In-System IBERT v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG246 October 4, 2017

Table of Contents

Introduction

The LogiCORE In-System IBERT IP enables 2D eye scans of Ultrascale™/Ultrascale+™ transceivers to be performed in Vivado Serial I/O Analyzer tool. It utilizes data from the user design to plot the eye scans of transceivers in real-time while they interact with the rest of the system. This IP can be integrated with the user logic in the design or Xilinx transceiver-based IPs such as GT Wizard, Aurora, etc. This document details the IP functionality and different ways for adding it to the user design.

Features

- Provides a communication path to the Vivado® Serial I/O Analyzer feature.
- Utilizes data from user design to scan and measure the eye.
- Provides access to DRP and selected Transceiver ports.
- Requires a system clock that can be sourced from a pin or one of the enabled transceivers.

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	UltraScale™, UltraScale+™
Supported User Interfaces	N/A
Resources	See Table 2-1
Provided with Core	
Design Files	RTL
Example Design	Verilog: GT Wizard Example Design
Test Bench	Not Provided
Constraints File	XDC
Simulation Model	Not Provided
Supported S/W Driver	N/A
Tested Design Flows[2]	
Design Entry	Vivado® Design Suite and IP Integrator (IPI)
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 In-System IBERT (ISI) core provides RX margin analysis through eye scan plots on the RX data of transceivers in UltraScale™ and UltraScale+™ devices. The core is parameterizable to use different types of transceivers (GTH or GTY), and select targeted transceiver channels.

The configuration and tuning of the GTH/GTY transceivers is accessible though logic which communicates with the Dynamic Reconfiguration Port (DRP) of the transceivers, to change attribute settings, as well as registers that control the values on the following ports: `rxrate`, `rxlpmen`, `txdiffctrl`, `txpostcursor` and `txprecursor`.

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 to plot eye scans by accessing PMA attributes and ports. The PMA features of the GTH/GTY transceivers are supported and controllable, including:

- TX pre-emphasis and post-emphasis
- TX differential swing
- RX equalization

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

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

DRP and Port Access

GT attributes can be changed via the DRP interface logic of the IP. This allows the runtime software to monitor and change any attribute of the GTH/GTY transceivers.

In addition, you can optionally change the values of `rxrate`, `rxlpmen`, `txdiffctrl`, `txpostcursor` and `txprecursor` ports of transceivers via registers inside the IP which are connected to these optional ports. Both DRP and optional ports are accessible at runtime using the Vivado Serial I/O Analyzer.

Applications

The core is designed to be used in any applications which require eye scan and sweep functionality for UltraScale/UltraScale+ GTH/GTY transceivers.

Licensing and Ordering

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 In System IBERT (ISI) core is the supported method for plotting 2D eye scans and sweep tests using one or more serial transceivers in a Xilinx® UltraScale™ or UltraScale+™ device. In addition to eye scan plot and sweep, this core also provides access to DRP ports and selected transceiver ports. These concepts, as well as technical specifications, are described in this chapter.

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 250 MHz.

Resource Utilization

Table 2-1 provides information on resource utilization for the ISI core.

Table 2-1: **Resource Utilization**

Configuration	LUTs	FFs	Carry8 F7	MUX	BRAM
One Channel enabled	1013	1111	11	32	17

Port Descriptions

The core ports are shown in Table 2-2.

Table 2-2: **ISI I/O Signals**

Sr No.	Name	I/F	I/O	Width	Clk Domain	Optional	Description
1	gt<n>_drpaddr_o	DRP	O	GTHE3:9 GTYE3, GTHE4, GTYE4:10	Clk	No	Should be connected to DRPADDR on transceiver channel primitives
2	gt<n>_drpen_o		O	1	Clk	No	Should be connected to DRPEN on transceiver channel primitives
3	gt<n>_drpdi_o		O	16	Clk	No	Should be connected to DRPDI on transceiver channel primitives
4	gt<n>_drpwe_o		O	1	Clk	No	Should be connected to DRPWE on transceiver channel primitives
5	gt<n>_drpdo_i		I	16	Clk	No	Should be connected to DRPDO on transceiver channel primitives
6	gt<n>_drprdy_i		I	1	Clk	No	Should be connected to DRPRDY on transceiver channel primitives
7	drpclk_o	None	O	1×Num channels	N/A	No	This can be used as the DRPCLK input to transceiver if required. If source for 'clk' port of in_system_ibert and DRPCLK port of GT is the same, then no need to connect this port to DRPCLK of GT channel.
8	eyescanreset_o	None	O	1×Num channels	Clk	No	Should be connected to EYESCANRESET on transceiver channel primitives
9	rxrate_o	None	O	3×Num channels	rxoutclk_i	No	Should be connected to RXRATE on transceiver channel primitives

Table 2-2: **ISI I/O Signals *(Cont'd)***

Sr No.	Name	I/F	I/O	Width	Clk Domain	Optional	Description
10	txdiffctrl_o	None	O	GTHE3: 4×Num channels GTYE3, GTHE4, GTYE4: 5×Num channels	Clk	No	Should be connected to TXDIFFCTRL on transceiver channel primitives
11	txprecursor_o	None	O	5×Num channels	Clk	No	Should be connected to TXPRECURSOR on transceiver channel primitives
12	txpostcursor_o	None	O	5×Num channels	Clk	No	Should be connected to TXPPOSTCURSOR on transceiver channel primitives
13	rxlpmen_o	None	O	1×Num channels	rxoutclk_i	No	Should be connected to RXLPMEN on transceiver channel primitives
14	clk	None	I	1	N/A	No	System clock. Same clock is given as drpclk_o
15	rxoutclk_i	None	I	1×Num channels	N/A	No	Connect RXUSRCLK2 i.e RXOUTCLK of transceiver channel. This clock is used for synchronization. If RXUSRCLK2/RXOUTCLK is not available at interface then connect common user clock or txusrclk or DRP clock.
16	drpclk_i	None	I	1×Num channels	N/A	Yes	Connect clock required to drive DRP transaction or clock connected to DRPCLK port of GT. This clock is used for synchronizing DRP transactions from ISI to GT if same clock is not connected to 'clk' port of ISI and DRPCLK port of GT. This port is required in applications where multiple GTs are running at different DRPCLK clock speeds and have a single ISI instance.

Table 2-2: **ISI I/O Signals *(Cont'd)***

Sr No.	Name	I/F	I/O	Width	Clk Domain	Optional	Description
17	rxrate_i	None	I	3×Num channels	rxoutclk_i	Yes	User accessible port. connected to RXRATE on transceiver channel primitives
18	txdiffctrl_i	None	I	GTHE3: 4×Num channels GTYE3, GTHE4, GTYE4: 5×Num channels	Clk	Yes	User accessible port. connected to TXDIFFCTRL on transceiver channel primitives
19	txprecursor_i	None	I	5×Num channels	Clk	Yes	User accessible port. connected to TXPRECURSOR on transceiver channel primitives
20	txpostcursor_i	None	I	5×Num channels	Clk	Yes	User accessible port. connected to TXPOSTCURSOR on transceiver channel primitives
21	rxlpmen_i	None	I	1×Num channels	rxoutclk_i	Yes	User accessible port. connected to RXLPMEN on transceiver channel primitives

Designing with the Core

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

General Design Guidelines

GT Type selection

Since the ISI core supports both GTH and GTY transceivers, you can select the GT type based on the availability of transceiver type in the selected device.

GTY Transceiver Naming Style

There are two naming conventions for the GTY transceiver, based on the location in the serial transceiver tile in the device:

- **XmYn** - where m and n indicate the X,Y coordinates of the serial transceiver location.
- **m_n** - where m and n indicate serial Transceiver number and the associated Quad number respectively.

Serial Transceiver Location

Ports and DRP interfaces are determined by the total number of serial transceivers selected.

Based on the total number of serial transceivers selected, you 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.

The ISI core has no transceiver location constraints, nor are any attributes updated for selected transceivers. The selected transceiver information is only used to create an ISI template and group/display the selected transceiver in the Serial IO analyzer after downloading the bit file.

Clocking

System Clock

The ISI core requires a free-running system clock for communication and clocking of other logic that is included in the core. This clock can be selected at generation time to originate from an FPGA pin, or from a dedicated `REFCLK` input of one of the GTH/GTY transceivers. For the core to operate properly, this system clock source must remain operational and stable when the FPGA is configured with the ISI core design. This system clock is connected to the `clk` port of the ISI and has a maximum frequency of 250 MHz.

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 ISI core.

Receiver Clock

This clock is used to synchronize the rxrate value. `RXUSRCLK2` should be connected to the `rxoutclk_i` port of the ISI instance.

DRP Clock

The DRP clock is an optional input clock for ISI which is connected to the `drpclk_i` port. The clock is a vectored port, and its width depends on the number of channels/lanes selected. The DRP clock is used to synchronize all DRP transaction in the DRP clock domain inside the ISI before it is sent over the transceiver.

The clock is required when the design or application has multiple transceivers using different DRP clocks (connected to `DRPCLK` port of GT). In this case, each DRP clock source should be connected to the `drpclk_i` [*:0] port. In ISI, all the required data or drive values are generated on the system clock (`clk`) and then synchronized with their respective DRP clocks and sent to the corresponding GT.

