j$ specifies the GTY transceiver site location.
2. $Q_k$ specifies the GTY transceiver Quad site location.

# Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

## General Design Guidelines

### GTY Transceiver Naming Style

There are two conventions for naming the GTY transceiver, based on the location in the serial transceiver tile in the device. In the XmYn naming convention, m and n indicate the X and Y coordinates of the serial transceiver location. In serial transceiver m_n naming convention, m and n indicate serial transceiver number and the associated quad.

### Line Rate Support

The IBERT core supports a maximum of three different line rates in a single design. For each of these line rates, you can select a custom value based on your requirements. Specify the number of serial transceivers for each line rate that is programmed with these settings. Because usage of QPLL is recommended for line rates above 8 Gb/s, you can select QPLL/CPLL for each line rate falling in the range 0.5 Gb/s to 30.5 Gb/s.

### Serial Transceiver Location

Based on the total number of serial transceivers selected, provide the specific location of each serial transceiver that you intend to use. The region shown in the panel indicates the location of serial transceivers in the tile. This demarcation of region is based on the physical placement of serial transceivers with respect to median of BUFGs available for each device.

# Clocking

## System Clock

The IBERT for UltraScale™ Architecture GTY Transceivers core requires a free-running system clock for communication and other logic that is included in the core. This clock can be chosen at generation time to originate from an FPGA pin, or from a dedicated REFCLK input of one of the GTY transceivers. For the core to operate properly, this system clock source must remain operational and stable when the FPGA is configured with the IBERT for UltraScale/UltraScale+™ GTY transceivers core design.

If the system clock is running faster than 100 MHz, it is divided down internally using a mixed-mode clock manager (MMCM) to satisfy timing constraints. The clock source selected must be stable and free running after the FPGA is configured with the IBERT design. The system clock is used for core communication and as a reference for system measurements. Therefore, the clock source selected must remain operational and stable when using the IBERT for UltraScale/UltraScale+ GTY Transceivers core.

## Receiver Output Clock

The receiver clock probe enable is provided to pull out a recovered clock from any serial transceiver, if desired. When enabled, a new panel appears just before the summary page where you can fill in the serial transceiver source and probe pin standards. The RXOUTCLK and TXOUTCLK signals of each GT transceiver are available at the example design top. The selected RXOUTCLK of a single GT is connected to an external pin.

## Reference Clock

The reference clock source should be provided for all the serial transceivers selected. The drop-down list provides you with possible sources based on local clocks in the same Quad and shared clocks from north/south Quads.

# Resets

Runtime resets are available for the IBERT counters and all GT resets are available. The reset controller helper block from the UltraScale Transceiver Wizard is used to properly time and sequence resets. See the *UltraScale FPGAs Transceivers Wizard LogiCORE IP Product Guide* (PG182) [Ref 1] for details.

# Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and 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) [Ref 2]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 3]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 4]

## Customizing and Generating the Core

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 Vivado 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) [Ref 3] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 4].

***Note:*** Figures in this chapter are illustrations of the Vivado Integrated Design Environment (IDE). The layout depicted here might vary from the current version.

The IBERT for UltraScale™/UltraScale+™ GTY Transceivers core can be found in /Debug & Verification/Debug/ in the Vivado IP catalog.

To access the core name, perform the following:

1. Open a project by selecting **File** then **Open Project** or create a new project by selecting **File** then **New Project**.
2. Open the IP catalog and navigate to any of the taxonomies.
3. Double-click **IBERT UltraScale GTY** to bring up the IBERT Customize IP dialog box.

## Entering the Component Name

The Component Name field can consist of any combination of alphanumeric characters including the underscore symbol. However, the underscore symbol cannot be the first character in the component name.

Figure 4-1 to Figure 4-5 show the IBERT Customize IP dialog boxes with information about customizing ports.

## Protocol Definition

A protocol is a line rate/data width/reference clock rate combination. Up to three protocols can be defined for an IBERT for UltraScale/UltraScale+ GTY Transceivers core, and any number of available Quads can be designated as any protocol defined.

