KCU105 10GBASE-R Ethernet TRD User Guide

KUCon-TRD04

Vivado Design Suite

UG921 (v2015.4) November 25, 2015

XILINX
ALL PROGRAMMABLE™

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
11/25/2015	2015.4	Updated all references to Vivado Design Suite version 2015.3 to 2015.4. Revised the resource numbers in Table 1-1.
10/07/2015	2015.3	Updated all references to Vivado Design Suite version 2015.2 to 2015.3. Revised the resource numbers in Table 1-1.
06/30/2015	2015.2	Updated all references to Vivado Design Suite version 2015.1 to 2015.2.
04/27/2015	2015.1	Updated all references to Vivado Design Suite version 2014.4.1 to 2015.1.
02/25/2015	2014.4.1	Initial Xilinx Release.

Table of Contents

Appendix E: Additional Resources and Legal Notices

Introduction

This document describes the features and functions, and how to setup, operate, test, and modify the 10GBASE-R Ethernet targeted reference design (10GBASE-R TRD).

10GBASE-R TRD Overview

The 10GBASE-R TRD targets the Kintex® UltraScale™ XCKU040-2FFVA1156E FPGA running on the KCU105 evaluation board. It demonstrates 10-Gigabit Ethernet connectivity using 10-Gigabit Ethernet PCS/PMA IP (10GBASE-R) and 10-Gigabit Ethernet MAC IP (10G MAC) cores on two channels. The data source for these channels can be configured to be either from an internal or external Traffic Generator. Figure 1-1 shows the 10GBASE-R TRD block diagram.

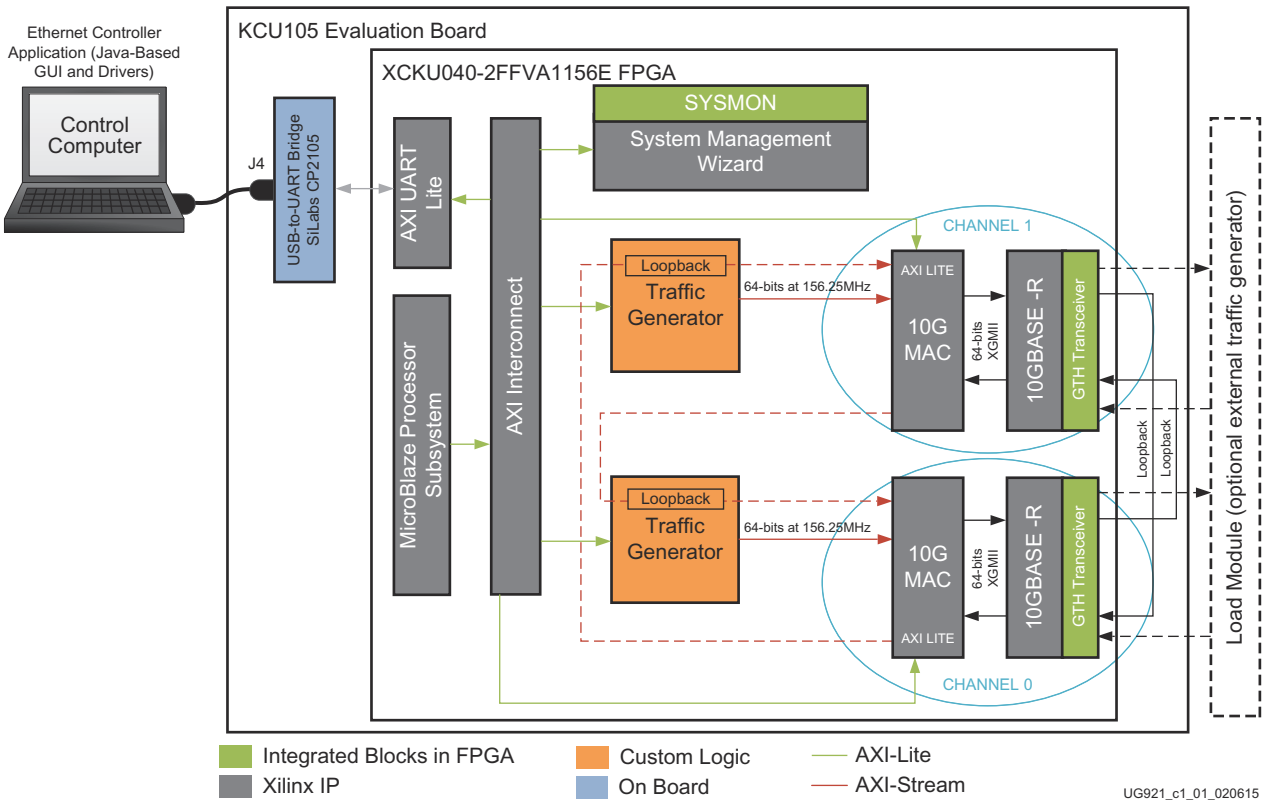


Figure 1-1: **The 10GBASE-R TRD Environment**

The design has two 10 Gb/s Ethernet channels; channel 0 and channel 1. Data from one channel is looped to the other as shown in Figure 1-1. Internal generator mode and external generator mode are supported:

- **Internal generator mode**. In this mode, the Traffic Generator block generates the transmit data. The data is looped back on the PHY side using fiber optic cables. The looped-back data becomes the receive data on the other channel and is verified by the 10-Gigabit Ethernet MAC core.
- **External generator mode**. In this mode, an external generator generates the data. Appendix D describes using an external Ixia NGY-NP4-01 10-Gigabit application network processor load module (Ixia load module). The data received by the 10-Gigabit Ethernet MAC IP core is looped back in the Traffic Generator block and passed as transmit data to the other 10-Gigabit Ethernet MAC IP core. The Ixia load module verifies this data when received.

A MicroBlaze™ processor-based subsystem monitors the 10-Gigabit Ethernet MAC IP core statistics and passes the information to the Ethernet Controller application (GUI) running on the control computer using the USB-to-UART port on the KCU105 board. This subsystem also controls the Traffic Generator and monitors Ethernet performance.