Resets

The ISI cores uses the `EYESCANRESET` port of the transceiver. This port is user accessible and can be set through the Serial IO Analyzer tool or the TCL command.

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 IDE. The layout depicted here might vary from the current version.

The ISI 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 **In-System IBERT** to bring up the Customize IP dialog box.

Entering the Component Name

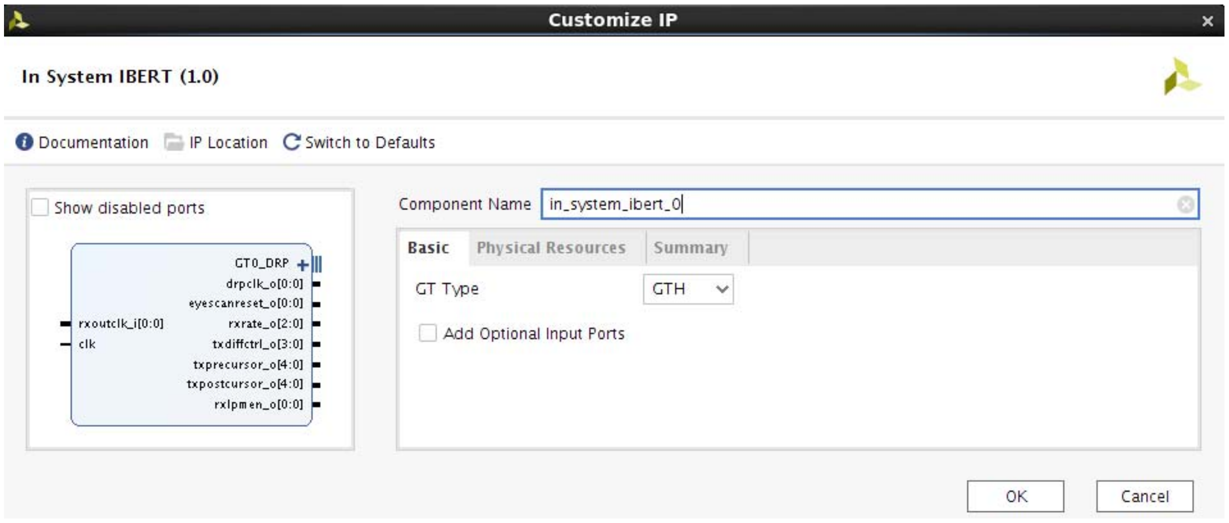
The **Component Name** field can consist of any combination of alphanumeric characters including the underscore symbol. However, the underscore symbol cannot be the first character in the component name.

Figure 4-1 to Figure 4-3 show the In-System IBERT Customize IP dialog boxes with information about customizing ports.

The ISI IP Parameters are organized into three tabs, Basic, Physical Resources and Summary. These are described below.

Basic Tab

The basic tab (Figure 4-1) provides the basic configuration options available.


Customize IP

In System IBERT (1.0)

Show disabled ports ☐

Component Name: in_system_ibert_0

GT Type: GTH

☐ Add Optional Input Ports

Figure 4-1: **Basic Tab**

Component Name - The name of the generated IP is set in the Component Name field. The default name is **in_system_ibert_0**. This must be set to a name that is unique within your project.

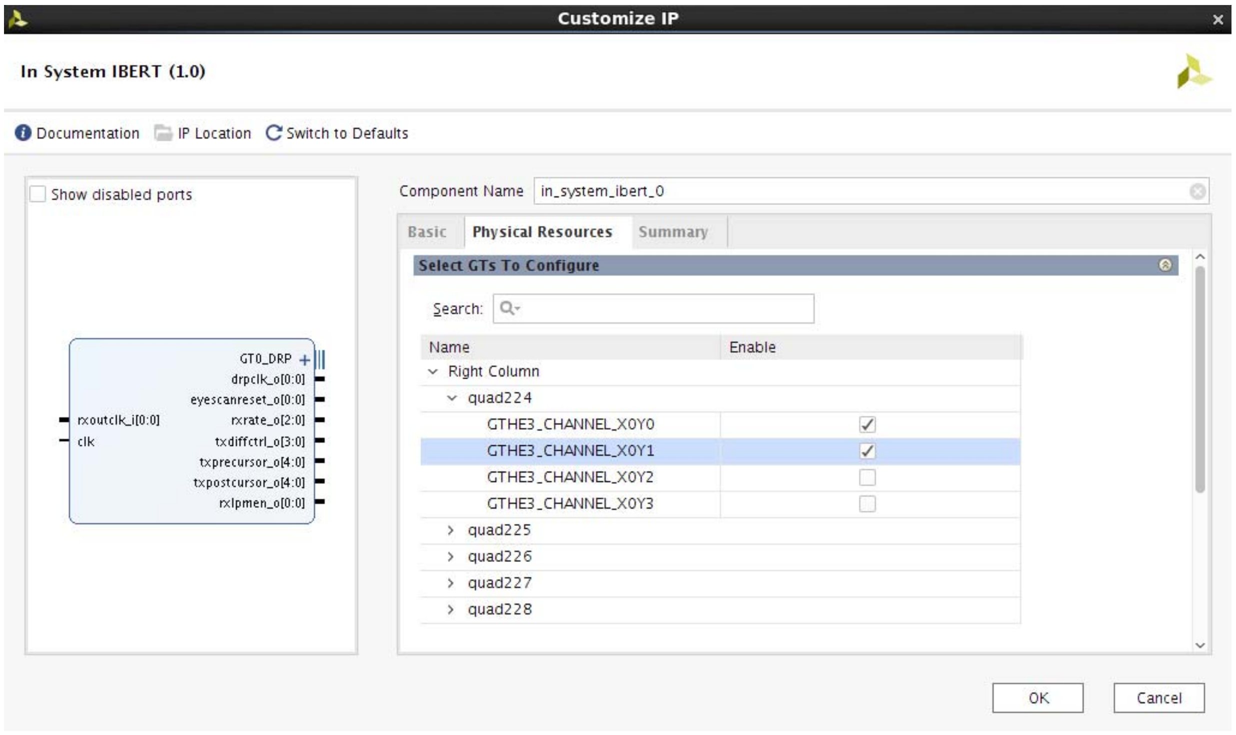
Show Disabled Ports - The IP symbol is shown on the left-hand side of the Customize IP dialog box, and displays only *enabled* ports by default. It organizes input ports on the left-hand side of the symbol and output ports on the right-hand side. The IP symbol is updated as you make customization choices and use the optional ports enablement interface.

GT Type - Select the type of serial transceiver to configure. Available choices are limited to the transceiver types present within the selected device.

Add Optional Input Ports - The optional ports listed in Table 2-2 will be added or enabled if this check box is selected. Default value is false i.e., optional ports are disabled.

Physical Resources Tab

The Physical Resources Tab is shown in Figure 4-2. When customizing options on this tab, it is important to understand that choices you make here affect generated HDL and constraints. Select the options that are appropriate for your project and system. See the UltraScale Architecture GTH Transceivers User Guide ([Ref 9]), for details on transceiver Quad architecture. Select the targeted transceiver by selecting the appropriate check box.



Component Name: in_system_ibert_0

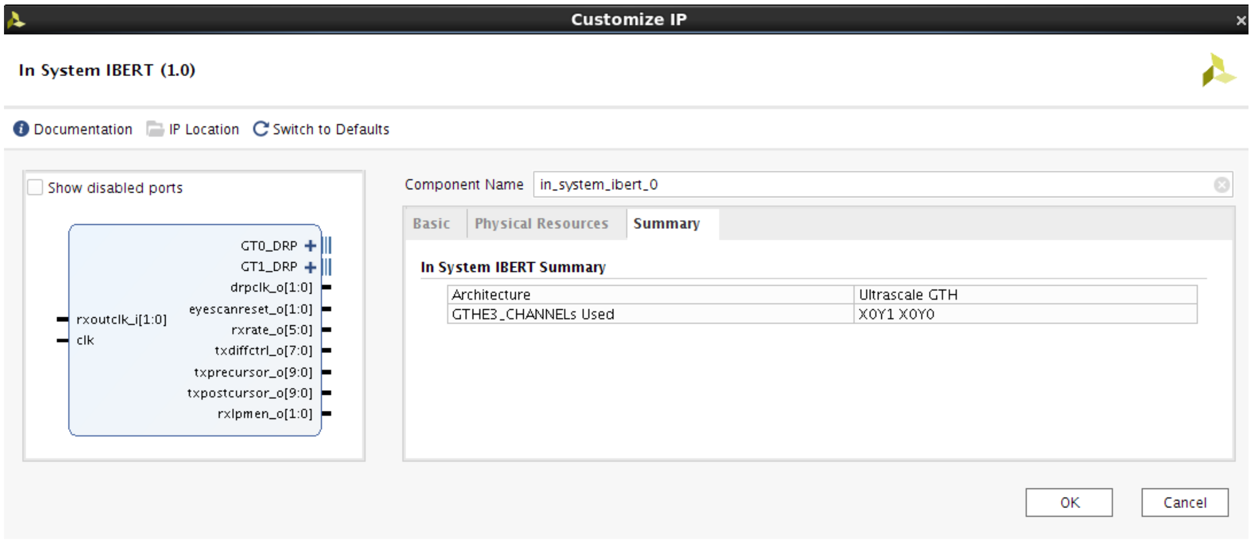
Name	Enable
Right Column	
quad224	
GTHE3_CHANNEL_X0Y0	✓
GTHE3_CHANNEL_X0Y1	✓
GTHE3_CHANNEL_X0Y2	☐
GTHE3_CHANNEL_X0Y3	☐
quad225	
quad226	
quad227	
quad228	

Figure 4-2: **Physical Resources Tab**

Note: The transceiver coordinates and numbers mentioned on this page are derived from device architecture which helps you to select the required transceiver channel. This information is used to group transceivers of same number and show in the Serial Input Output (SIO) analyzer GUI after downloading bit file on hardware. The ISI is not responsible for generating lock constraints for any channel.