1. Choose the number of protocols desired.
2. In the Protocol combination box, select either **Custom** or a pre-defined protocol. If Custom, type in the line rate (the rate appears in red text if outside the range allowed).
   a. Select the data width.
   b. Choose the REFCLK rate and the number of Quads running at this rate.
3. Changing the line rate entered changes the choices in the REFCLK combination box.
4. The Quad PLL (QPLL0/QPLL1) is selected by default. To select CPLL instead, select **CPLL** from the combination box named PLL.

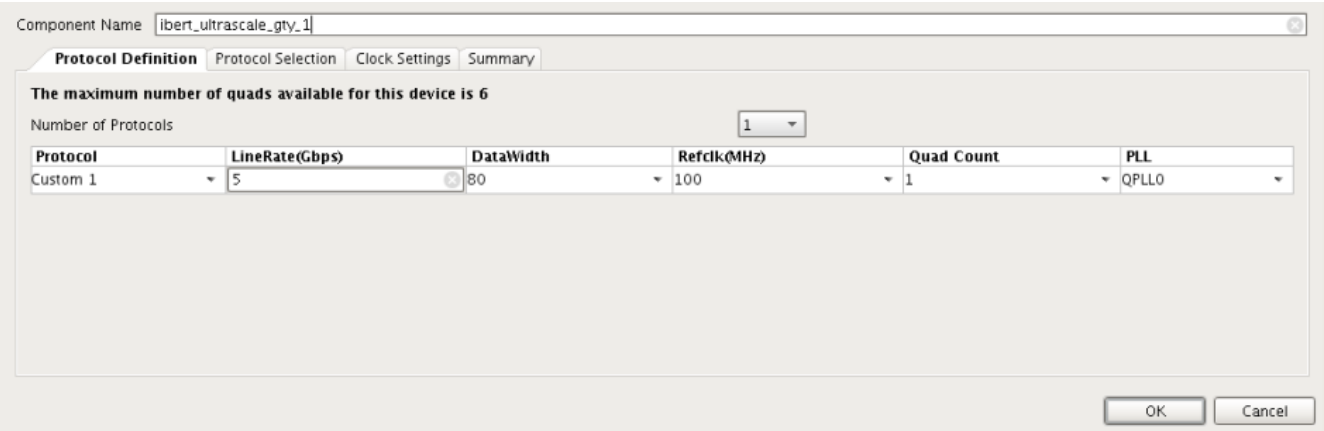


*Figure 4-1:* **Vivado Customize IP Dialog Box – Protocol Definition**

## Advance Settings

Advanced serial transceiver receiver settings are customized by options in each protocol frame (Custom_1, Custom_2, and Custom_3). The number of sections available on this page is equal to the **Number of Protocols** selected under the **Protocol Definition** tab.