Components, Features, and Functions

The 10GBASE-R TRD includes:

- 10-Gigabit Ethernet PCS/PMA IP core:
  - Uses a GTH transceiver (GTHE3) running at 10.3125 Gb/s line rate.
  - Provides a single data rate (SDR) 10-Gigabit Ethernet Media Independent Interface (XGMII) which connects to the 10-Gigabit Ethernet MAC IP core. The XGMII interface runs at 156.25 MHz and the data path is 64-bits wide.
  - Serial interface to fiber-optic connections.
- 10-Gigabit Ethernet MAC IP core:
  - Connects to the 10-Gigabit Ethernet PCS/PMA IP core using the XGMII interface
  - Provides AXI4-Stream protocol support on the user interface running at 156.25 MHz
  - Is monitored through an AXI4-Lite interface
- Traffic Generator and Monitor:
  - Generates Ethernet traffic or loops back Ethernet traffic selected by user input
  - Monitors bandwidth utilization on the transmit and receive AXI4-Stream interfaces of the 10-Gigabit Ethernet MAC IP core
  - Is configured and monitored through an AXI4-Lite interface

- System Management Wizard:
  - Uses the UltraScale™ architecture System Monitor (SYSMON) block to monitor FPGA temperature, voltages and currents
  - Is configured and monitored through an AXI4-Lite interface
- AXI UART Lite:
  - Provides the controller interface for asynchronous serial data transfer. This interface connects to the USB-to-UART port on the KCU105 board, and is used to communicate with the control computer.
  - Provides an AXI4-Lite interface on the other end to communicate with the MicroBlaze processor subsystem.
- MicroBlaze processor subsystem and AXI Interconnect:
  - Communicates with the 10-Gigabit Ethernet MAC IP core, Traffic Generator and Monitor, System Management Wizard, and AXI UART Lite using the AXI4-Lite protocol.
  - Drivers running on the MicroBlaze processor subsystem interpret commands received from the Ethernet Controller application running on the control computer and convert them to AXI4-Lite transactions.
- Ethernet Controller application:
  - Provides a graphical user interface running on the control computer to pass user inputs to the 10GBASE-R TRD and to display status through the KCU105 board USB-to-UART port.

Resource Utilization

Table 1-1 lists the resources used by the 10GBASE-R TRD after synthesis has run. Place and route can alter these numbers based on placements and routing paths, so use these numbers as a rough estimate of resource utilization. These numbers might vary based on the version of the 10GBASE-R TRD and the tools used to regenerate the design.

Table 1-1: **10GBASE-R TRD Resource Utilization**

Resource Type	Used	Available	Usage (%)
CLB LUTs	14,411	242,400	5.95
CLB Registers	16,818	484,800	3.47
Block RAM Tiles	35	600	5.83
Global Clock Buffers	9	240	1.88
BUFG_GT_SYNC	4	55	7.27
BUFG_GT	7	120	5.83
GTHE3_CHANNEL	2	20	10.00

Table 1-1: **10GBASE-R TRD Resource Utilization *(Cont'd)***

Resource Type	Used	Available	Usage (%)
GTHE3_COMMON	1	5	20.00
SYSMONE1	1	1	100.00

Setup

This chapter lists the requirements and describes how to do all preliminary setup of the KCU105 board, control computer, and software before bringing up the 10GBASE-R TRD.

IMPORTANT: *Perform the procedures described in this chapter before performing the bring up procedures described in Chapter 3, Bringing Up the Design.*

Requirements

Hardware

- KCU105 board with the Kintex® UltraScale™ XCKU040-2FFVA1156E FPGA
- Two USB cables, standard-A plug to micro-B plug
- Power Supply: 100 VAC–240 VAC input, 12 VDC 5.0A output
- Two SFP+ 10GBASE-SR/SW transceiver modules (Avago Technologies)
- One LC-LC duplex 10 Gb multimode 50/125 OM3 fiber optic patch cable, 2 x LC Male to 2 x LC Male, part number FO-10GGBLCX20-001 (Amphenol Corporation)

Computer

One computer is required, and is identified as the control computer throughout this document. It is required for running the Vivado Design Suite, configuring the FPGA, and running the Ethernet Controller application to control and monitor the reference design. It can be a laptop or desktop computer with Microsoft Windows 7 operating system.

Design Tools and Software

- Vivado® Design Suite 2015.4
- USB UART drivers (Silicon Laboratories CP210x VCP drivers)
- Java version 1.7
- Ethernet Controller application (GUI included with the 10GBASE-R TRD)

Download and installation instructions for each required software is described in Preliminary Setup.

Preliminary Setup

Complete these tasks before bringing up the design described in Chapter 3, Bringing Up the Design.

Install Vivado Design Suite

Install Vivado Design Suite 2015.4 on the control computer. Follow the installation instructions provided in *Vivado Design Suite User Guide Release Notes, Installation, and Licensing* (UG973) [Ref 1].

Download Targeted Reference Design Files

1. Download the 10GBASE-R TRD ZIP file Here:
   KCU105 Evaluation Kit Documentation
2. Unzip the contents of the file to a working directory.
3. The unzipped contents will be located at `<working_dir>/kcu105_10gbaser_trd`.

The 10GBASE-R TRD directory structure is described in Appendix A, Directory Structure.

Install the USB UART Drivers

Download the *CP210x USB to UART Bridge VCP drivers* (for Windows 7) from Silicon Labs. Follow the instructions in *Silicon Labs CP210x USB-to-UART Installation Guide* (UG1033) [Ref 2].