Summary Tab

Review the settings chosen in the Summary page (Figure 4-3). If they are satisfactory, click **OK** to generate the ISI core.



Component Name: in_system_ibert_0

In System IBERT Summary

Architecture	Ultrascale GTH
GTHE3_CHANNELs Used	X0Y1 X0Y0

Figure 4-3: **Summary Tab**

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

This ISI core does not have any location or timing constraints.

The clk port of the core is constrained to 100 MHz by default. However, it will be overwritten by the clock constraint of the system clock connected to the clk port as defined in the user design. No other clock constraint is required for the clk port of ISI core.

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 ISI core does not support simulation.

Synthesis and Implementation

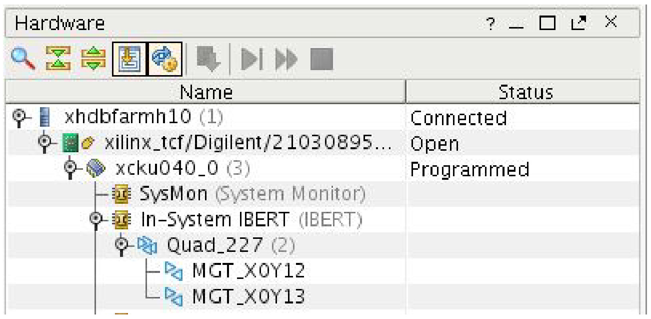
For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 3].

Interacting with Tcl Commands

After the design is loaded into the device, a set of `hw_sio` commands interact with the ISI Core for UltraScale+ Architecture GTH/GTY Transceivers. Refer to *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 5] for more details on Tcl commands.

Viewing Eye scan

1. Start **hw_server** on machine connected to targeted board.
2. From Vivado, connect to **hw_server** and program the bit file.
3. After programming the bit file successfully, you will see the ISI (IBERT) core detected as shown in Figure 4-4.



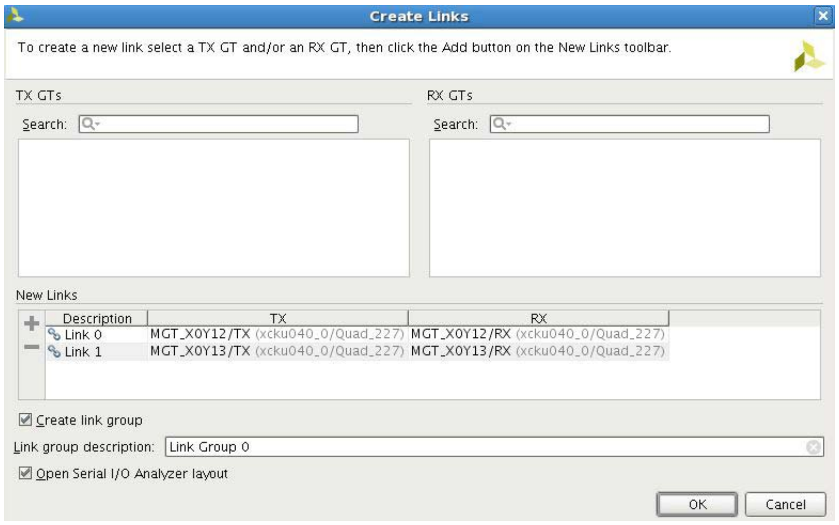
Name	Status
xhdbfarmh10 (1)	Connected
xilinx_tcf/Digilent/21030895...	Open
xcku040_0 (3)	Programmed
SysMon (System Monitor)	
In-System IBERT (IBERT)	
Quad_227 (2)	
MGT_X0Y12	
MGT_X0Y13	

Figure 4-4: **ISI Core detected**

4. Ensure the link is up for the transceiver. If the link is down, the eye scan plot will fail. In gtwizard design, the VIO core is used to show link status which should be high.

5. Create links using the **Create Links** option on the SIO analyzer as shown in Figure 4-5.

   Note: Link Auto-Detection is not supported. ISI does not have the link detection signal or RX/TX patterns required for auto-detecting links, Vivado is therefore unable to auto-detect any links with the ISI RX and TX.



To create a new link select a TX GT and/or an RX GT, then click the Add button on the New Links toolbar.

New Links

Description	TX	RX
Link 0	MGT_X0Y12/TX (xcku040_0/Quad_227)	MGT_X0Y12/RX (xcku040_0/Quad_227)
Link 1	MGT_X0Y13/TX (xcku040_0/Quad_227)	MGT_X0Y13/RX (xcku040_0/Quad_227)

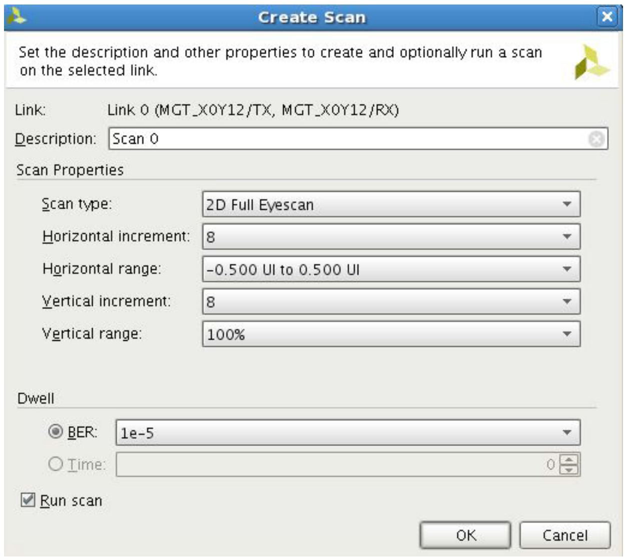
[✓] Create link group

Link group description: Link Group 0

[✓] Open Serial I/O Analyzer layout

Figure 4-5: **Create Links**

6. Right click on link and select **Create Scan**. The **Create Scan** dialog displays as shown in Figure 4-6.



Set the description and other properties to create and optionally run a scan on the selected link.

Link: Link 0 (MGT_X0Y12/TX, MGT_X0Y12/RX)

Description: Scan 0

Scan type: 2D Full Eyescan

Horizontal increment: 8

Horizontal range: -0.500 UI to 0.500 UI

Vertical increment: 8

Vertical range: 100%

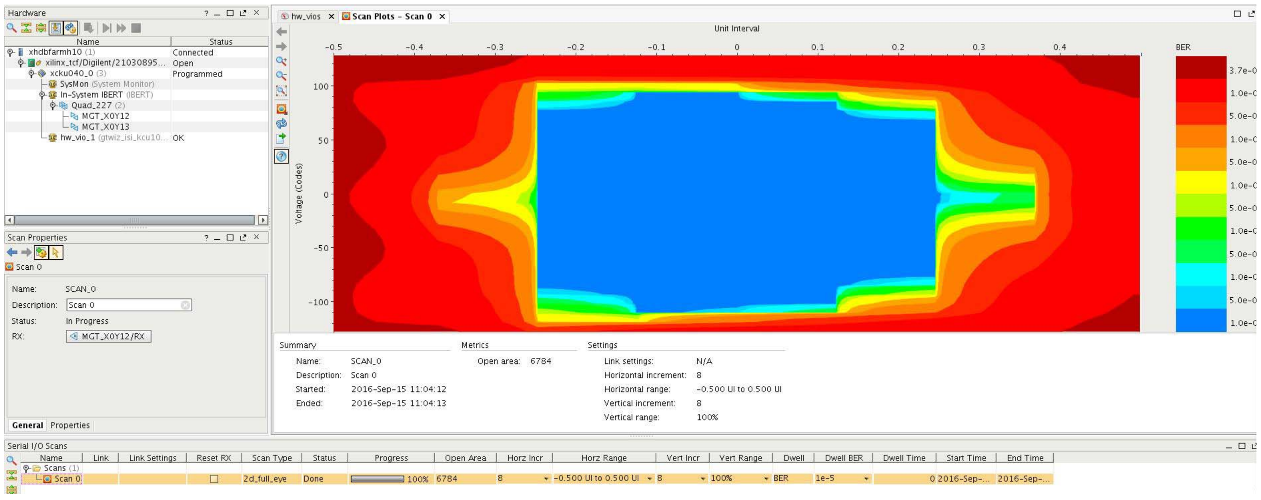
[✓ BER:] 1e-5

Time: 0

[✓] Run scan

Figure 4-6: **Create Scan**

7. Select the required configuration, or use the default configuration and click **OK**. The eye scan plot will display similarly to that shown in Figure 4-7.