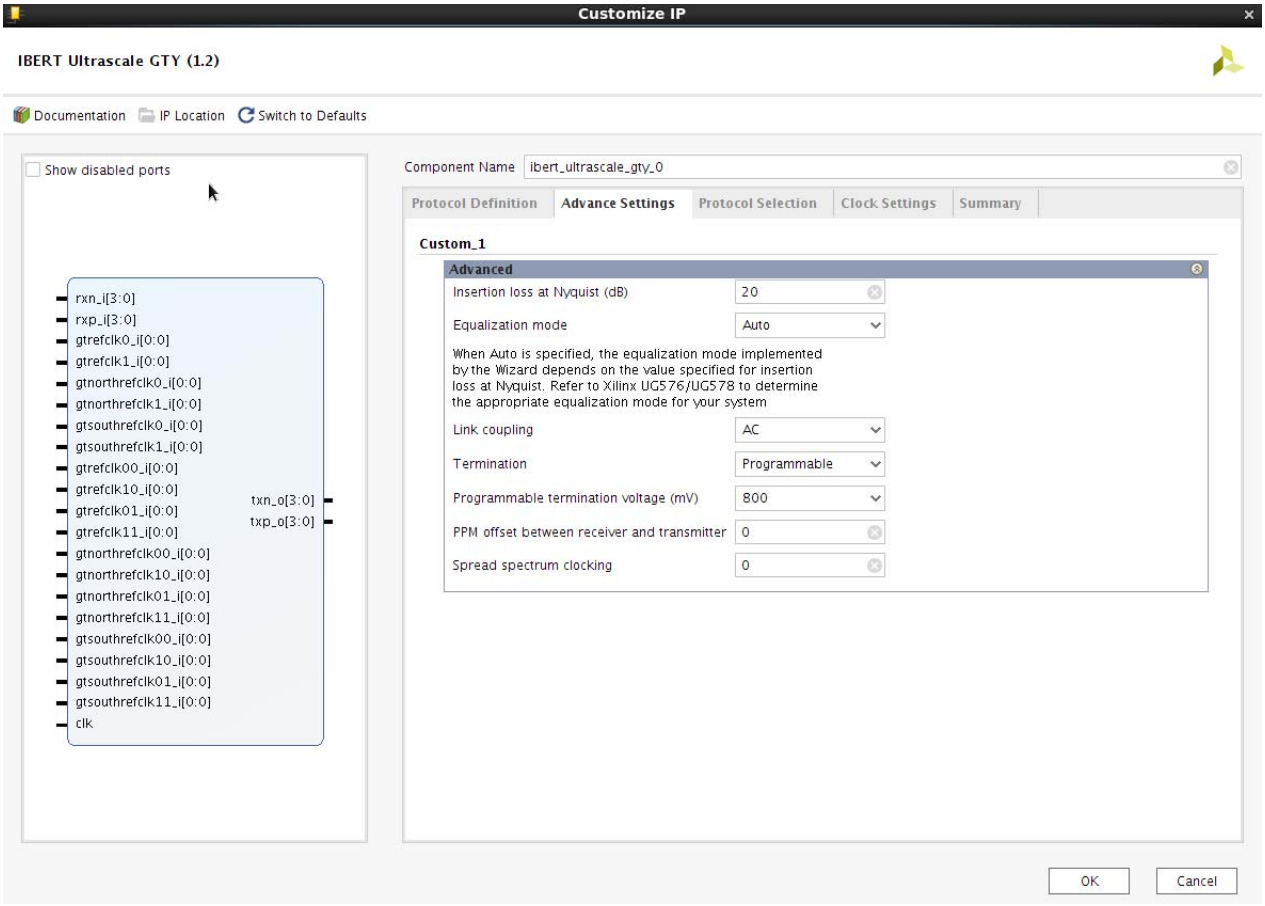


*Figure 4-2:* **Advance Settings**

- **Insertion loss at Nyquist (dB)**. Specify the insertion loss of the channel between the transmitter and receiver at the Nyquist frequency in dB.
- **Equalization mode**. Select between decision feedback equalization (DFE) mode and low-power mode (LPM) for the receiver equalization. When the Auto option is selected, the mode is set automatically based on the channel insertion loss, where a value greater than 14 dB causes DFE to be used; otherwise LPM is used. Refer to the *UltraScale Architecture GTY Transceivers User Guide* (UG578) [Ref 9] for further guidance.
- **Link coupling**. Options are AC and DC. Select AC if external AC coupling is enabled in the application, and DC otherwise.
- **Termination**. Select the receiver termination voltage. Your choice of termination should depend on the protocol and its link coupling.

- **Programmable termination voltage (mV)**. When termination is set to programmable, select the termination voltage in mV.
- **PPM offset between receiver and transmitter**. Specify the offset between received data and transmitter data in PPM. For example, if your protocol specifies ±100 ppm, you would enter 200 in this field. This offset affects the receiver CDR settings.
- **Spread spectrum clocking**. Specify the spread spectrum clocking (SSC) modulation in PPM. SSC affects the receiver CDR settings.