Configure the Control Computer COM Port

The 10GBASE-R TRD uses the Ethernet Controller application to communicate between the control computer and the KCU105 board. To configure the control computer COM ports for this purpose:

1. Connect the KCU105 board to the control computer and power supply as shown in Figure 2-1.

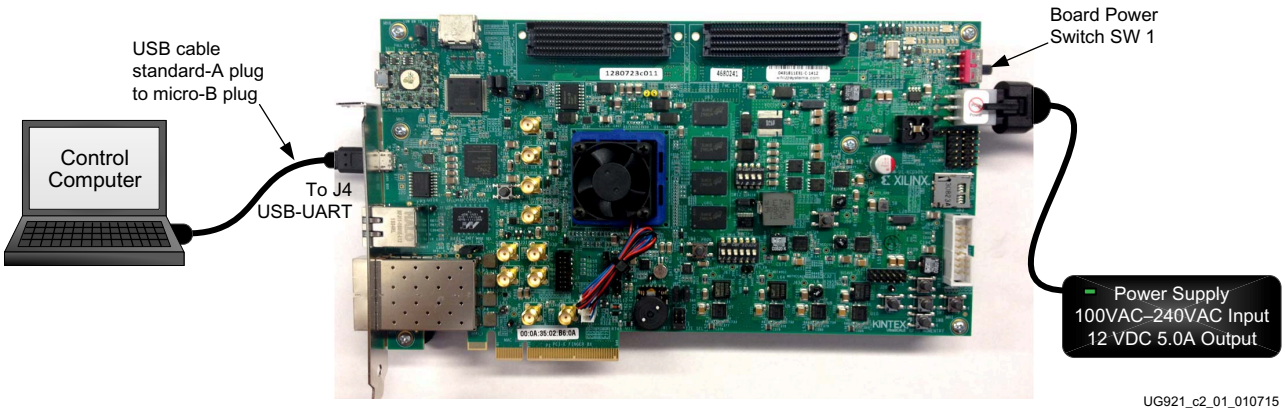


Figure 2-1: **Connection Diagram for Preliminary Setup**

2. Power on the KCU105 board by placing switch SW1 to the ON position (SW1 in Figure 2-1).
3. Open the control computer Device Manager (Figure 2-2). In the Windows task bar, Click **Start**, click **Control Panel**, and then click **Device Manager**.

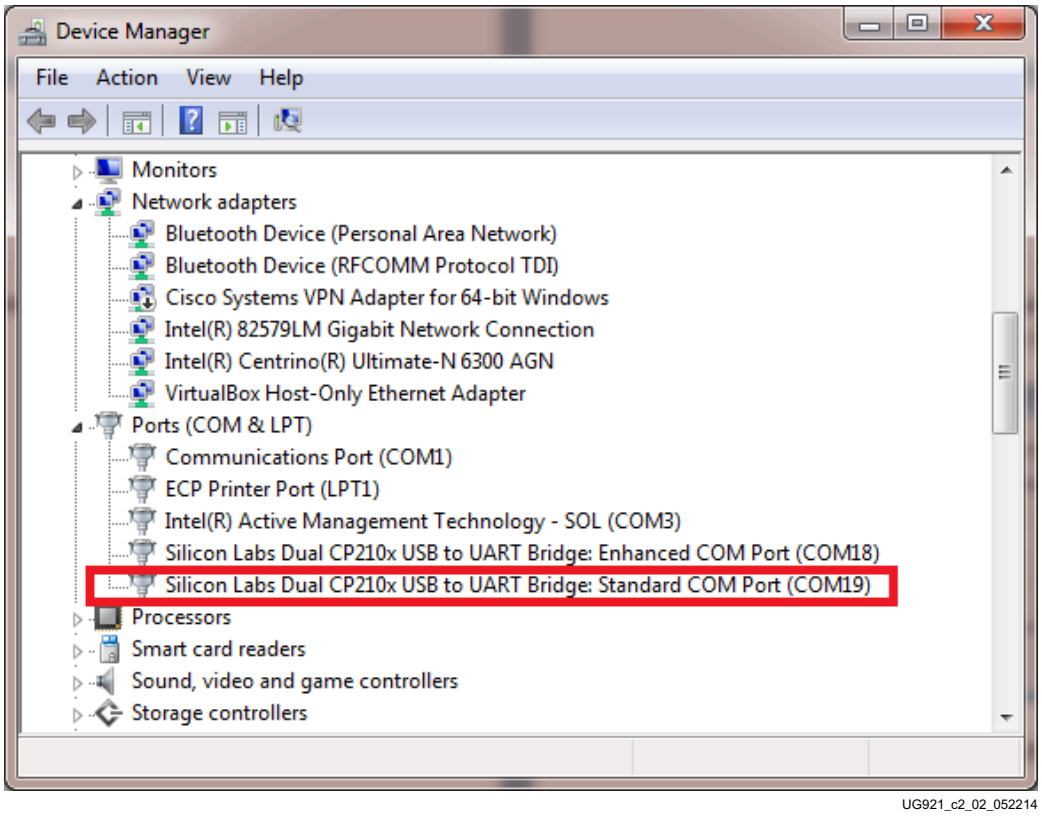


Figure 2-2: **Device Manager**

4. In the Device Manager window (Figure 2-2), expand **Ports (COM & LPT)**, right-click **Silicon Labs CP210x USB to UART Bridge: Standard COM Port**, and then click **Properties**.
5. In the properties window (Figure 2-3), select the **Port Settings** tab.
6. Set **Bits per second**, **Data bits**, **Parity**, **Stop bits**, and **Flow control** to the values shown in Figure 2-3, and click **OK**.