Name	Status
xhdbfarmh10 (1)	Connected
xilinx_tcf/Digilent/21030895...	Open
xcku040_0 (3)	Programmed
SysMon (System Monitor)	
In-System IBERT (IBERT)	
Quad_227 (2)	
MGT_X0Y12	
MGT_X0Y13	
hw_vio_1 (gtwiz_isi_kcu10...	OK

Scan 0

Name:	SCAN_0
Description:	Scan 0
Status:	In Progress
RX:	MGT_X0Y12/RX

Summary

Name:	SCAN_0
Description:	Scan 0
Started:	2016-Sep-15 11:04:12
Ended:	2016-Sep-15 11:04:13

Metrics

Open area:	6784

Settings

Link settings:	N/A
Horizontal increment:	8
Horizontal range:	-0.500 UI to 0.500 UI
Vertical increment:	8
Vertical range:	100%

Name	Link	Link Settings	Reset RX	Scan Type	Status	Progress	Open Area	Horz Incr	Horz Range	Vert Incr	Vert Range	Dwell	Dwell BER	Dwell Time	Start Time	End Time
Scans (1)																
Scan 0			☐	2d_full_eye	Done	100%	6784	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-5	0	2016-Sep-...	2016-Sep-...

Figure 4-7: **Eye Scan Plot**

Example Design

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

Purpose of the Example Design

Note: The ISI core does not have its own example design, this is delivered through the GT Wizard example design.

The GT Wizard IP should be customized as per the steps mentioned in Design Flow Steps in *UltraScale FPGAs Transceivers Wizard LogiCORE IP Product Guide* [Ref 1], up to the point *Structural Options tab*. In the Structural Options tab, set the '**Enable In-System IBERT core**' field to **Yes** (Include in Example Design).

After you have customized and generated a core instance, right-click the generated core and select **Open IP Example Design** in the Vivado IDE. A separate Vivado project opens with the GT Wizard 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 GT Wizard with the ISI core instance 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 the GT Wizard core and generated during IP generation.
- Provide a way to perform eye scans on received data and analyze RX margins.
- Provides sweep tests in which you can select the ranges for transceiver TX/RX attributes and port values.

The GT Wizard example design with ISI is 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.

Use Cases

In-System IBERT IP can be easily used with Xilinx GT-based IPs that provide the required ports for connectivity to ISI. The general flow for using ISI is as follows:

1. Configure and generate an instance of In-System IBERT IP (select GT type and target GTs).
2. Configure and generate the IP to be used with ISI with its additional transceiver debug ports enabled.
3. Instantiate the ISI and IP in the design and make necessary port connections.

The above steps are shown in detail for the following use case examples:

- ISI with Single Aurora IP with 2 GTs in RTL flow.
- ISI with Two Aurora IPs, each with 1 GT in IPI flow.

Use Case 1: ISI with Single Aurora IP with 2 GTs in RTL flow

1. Create the AURORA8B10 IP:
    a. Create the **aurora8b10b** IP for two lanes as per PG067 AXI Chip2Chip v4.2 LogiCORE IP Product Guide [Ref 10].

    b. Select/enable the **Additional transceiver control and status ports** checkbox from the **Core Options** tab in the GUI. This will display all transceiver debug ports required for the ISI to connect to.

    Note:
    1. DRP ports are shown at the top for accessibility.
    2. Selecting **Vivado Lab Tools** on the **Core Options** page will not add In-System IBERT to the aurora IP; it will just add ILA & VIO cores.

    c. Generate the output products and open the example design on the generated IP. This will open a separate Vivado window.

    d. In this example design, the Vivado terminal creates the ISI core from the IP Catalog - select same transceiver which was selected in aurora core.

    e. Open the aurora example top level Verilog file to edit and add the ISI core generated in the above step.

    f. Copy the instantiation template of the ISI core generated by the tool and instantiate it in the aurora example top Verilog file. Refer below for instantiation (ISI core without optional input port):

```
in_system_ibert_0 aurora_8b10b_isi_inst (

   .drpclk_o(drpclk_o),

   .gt0_drpen_o(den_in_i),
   .gt0_drpwe_o(dwe_in_i),

   .gt0_drpaddr_o(daddr_in_i),

   .gt0_drpdi_o(di_in_i),

   .gt0_drprdy_i(drdy_out_unused_i),

   .gt0_drpdo_i(drpdo_out_unused_i),

   .gt1_drpaddr_o  (daddr_in_lane1_i),

   .gt1_drpen_o (den_in_lane1_i),

   .gt1_drpdi_o (di_in_lane1_i),

   .gt1_drprdy_i  (drdy_out_lane1_unused_i),

   .gt1_drpdo_i (drpdo_out_lane1_unused_i),

   .gt1_drpwe_o (dwe_in_lane1_i),

   .eyescanreset_o(eyescanreset_o),

   .rxrate_o(rxrate_o),

   .txdiffctrl_o(txdiffctrl_o),

   .txprecursor_o(txprecursor_o),

   .txpostcursor_o(txpostcursor_o),

   .rxlpmen_o(rxlpmen_o),

   .rxoutclk_i(user_clk_i),

   .drpclk_i(init_clk_i),

   .clk(init_clk_i)

);
```

Comment the below ports assignments:

```
//assign  daddr_in_i = 9'h0;

//assign  den_in_i = 1'b0;

//assign  di_in_i = 16'h0;

//assign  dwe_in_i = 1'b0;
```

```
//assign  daddr_in_lane1_i = 9'h0;

//assign  den_in_lane1_i = 1'b0;

//assign  di_in_lane1_i = 16'h0;

//assign  dwe_in_lane1_i = 1'b0;
```

Declare the below wires:

wire [1 : 0] eyescanreset_o;

wire [5 : 0] rxrate_o;

wire [7 : 0] txdiffctrl_o;

wire [9 : 0] txprecursor_o;

wire [9 : 0] txpostcursor_o;

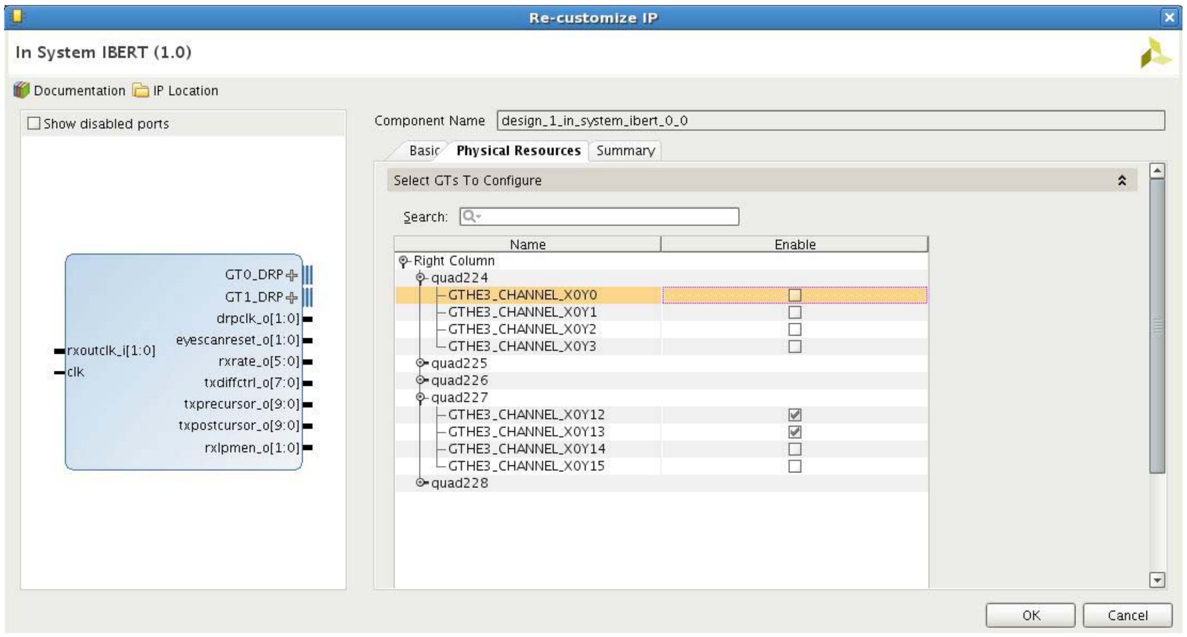
wire [1 : 0] rxlpmen_o;

wire [1 : 0] drpclk_o;

Use Case 2: ISI with Two Aurora IPs, each with 1 GT in IPI flow

Note: These steps are for the KCU105 board and the available transceivers in this device. This section only includes steps for creating the ISI core, two aurora cores and the connection between these cores. It does not cover creating the complete design along with other peripherals such as the axi_traffic_generator, Microblaze, clocking wizard, etc.

1. Create the block design.
2. Add `in_system_ibert` IP to the block design.
3. Select **Add Optional Ports** if required.
4. Re-customize the ISI IP for two channels (for example, channel X0Y12 and X0Y13 under quad227 as shown in Figure 5-1). De-select the first channel that is X0Y0 under quad224.