## Protocol Selection

In the Protocol Selection tab, the Quads available in the device/package combination are shown. To allocate a Quad to a specific protocol, select it in the **Protocol Selected** combination. The legal choices for the reference clock input are listed in the REFCLK selection combination. Each channel uses its own TXOUTCLK as TXUSRCLK. The TXUSCLK source selection option is removed (as compared to IBERT 7 series GTH transceiver).

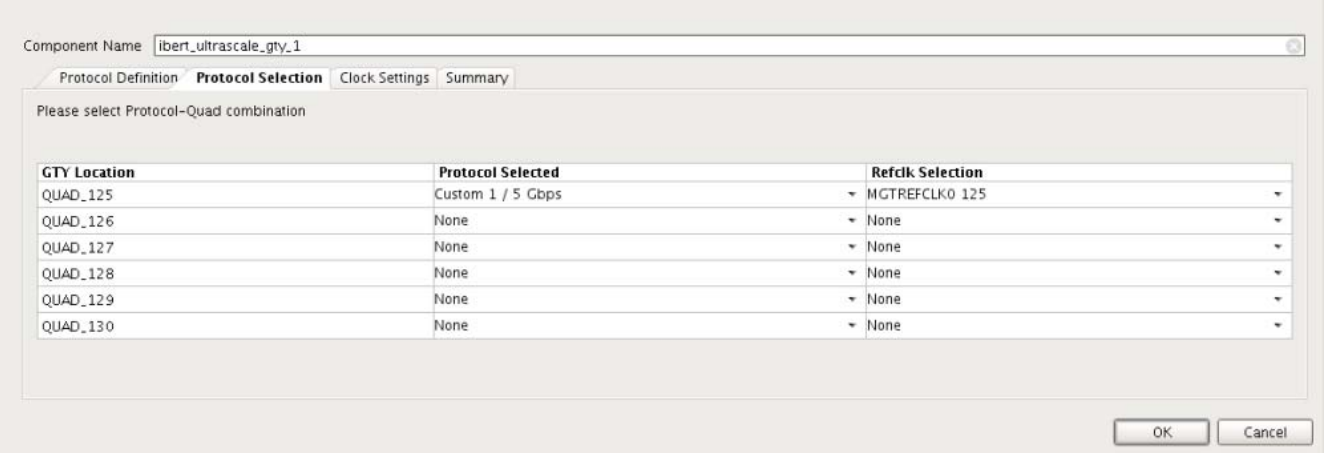


*Figure 4-3:* **Vivado Customize IP Dialog Box – Protocol Selection**

## Clock Settings

In the **Clock Settings** tab, select **Add RXOUTCLK Probes** to drive an output pin or pin pair with the RXOUTCLK of required lane (0 to 3) of the Quad. Select the **I/O Standard** from the list and assign valid pin locations. For System Clock, specify an I/O Standard, valid pin locations, and frequency to complete the system clock settings. Alternatively, any enabled Quad reference clock can be selected instead. In addition to RXOUTCLK, TXOUTCLK is available at the example design top instance.