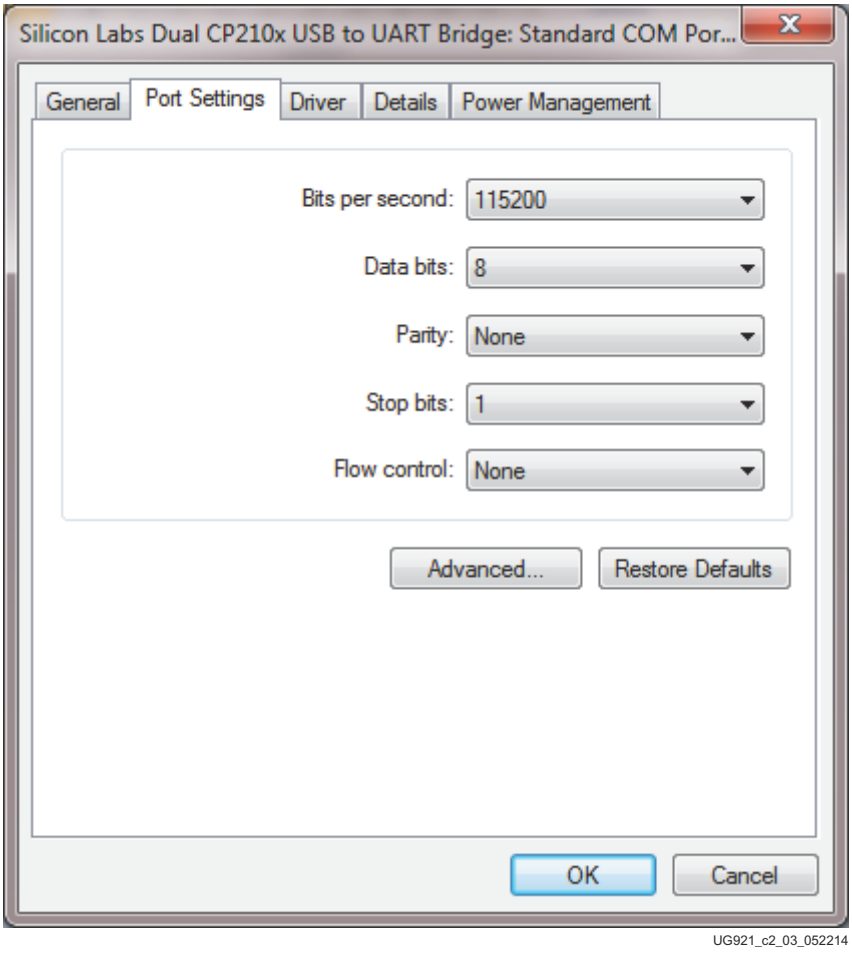


Figure 2-3: **Port Settings**

Install Java 1.7

Installation of Java 1.7 is required for the Ethernet Controller Application.

Download Java SE Runtime Environment 7 from Oracle and install it on the control computer. Follow the installation instructions provided with the software.

Install the Ethernet Controller Application

The Ethernet Controller application is a Java-based application, so confirm Java has been installed as described in Install Java 1.7, page 12.

1. Browse to `<working_dir>/kcu105_10gbaser_trd/software/GUI` (Figure 2-4).

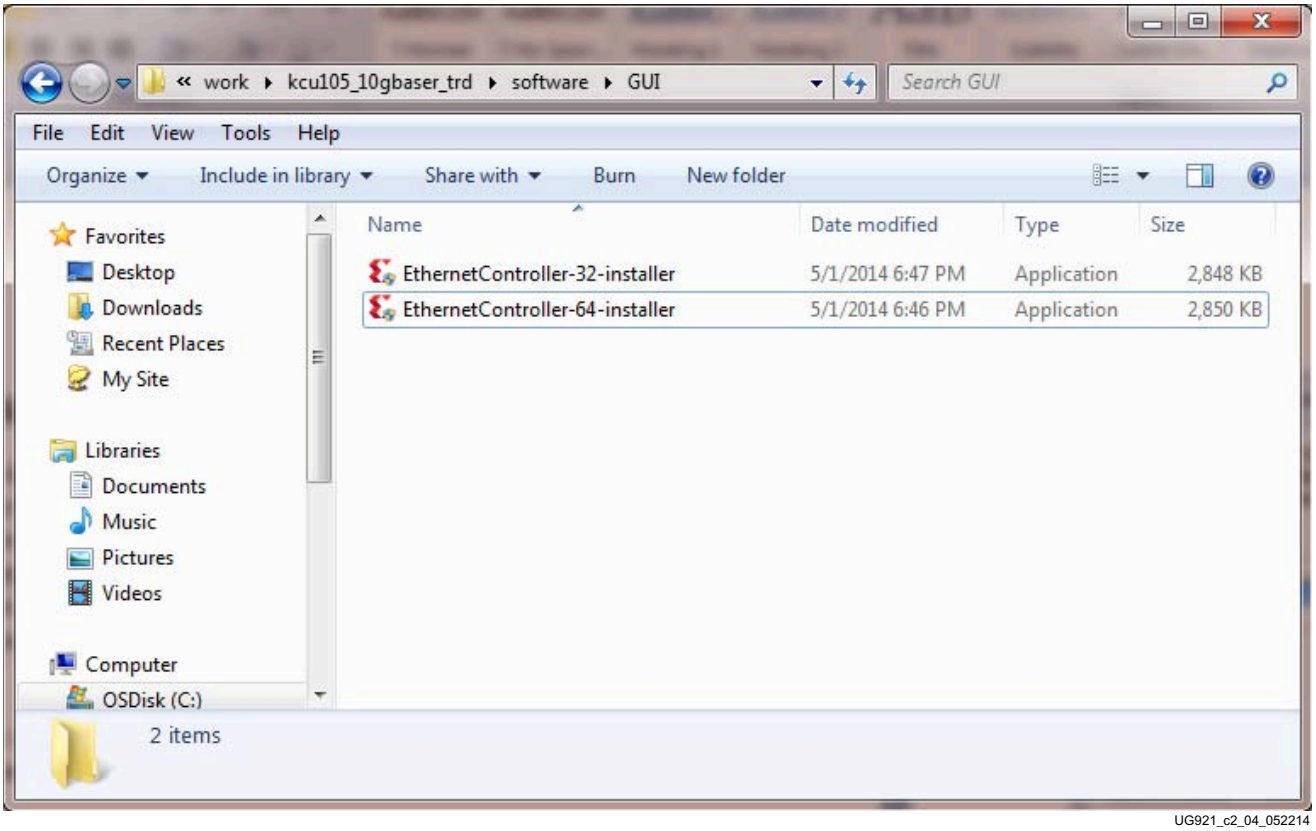


Figure 2-4: **Directory Location, Ethernet Controller Installer**

2. Right-click either the **EthernetController-32-installer** (for a 32-bit operating system) or **EthernetController-64-installer** (for a 64-bit operating system) and click **Run as administrator** (Figure 2-4).

3. Click **Yes** in the dialog box that opens.
4. In the License Agreement display (Figure 2-5), click **I Agree** to continue installation.