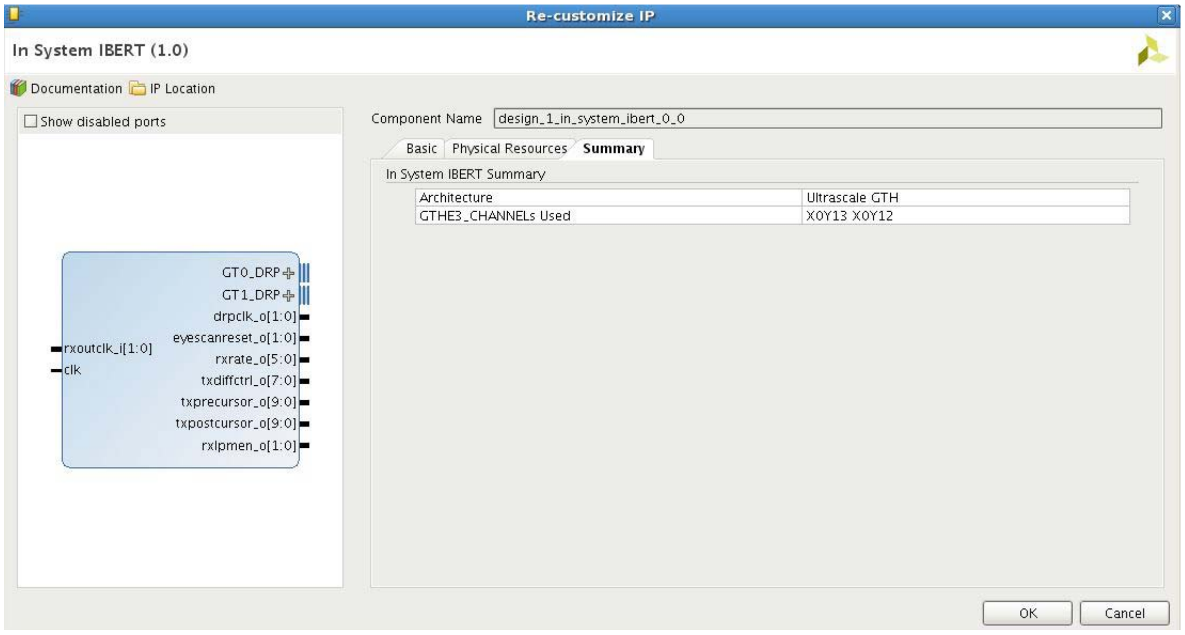
Component Name: design_1_in_system_ibert_0_0

Select GTs To Configure

Name	Enable
Right Column	
quad224	
GTHE3_CHANNEL_X0Y0	☐
GTHE3_CHANNEL_X0Y1	☐
GTHE3_CHANNEL_X0Y2	☐
GTHE3_CHANNEL_X0Y3	☐
quad225	
quad226	
quad227	
GTHE3_CHANNEL_X0Y12	✓
GTHE3_CHANNEL_X0Y13	✓
GTHE3_CHANNEL_X0Y14	☐
GTHE3_CHANNEL_X0Y15	☐
quad228	

Figure 5-1: **Recustomize IP**

5. Click on the **Summary** tab of GUI and check the selected transceiver as shown in Figure 5-2.



Component Name: design_1_in_system_ibert_0_0

In System IBERT Summary

Architecture	Ultrascale GTH
GTHE3_CHANNELs Used	X0Y13 X0Y12

Figure 5-2: **Summary Tab**

6. Click **OK** to generate the IP. This will add the ISI core as shown in Figure 5-3.

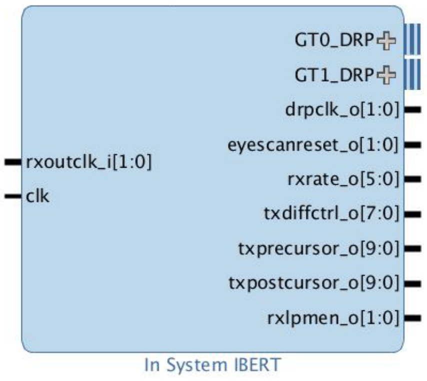
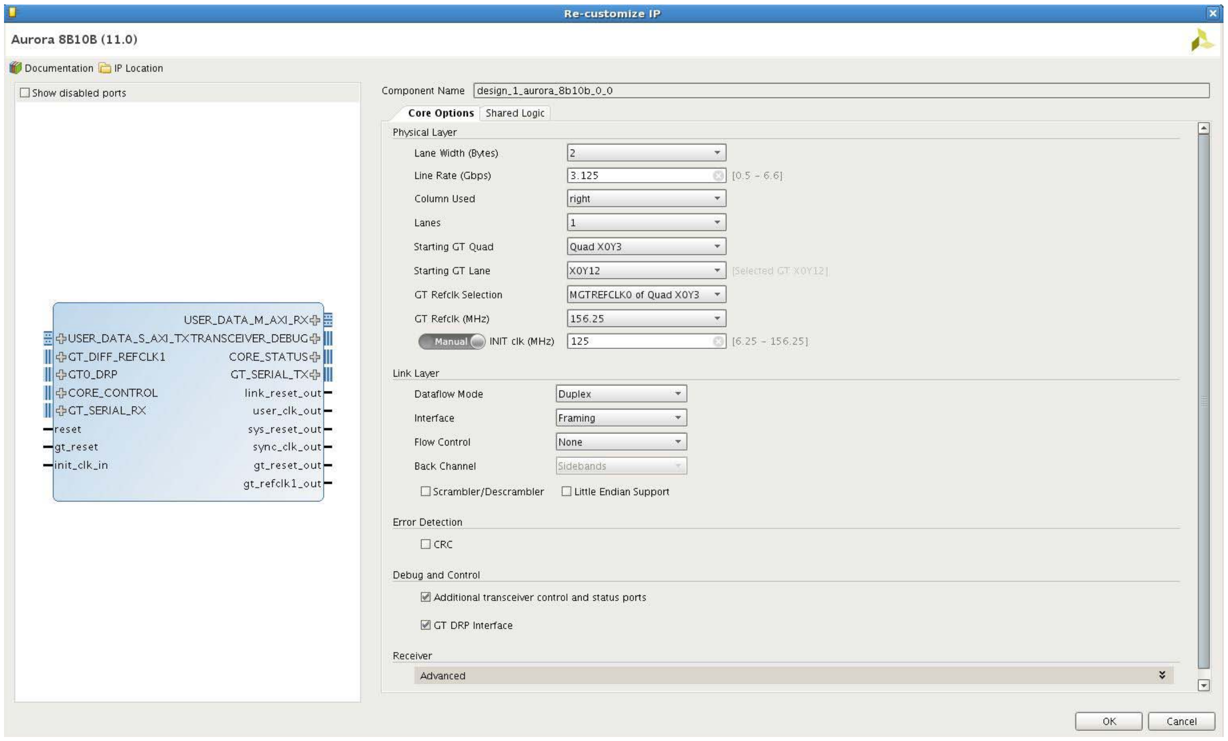


Figure 5-3: **ISI Core**

7. Add the aurora8b10 IP to the block design.
8. Re-customize the aurora core and use the **Core Options** tab to configure it for one channel using the parameters shown in Figure 5-4.



Re-customize IP

Aurora 8B10B (11.0)

Documentation IP Location

☐ Show disabled ports

Component Name design_1_aurora_8b10b_0_0

Core Options | Shared Logic

Physical Layer

Lane Width (Bytes)	2	
Line Rate (Gbps)	3.125	[0.5 - 6.6]
Column Used	right	
Lanes	1	
Starting GT Quad	Quad X0Y3	
Starting GT Lane	X0Y12	[Selected GT X0Y12]
GT Refclk Selection	MGTREFCLK0 of Quad X0Y3	
GT Refclk (MHz)	156.25	
Manual INIT clk (MHz)	125	[6.25 - 156.25]