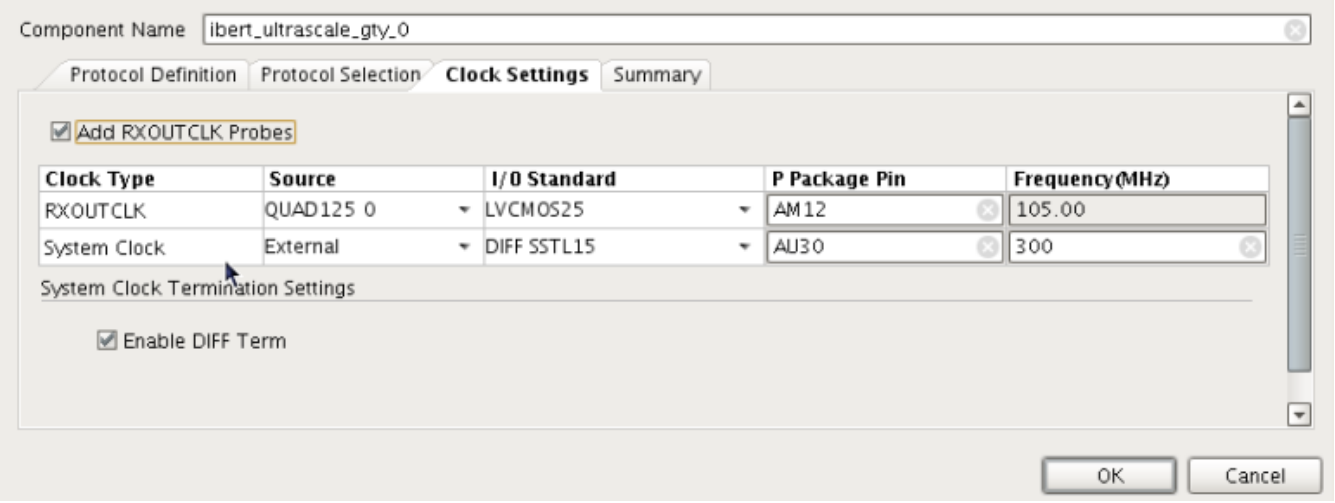


*Figure 4-4:* **Vivado Customize IP Dialog Box – Clock Settings**

## Summary

Review the settings chosen in the Summary page. If they are satisfactory, click **OK** to generate the IBERT for UltraScale/UltraScale+ GTY Transceivers core.