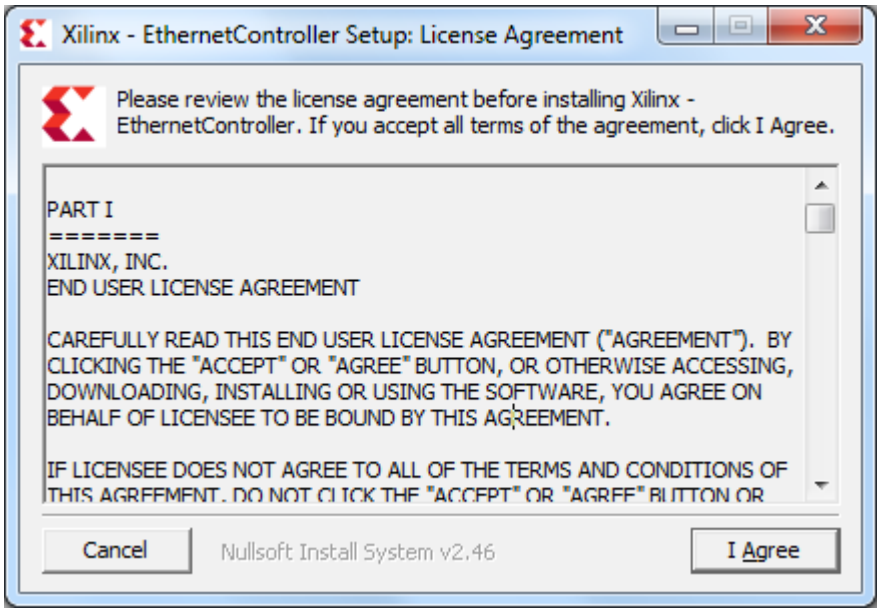


Figure 2-5: **License Agreement**

5. Browse to the location where the Ethernet Controller application will be installed and click **Install** (Figure 2-6).

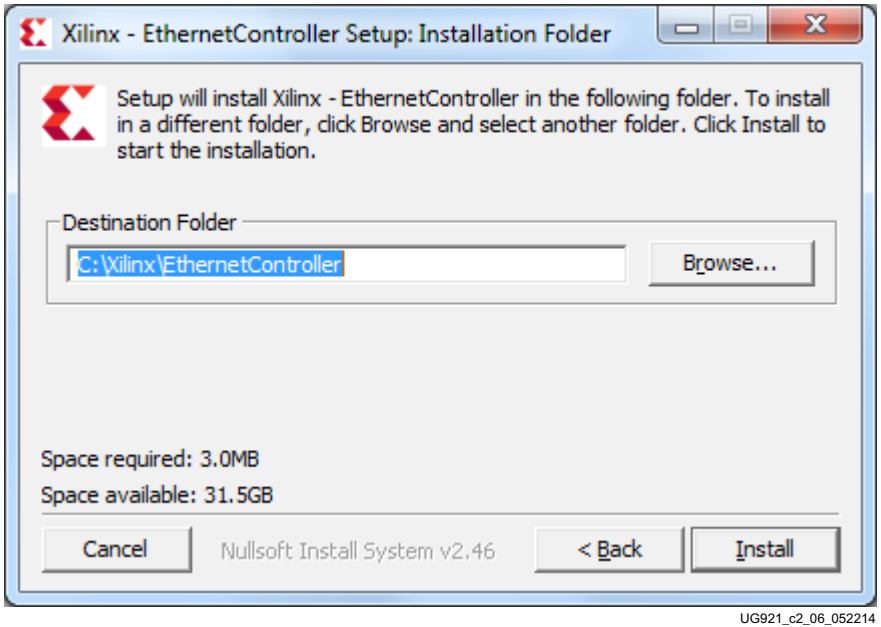


Figure 2-6: **Ethernet Controller Installation Location**

6. Click **Close** after installation is complete (Figure 2-7).

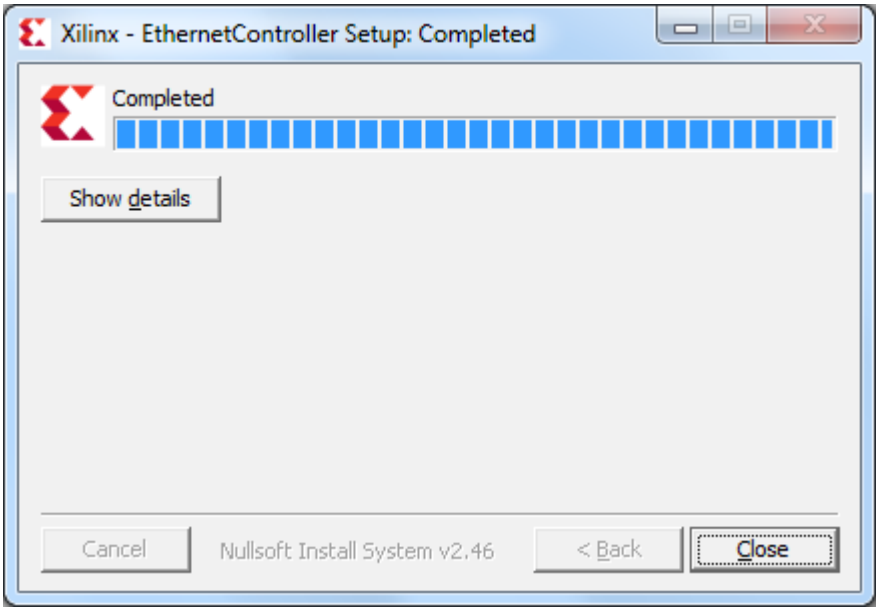


Figure 2-7: **Installation Complete**

TIP: *To uninstall the Ethernet Controller application after design bring up, open the Control Panel. In the Control Panel Click* ***All Control Panel Items>Programs and Features*** *and uninstall program "****Xilinx Ethernet Controller - Powered by Xilinx****."*

Ready to Bring Up the Design

After all procedures in this chapter are complete, go to Chapter 3, Bringing Up the Design.

Bringing Up the Design

This chapter describes how to bring up the 10GBASE-R TRD by programming the FPGA with the bitstream and running the 10GBASE-R TRD under the control of the Ethernet Controller application. The procedure for running the internal Traffic Generators is provided at the end of this chapter.

IMPORTANT: *Perform the preliminary setup procedures described in Chapter 2, Setup before performing the bring up procedures described in this chapter.*

Program the Board

1. Connect the KCU105 board to the control computer and power supply as shown in Figure 3-1.
2. Insert the SFP+ modules into the SFP cage on the KCU105 board and the connect the fiber optic cables (also shown in Figure 3-1).
3. Set the KCU105 board switches and jumpers as shown in Figure 3-1.