Link Layer

Dataflow Mode	Duplex
Interface	Framing
Flow Control	None
Back Channel	Sidebands

☐ Scrambler/Descrambler ☐ Little Endian Support

Error Detection

☐ CRC

Debug and Control

☑ Additional transceiver control and status ports

☑ GT DRP Interface

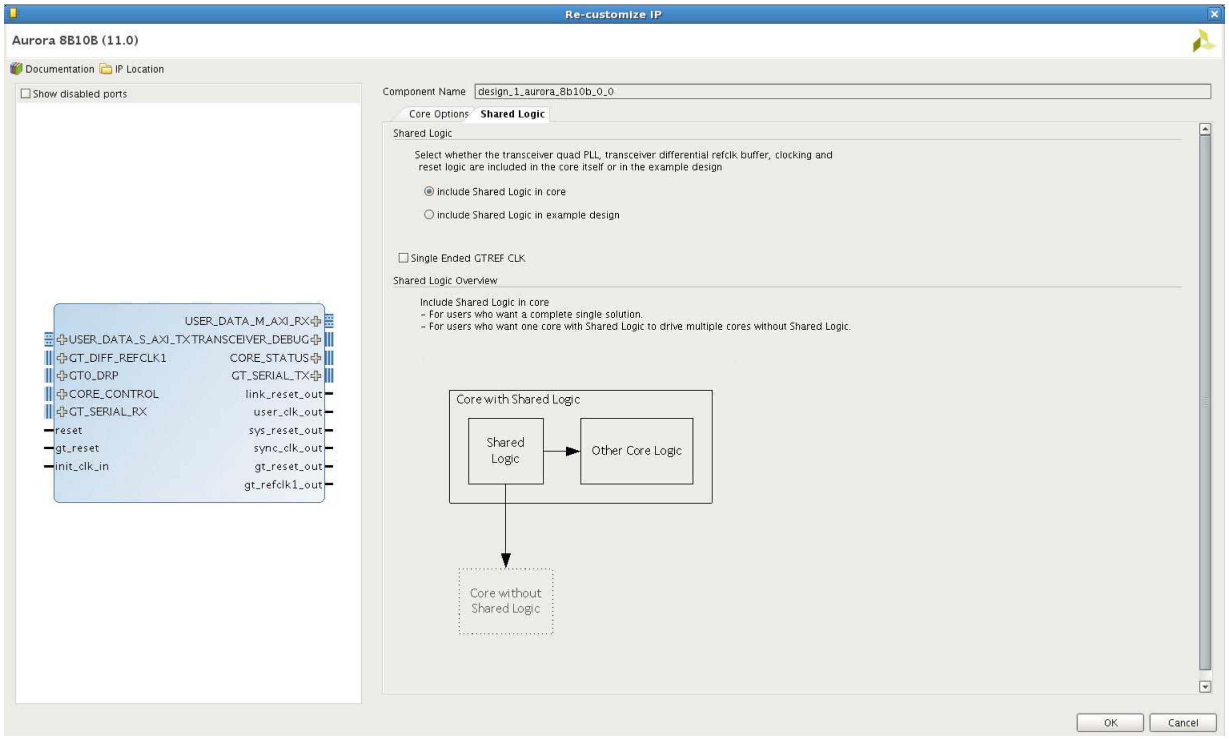
Receiver

Advanced

Figure 5-4: **Core Options**

9. Also in the **Core Options** tab, select the **Additional transceiver control and status ports** and **GT DRP Interface** checkboxes (this will display all the required ports for ISI core at its interface).
10. Click the **Shared Logic** tab (see Figure 5-5).

11. Select **Include Shared Logic in core.** This option is selected because two aurora IPs are being created for two transceivers in the same quad, so this allows the first core to include GT common and provides the clock for the next core.



Component Name: design_1_aurora_8b10b_0_0

Shared Logic

Select whether the transceiver quad PLL, transceiver differential refclk buffer, clocking and reset logic are included in the core itself or in the example design

[✓ include Shared Logic in core] include Shared Logic in example design

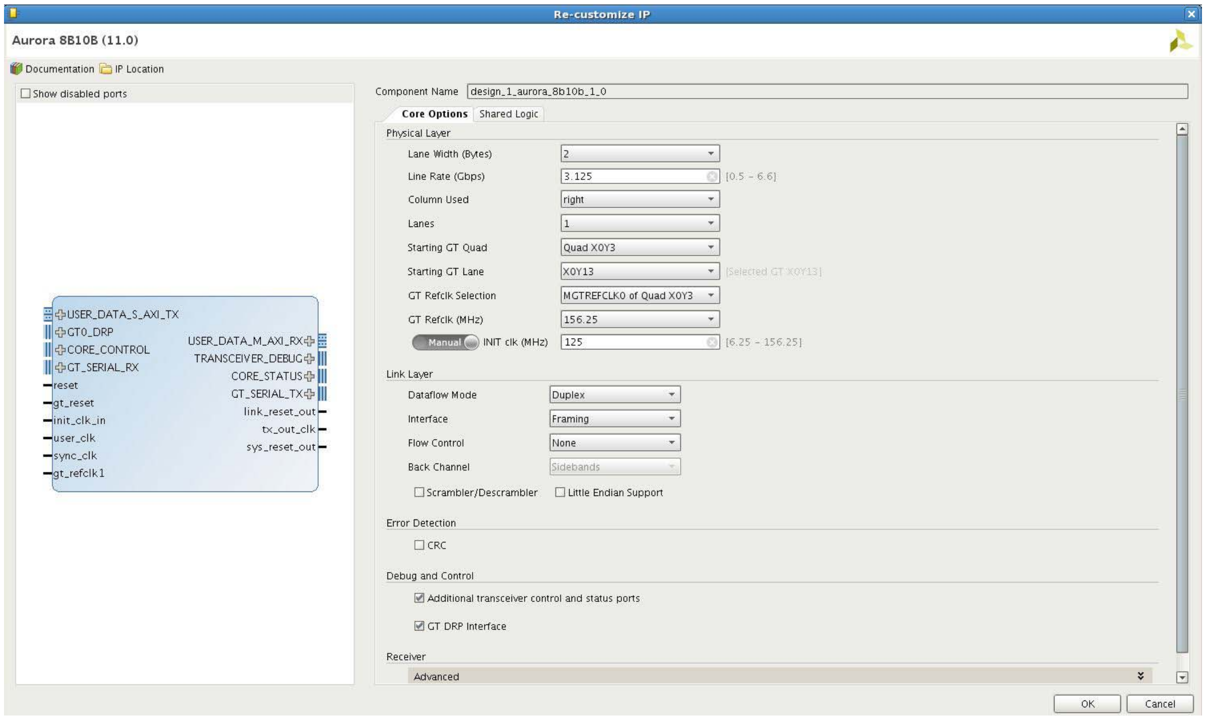
☐ Single Ended GTREF CLK

Shared Logic Overview

Include Shared Logic in core
- For users who want a complete single solution.
- For users who want one core with Shared Logic to drive multiple cores without Shared Logic.

Figure 5-5: **Shared Logic**

12. Click on **OK** to generate the first aurora IP.
13. Add one more aurora IP to the block design, and open for customization.
14. Configure as shown in Figure 5-6.



Re-customize IP

Aurora 8B10B (11.0)

Documentation IP Location

☐ Show disabled ports

Component Name: design_1_aurora_8b10b_1_0

Core Options | Shared Logic

Physical Layer

Lane Width (Bytes)	2	
Line Rate (Gbps)	3.125	[0.5 - 6.6]
Column Used	right	
Lanes	1	
Starting GT Quad	Quad X0Y3	
Starting GT Lane	X0Y13	[Selected GT X0Y13]
GT Refclk Selection	MGTREFCLK0 of Quad X0Y3	
GT Refclk (MHz)	156.25	
Manual INIT clk (MHz)	125	[6.25 - 156.25]

Link Layer

Dataflow Mode	Duplex
Interface	Framing
Flow Control	None
Back Channel	Sidebands

☐ Scrambler/Descrambler ☐ Little Endian Support

Error Detection

☐ CRC

Debug and Control

☑ Additional transceiver control and status ports

☑ GT DRP Interface

Receiver

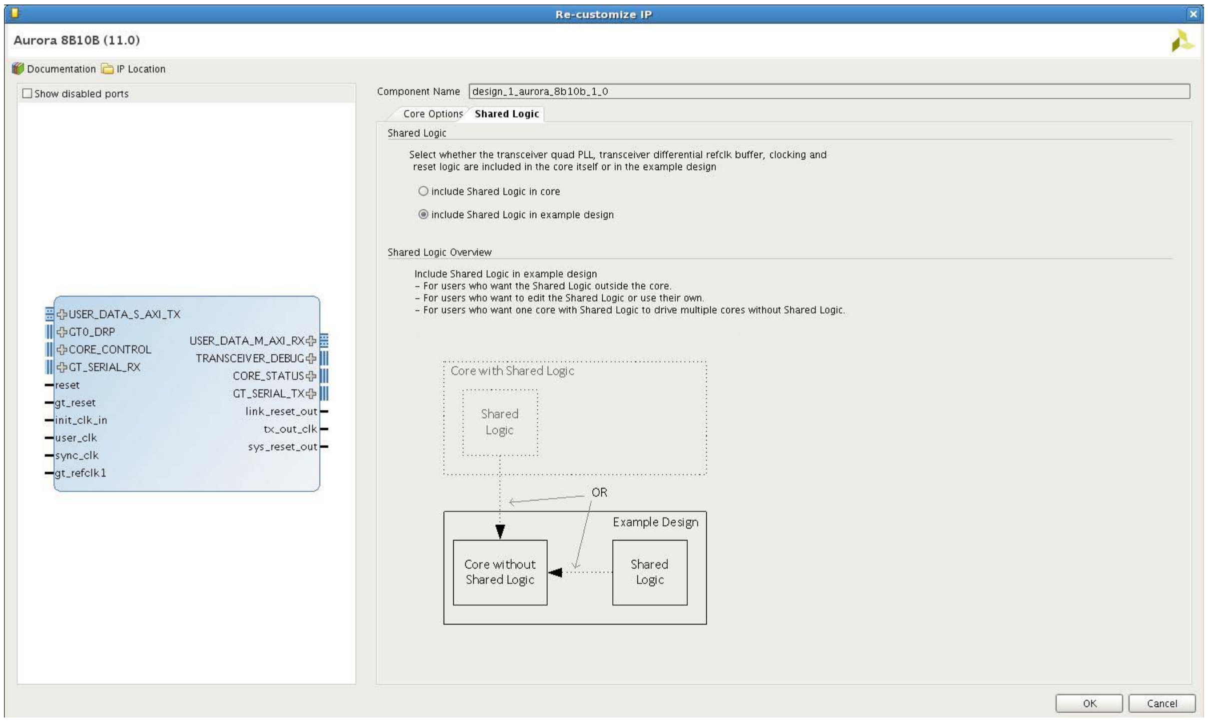
Advanced

OK Cancel

Figure 5-6: **Core Options**

15. Click the **Shared Logic** tab.

16. Select **Include Shared Logic in Example Design** as shown in Figure 5-7.



Re-customize IP

Aurora 8B10B (11.0)

Documentation IP Location

☐ Show disabled ports

Component Name: design_1_aurora_8b10b_1_0

Core Options | Shared Logic

Shared Logic

Select whether the transceiver quad PLL, transceiver differential refclk buffer, clocking and reset logic are included in the core itself or in the example design

○ include Shared Logic in core

◉ include Shared Logic in example design

Shared Logic Overview

Include Shared Logic in example design
- For users who want the Shared Logic outside the core.
- For users who want to edit the Shared Logic or use their own.
- For users who want one core with Shared Logic to drive multiple cores without Shared Logic.