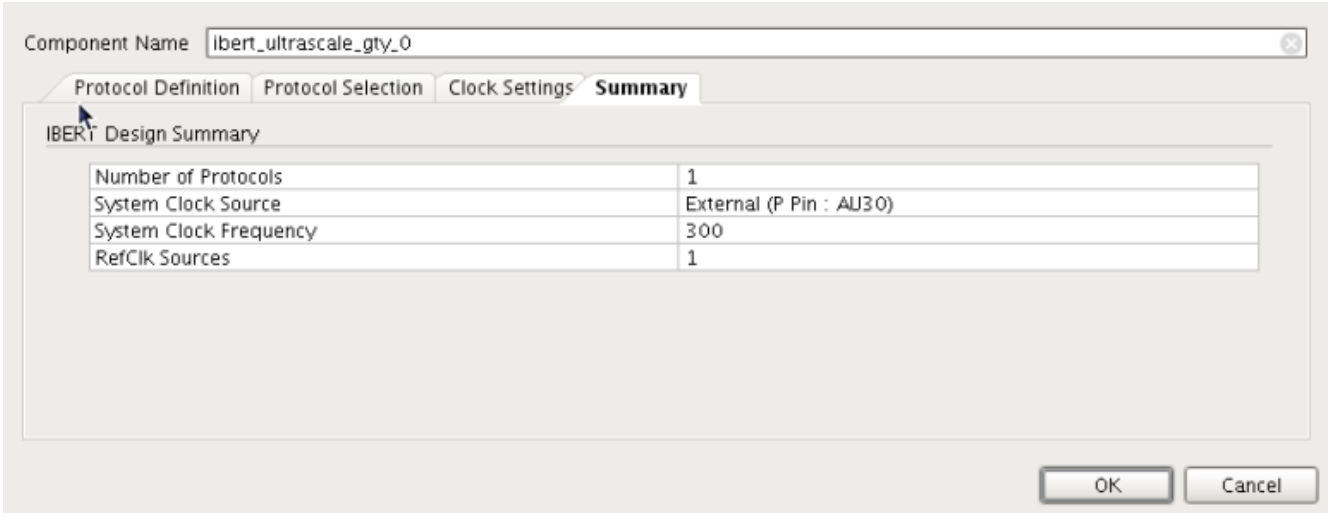


*Figure 4-5:* **Vivado Customize IP Dialog Box – Summary**

## User Parameters

*Table 4-1:* **Vivado IDE Parameter to User Parameter Relationship**

| Vivado IDE Parameter/Value | User Parameter/Value | Default Value |
|---|---|---|
| C REFCLK SOURCE QUAD <n> | C_REFCLK_SOURCE_QUAD_<n> | None |
| C PROTOCOL QUAD <n> | C_PROTOCOL_QUAD<n> | None |
| INS LOSS NYQ <q> | INS_LOSS_NYQ_<q> | 20 |
| RX EQ MODE <q> | RX_EQ_MODE_<q> | AUTO |
| RX COUPLING <q> | RX_COUPLING_<q> | AC |
| RX TERMINATION <q> | RX_TERMINATION_<q> | PROGRAMMABLE |
| RX TERMINATION PROG VALUE <q> | RX_TERMINATION_PROG_VALUE_<q> | 800 |
| RX PPM OFFSET <q> | RX_PPM_OFFSET_<q> | 0 |
| RX SSC PPM <q> | RX_SSC_PPM_<q> | 0 |
| C RXOUTCLK GT LOCATION | C_RXOUTCLK_GT_LOCATION | None |
| C REFCLK SOURCE QUAD <n> | C_REFCLK_SOURCE_QUAD_<n> | None |
| C ENABLE DIFF TERM | C_ENABLE_DIFF_TERM | TRUE |
| C SYSCLK FREQUENCY | C_SYSCLK_FREQUENCY | 300 |
| C SYSCLK IO PIN LOC N | C_SYSCLK_IO_PIN_LOC_N | UNASSIGNED |
| C SYSCLK IO PIN LOC P | C_SYSCLK_IO_PIN_LOC_P | UNASSIGNED |
| C SYSCLK IS DIFF | C_SYSCLK_IS_DIFF | 1 |
| C SYSCLOCK SOURCE INT | C_SYSCLOCK_SOURCE_INT | External |
| C SYSCLK MODE EXTERNAL | C_SYSCLK_MODE_EXTERNAL | 1 |
| C PROTOCOL QUAD COUNT <q> | C_PROTOCOL_QUAD_COUNT_<q> | 1 |
| C PROTOCOL REFCLK FREQUENCY <q> | C_PROTOCOL_REFCLK_FREQUENCY_<q> | 100 |
| C PROTOCOL MAXLINERATE <q> | C_PROTOCOL_MAXLINERATE_<q> | 5 |
| C PROTOCOL COUNT | C_PROTOCOL_COUNT | 1 |
| C GT TYPE | C_GT_TYPE | GTY |
| Component_Name | Component_Name | ibert_ultrascale_gth_v1_3 |
| C RXOUTCLK FREQUENCY | C_RXOUTCLK_FREQUENCY | 105 |

**Notes:**

1. <n>' is range from '0 to 35' and '<q>' is range from '1 to 3'

### Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 3].

## Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

### Required Constraints

The IBERT for UltraScale Architecture GTY Transceivers core is generated with its own timing and location constraints, based on the choices made when customizing the core. No additional constraints are required.

### Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

### Clock Frequencies

This section is not applicable for this IP core.

### Clock Management

This section is not applicable for this IP core.

### Clock Placement

This section is not applicable for this IP core.

### Banking

This section is not applicable for this IP core.

### Transceiver Placement

This section is not applicable for this IP core.

### I/O Standard and Placement

This section is not applicable for this IP core.

# Simulation

The IBERT for UltraScale/UltraScale+ GTY Transceivers core does not support simulation.

# Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 3].

## Interacting with Tcl Commands

After the design is loaded into the device, a set of `hw_sio` commands interact with the IBERT for UltraScale Architecture GTY Transceivers core. See the *Generating an IBERT Core using the Vivado IP Catalog* chapter in *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 5] for more details on Tcl commands.

# Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite.