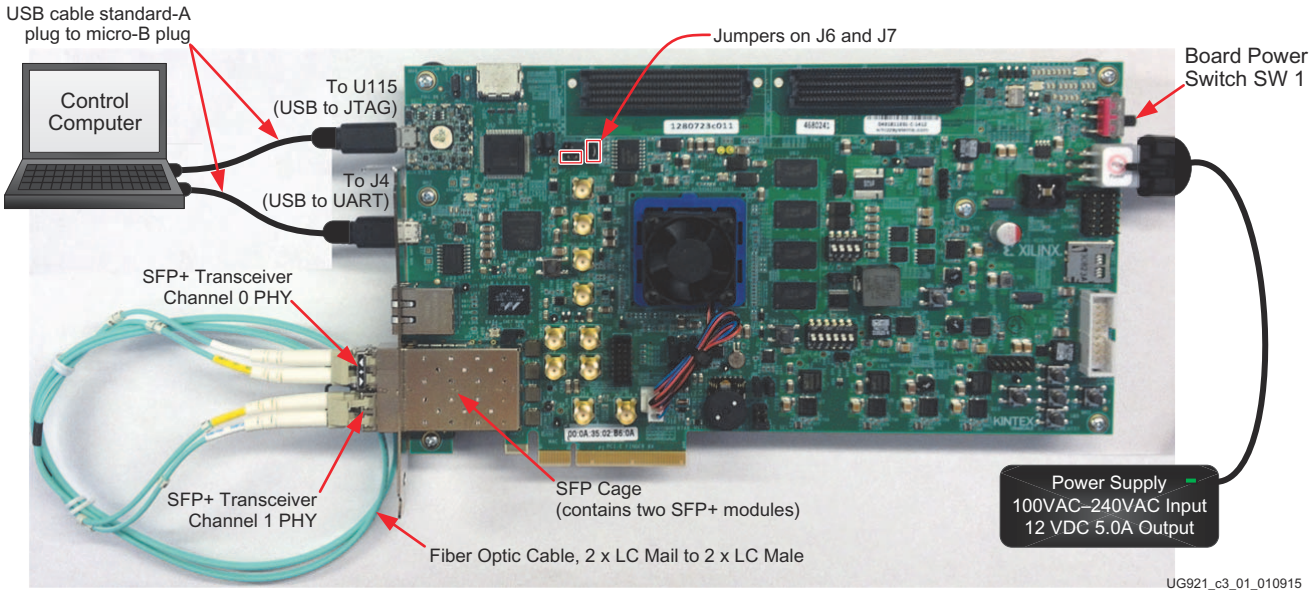


Figure 3-1: **Connection Diagram for Bring Up**

4. Power on the KCU105 board by placing switch SW1 to the ON position (SW1 in Figure 3-1).
5. Launch the Vivado Integrated Design Environment (IDE) on the control computer:
   a. Select **Start > All Programs > Xilinx Design Tools > Vivado 2015.4 > Vivado 2015.4**.
   b. On the getting started page, click **Open Hardware Manager** (Figure 3-2).

Figure 3-2: **Vivado IDE Getting Started Page, Open Hardware Manager**

6. Open the connection wizard to initiate a connection to the KCU105 board:
   a. Click **Open a new hardware target** (Figure 3-3).

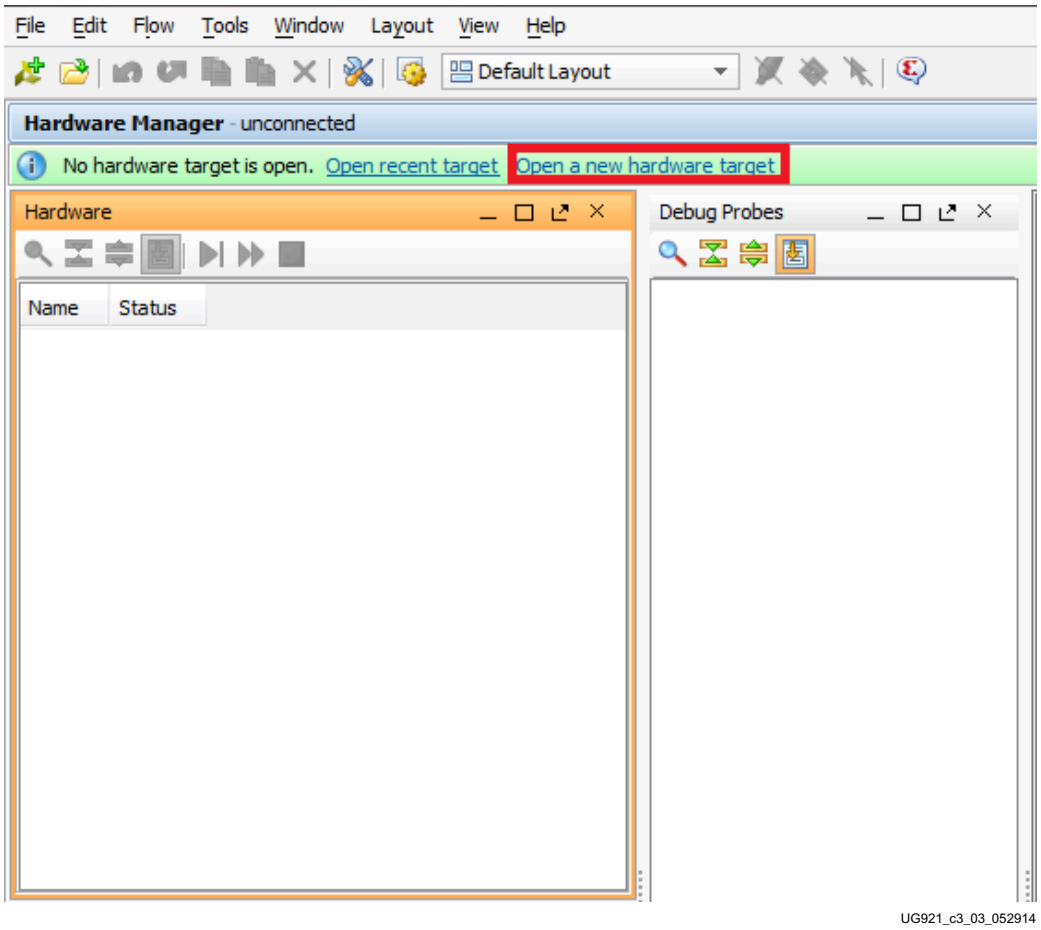


Figure 3-3: **Using the User Assistance Bar to Open a Hardware Target**

b. Configure the wizard to establish connection with the KCU105 board by selecting the default value on each wizard page. Click **Next > Next > Next > Finish**.

c. In the hardware view, right-click **xcku040** and click **Program Device** (Figure 3-4).