OK Cancel

Figure 5-7: **Shared Logic**

The Block Design has now two aurora cores and one ISI core as shown in Figure 5-8.

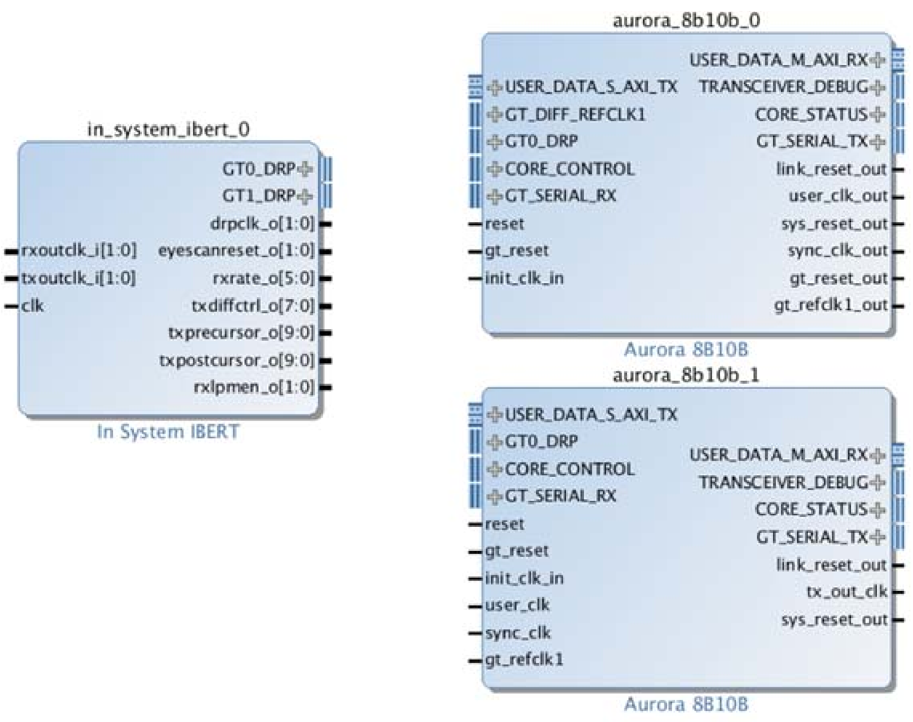
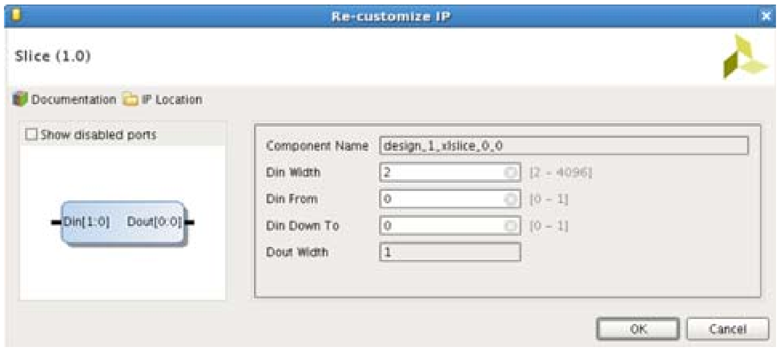


Figure 5-8: **Block Design showing 2xAurora cores and 1xISI core**

17. Slice the ISI output ports to enable connection to two Aurora IP cores.

    For example, `eyescanreset_o` is two bits. This needs to be sliced into two single bit outputs.

    To achieve this, add one slice using the settings shown in Figure 5-9 to create `eyescanreset_o[0:0]`.

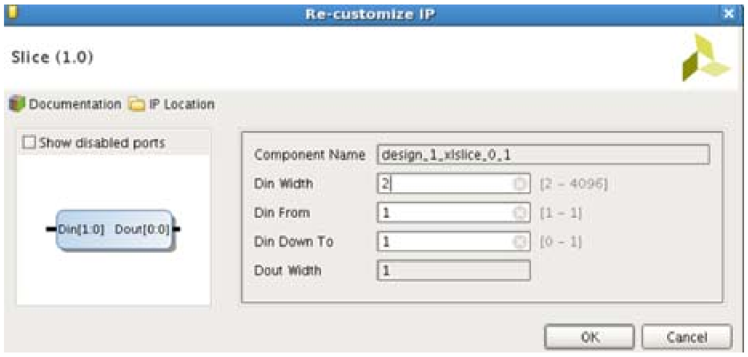


Slice (1.0)

Component Name	design_1_xlslice_0_0	
Din Width	2	[2 - 4096]
Din From	0	[0 - 1]
Din Down To	0	[0 - 1]
Dout Width	1	

Figure 5-9: **Slice 0_0**

Add a second slice using the settings shown in Figure 5-10 to create `eyescanreset_o[1:1]`.



Slice (1.0)

Component Name	design_1_xlslice_0_1	
Din Width	2	[2 - 4096]
Din From	1	[1 - 1]
Din Down To	1	[0 - 1]
Dout Width	1	

Figure 5-10: **Slice 0_1**

Similarly slice for `rxrate_o, txdiffctrl_o, txprecursor_o, txpostcursor_o` and `rxlpmen_o` to slice the ports into two equal parts.

After adding all sliced IPs, the block design should look similar to Figure 5-11.

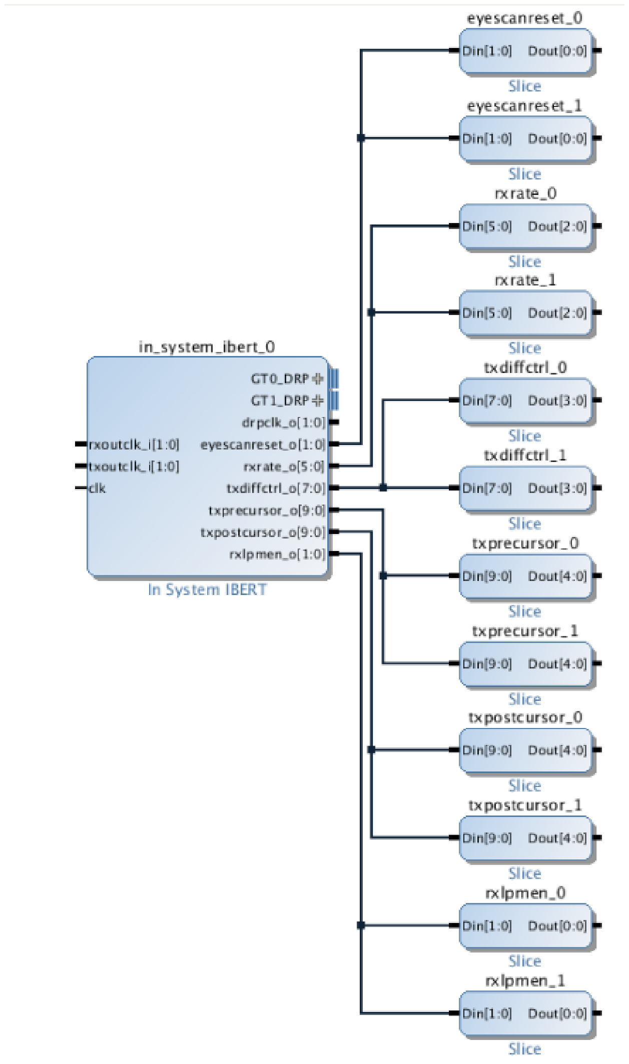


Figure 5-11: **Sliced IP**

18. Connect the ISI and the first aurora core (`aurora_8b10b_0`).

    a. Connect `GT0_DRP` of the ISI core to `GT0_DRP` of the `aurora_8b10b_0` core.

    b. Expand the `TRANSCEIVER_DEBUG` interface of `aurora_8b10b_0`. This interface contains the `eyescanreset`, `rxrate`, `txdiffctrl`, `txprecursor`, `txpostcursor` and `rxlpmen` transceiver ports.

    c. Connect the output of each sliced IP created with the lower bits of the ISI core to the respective ports of the aurora core.

    For example `eyescanreset_o[0:0]` from sliced IP `eyescanreset_0` to `aurora_8b10b_0` should be connected as shown in Figure 5-12.