## Purpose of the Example Design

An example design can be generated for any customization of the UltraScale™ and UltraScale+™ IBERT core. After you have customized and generated a core instance, right-click the generated core and select **Open IP Example Design** in the Vivado IDE for that instance. A separate Vivado project opens with the IBERT example design as the top-level module. The example design instantiates the customized core. The recommended and supported flow is to use the example design as-is, without modifications outside the Vivado IDE.

The purpose of the IBERT IP example design is to:

- Provide a quick demonstration of the customized core instance operating in hardware through the use of a link status indicator based on PRBS generators and checkers which are part of core and generated during IP generation.
- Provide a system which includes reference clock buffers and example system-level constraints.
- Speed up hardware bring-up and debug through the inclusion of a pattern generator and checker.

The example design 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 so it can be used to check for data integrity and hardware links, either through loopback or connection to a suitable link partner. All key status signals, driving basic control signals, and hardware I/O interaction can be done using the Serial I/O Analyzer from the Vivado Hardware Manager after downloading the example design generated bit file.

# Migrating and Upgrading

This appendix contains information about migrating a design from the ISE® Design Suite to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

## Migrating to the Vivado Design Suite

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 6].

## Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when upgrading to a more current version of this IP core in the Vivado Design Suite.

### Parameter Changes

No changes.

### Port Changes

No changes.

### Other Changes

No changes.

# Debugging

This appendix includes details about resources available on the Xilinx Support web page and debugging tools. In addition, this appendix provides a step-by-step debugging process to guide you through debugging the IBERT for UltraScale™/UltraScale+™ GTY Transceivers core.

## Finding Help on Xilinx.com

To help in the design and debug process when using the IBERT for UltraScale/UltraScale+ GTY Transceivers core, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

### Documentation

This product guide is the main document associated with the IBERT for UltraScale/ UltraScale+ GTY Transceivers core. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

### Known Issues

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily, ensuring that users have access to the most accurate information available.

Answer Records for this core are listed below, and can also be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as:

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

**Master Answer Record for the IBERT for UltraScale/UltraScale+ GTY Transceivers Core**

AR 54607

## Technical Support

Xilinx provides technical support at the Xilinx Support web page for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

# Debug Tools

There are many tools available to address IBERT for UltraScale/UltraScale+ GTY Transceivers design issues. It is important to know which tools are useful for debugging various situations.

## Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx.

The Vivado logic analyzer is used to interact with the logic to debug LogiCORE IP cores, including:

- ILA 4.0 (and later versions)
- VIO 3.0 (and later versions)

## License Checkers

If the IP requires a license key, the key must be verified. The Vivado design tools have several license check points for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with an error. License checkpoints are enforced by the following tools:

- Vivado synthesis
- Vivado implementation
- Bitstream generation

**IMPORTANT:** *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

# Additional Resources and Legal Notices

## Xilinx Resources

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

## References

These documents provide supplemental material useful with this product guide:

1. *UltraScale FPGAs Transceivers Wizard LogiCORE IP Product Guide* (PG182)
2. *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994)
3. *Vivado Design Suite User Guide, Designing with IP* (UG896)
4. *Vivado Design Suite User Guide: Getting Started* (UG910)
5. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
6. *ISE to Vivado Design Suite Migration Guide* (UG911)
7. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
8. *Vivado Design Suite User Guide: Implementation* (UG904)
9. *UltraScale Architecture GTY Transceivers User Guide* (UG578)

# Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|---|---|---|
| 04/05/2017 | 1.2 | Added Advance Settings and User Parameters sections to Chapter 4. |
| 06/08/2016 | 1.2 | Updated to include Example Design flow. |
| 04/06/2016 | 1.2 | Added support for UltraScale+™ devices. |
| 09/30/2015 | 1.2 | Added Kintex® UltraScale™ support. Updated clocking RXOUTCLK and TXOUTCLK signals. |
| 04/01/2015 | 1.1 | Updated GUI screens. Updated Line Rate Support. |
| 10/01/2014 | 1.0 | Initial Xilinx release. |

# Please Read: Important Legal Notices