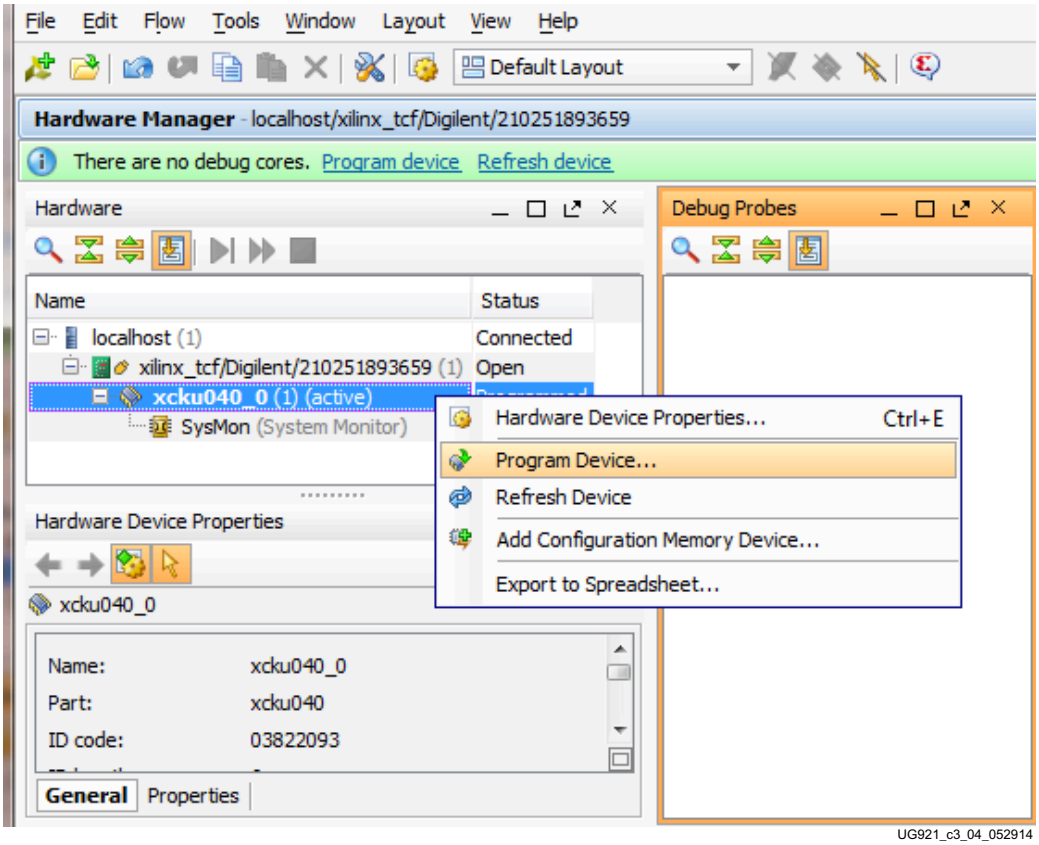


Figure 3-4: **Select Device to Program**

IMPORTANT: *There are no debug cores in this reference design. For this reason,* WARNING: [Labtools 27-3123] The debug hub core was not detected at User Scan Chain 1 or 3 *can be ignored.*

d. In the **Bitstream file** field, browse to the location of the BIT file:

```
<working_dir>/kcu105_10gbaser_trd/ready_to_test/kcu105_10gbaser_download.bit
```

and click **Program** (Figure 3-5).

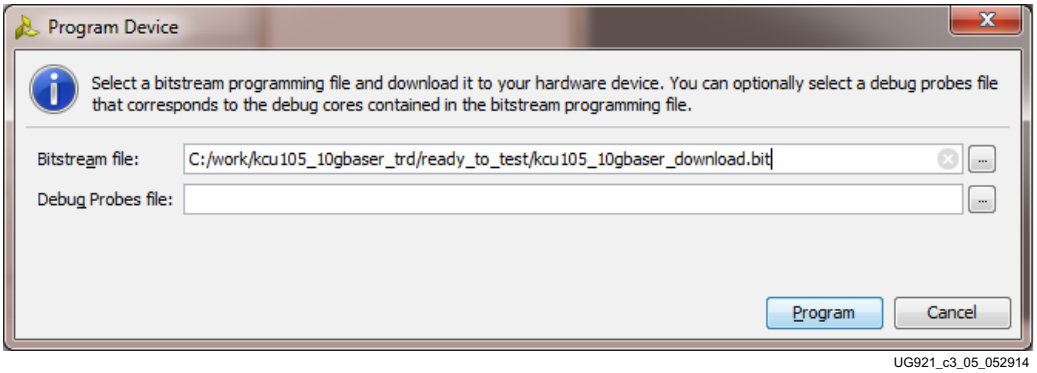


Figure 3-5: **Program Device Window**

After completing these steps, continue on to Running the Design.

Running the Design

Launch the Ethernet Controller Application

1. Launch the Ethernet Controller application on the control computer. In Windows, click **Start > All Programs > Xilinx > EthernetController** (Figure 3-6).