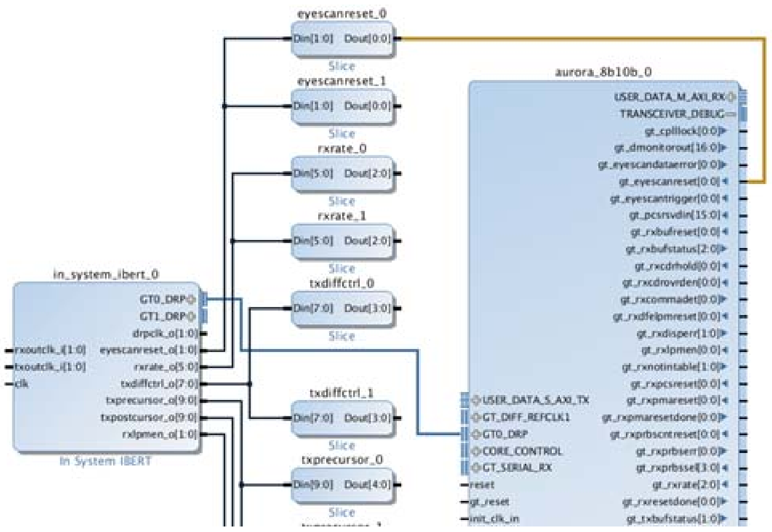


Figure 5-12: **Connected Sliced IP**

Similarly connect `rxrate_0`, `txdiffctrl_0`, `txprecursor_0`, `txpostcursor_0` and `rxlpmen_0` to `aurora_8b10b_0`.

19. Connect the ISI and the second aurora core (`aurora_8b10b_1`).
    a. Connect `GT1_DRP` of ISI core to `GT0_DRP` of `aurora_8b10b_1`.
    b. Follow steps b and c from step 18 to create the other required connections.
20. Create the hierarchy for the sliced IP as shown in Figure 5-13.

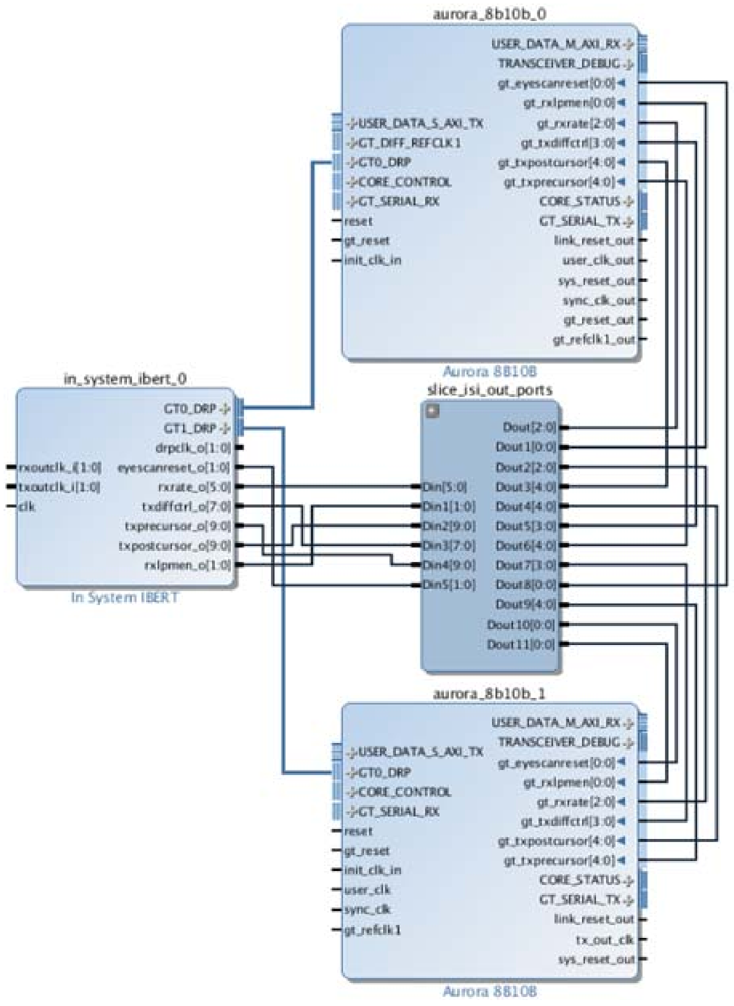


Figure 5-13: **Sliced IP Hierarchy**

21. Concatenate `usr_clk_out` of `aurora_8b10b_0` and connect it to the `rxoutclk_i` input port of the ISI core as shown in Figure 5-14.

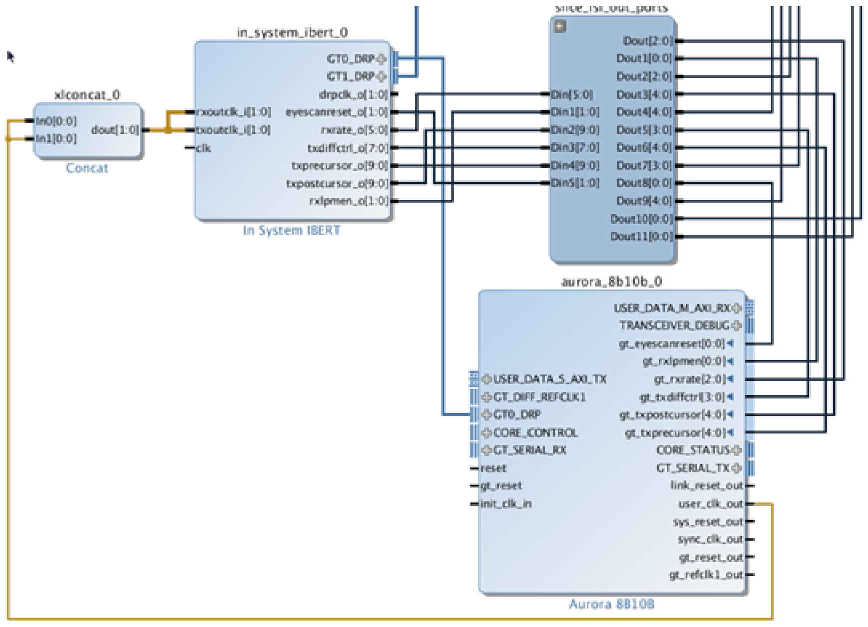


Figure 5-14: **Concatenate usr_clk_out**

22. Connect the two aurora cores' shared logic as detailed in Table 5-1.

Table 5-1: **Shared logic connections**

aurora_8b10b_0		aurora_8b10b_1	
Port Name	**Direction**	**Port Name**	**Direction**
user_clk_out	Out	user_clk	In
sync_clk_out	Out	sync_clk	In
sys_reset_out	Out	reset	In
gt_reset_out	Out	gt_reset_in	In
gt_refclk1_out	Out	refclk1_in	In

23. Make `init_clk_in` of `aurora_8b10b_0` external. Change the frequency of this external pin to 125 and connect it to `init_clk_in` of `aurora_8b10b_1` and clk or ISI.
24. Make the `GT_DIFF_REFCLK1` port of `aurora_8b10b_0` external.

This creates a system for a single ISI with two aurora IPs.

Add a traffic generator to drive `USER_DATA_S_AXI_TX` and `RX`. Add other components as required.

Additional Information

This section addresses some typical questions about In-System IBERT IP.

1. **"How many instances of In-System IBERT are required for a multi-lane protocol?"**

   There is only one instance of In-System IBERT required per design. In-System IBERT can work with all GTs used in the design.

2. **"How many instances of In-System IBERT are required if there are multiple IPs in the design?"**

   One or more depending on how many instances the user prefers.

3. **"Is there any logic in rx/tx domain of GTs driven by In-System IBERT?"**

   In-System IBERT has no logic in these domains, and includes synchronizers from the DRP clock domain to the TX/RX domains as necessary.

4. **"Is it required to generate separate cores according to the GT type (GTH/GTY)?"**

   Yes it is required to generate separate In-System IBERT cores according to the GT type.

5. **"What happens if the clock connected to the 'clk or drpclk' ports of the ISI is stopped or not available for some time?"**

   Creating an In-System IBERT design with an internal system clock prevents a scan being performed. When creating an eye scan, the status changes from **In Progress** to **Incomplete**. Eye scan is incomplete when the internal system clock (MGTREFCLK) is connected to clk/drpclk_i input port of In-System IBERT IP.

   You can consider using an external clock which does not exhibit this behavior. Alternatively, click on any available channel or lane available in the SIO GUI. Go to the properties window and find the MB_RESET reg under the LOGIC field. Set it to 1 and then toggle back to 0. Re-run eye scan or sweep.

Upgrading

This appendix is not applicable for the first release of the core.

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 ISI for UltraScale™/UltraScale+™ GTH/GTY Transceivers core.

Finding Help on Xilinx.com

To help in the design and debug process when using the ISI for UltraScale/UltraScale+ GTH/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 ISI for UltraScale/UltraScale+ GTH/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 Records for the In-System IBERT for UltraScale/UltraScale+ GTH/GTY Transceivers Core:

AR: 67867

AR: 69206

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

If, when performing eye scans, you see the following behaviors:

- **Open area is zero** message displays.
- Eye scan progress status is **Incomplete**.
- Eye scan progress freezes.

Perform these tests:

1. Make sure the RX Link status is high and traffic is live on the RX side of transceiver.
2. Toggle LOGIC.`EYESCANRESET` of the failing lane/channel.
3. Re-run eye scan after performing the above steps.

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.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (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 the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

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

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 GTH Transceivers User Guide (UG576)
10. AXI Chip2Chip LogiCORE IP Product Guide (PG067)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/04/2017	1.0	Updated core constraints description.
06/07/2017	1.0	AR Updated.
10/05/2016	1.0	Initial Xilinx release.

Please Read: Important Legal Notices