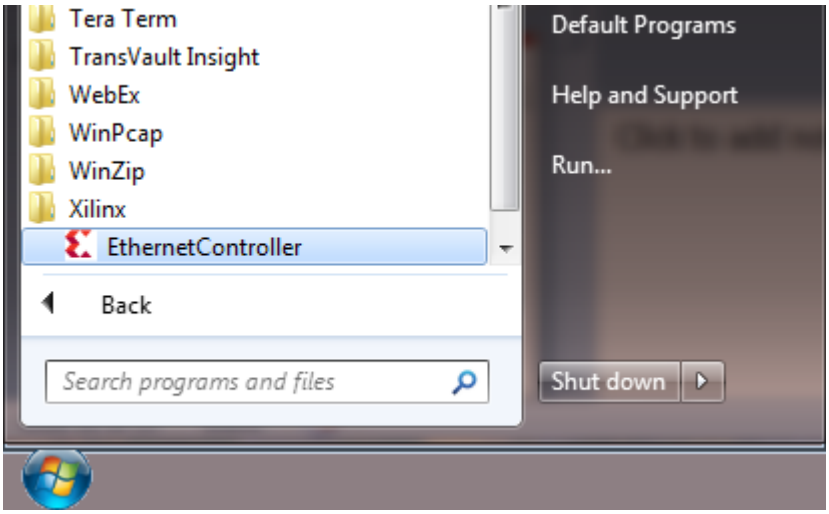


Figure 3-6: **Ethernet Controller Application**

2. Select the COM port associated with the Silicon Labs CP210x USB to UART Bridge and click **Connect** (Figure 3-7) to open the Ethernet Controller application for the 10GBASE-R TRD.

TIP: *The COM port associated with the Silicon Labs CP210x USB to UART Bridge can be identified using the Windows Device Manager. See step 3, page 11.*

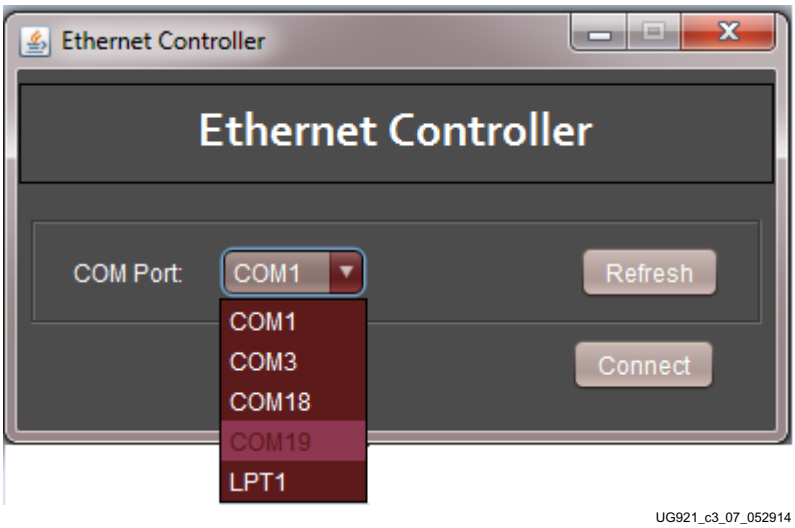


Figure 3-7: **Select COM Port Associated with the USB to UART Bridge**

Running Internal Traffic Generators

1. Ethernet channel 0 and channel 1 are up and ready when the ETH0 PHY and ETH1 PHY indicators are green (Figure 3-8). In the control panel for both channel 0 and channel 1, select **Internal Generator** and enter **1500** in the payload field, and click **Start** for both channels.

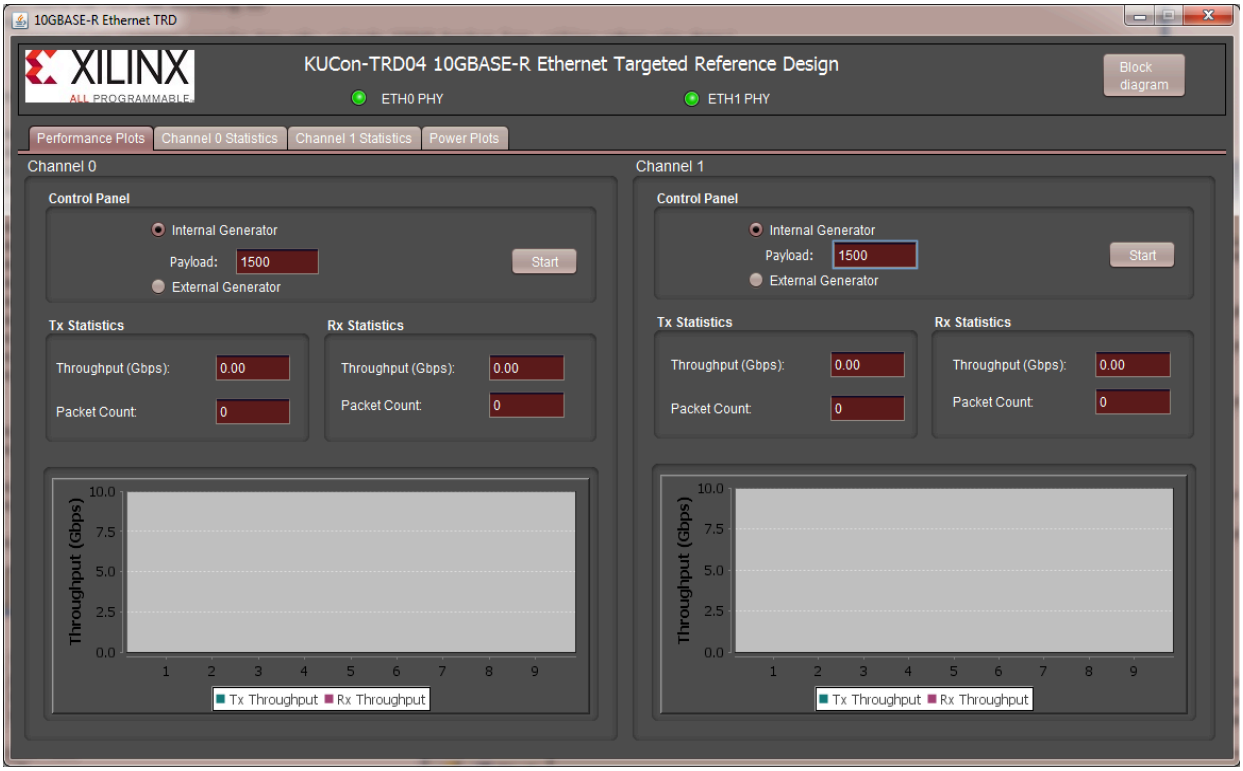


Figure 3-8: **Set Payload Size on Channel 0 and Channel 1**

Figure 3-9 shows the performance achieved at this payload size is 9.74 Gb/s per channel per direction. The allowed payload values that can be entered are 46 bytes to 1,500 bytes.

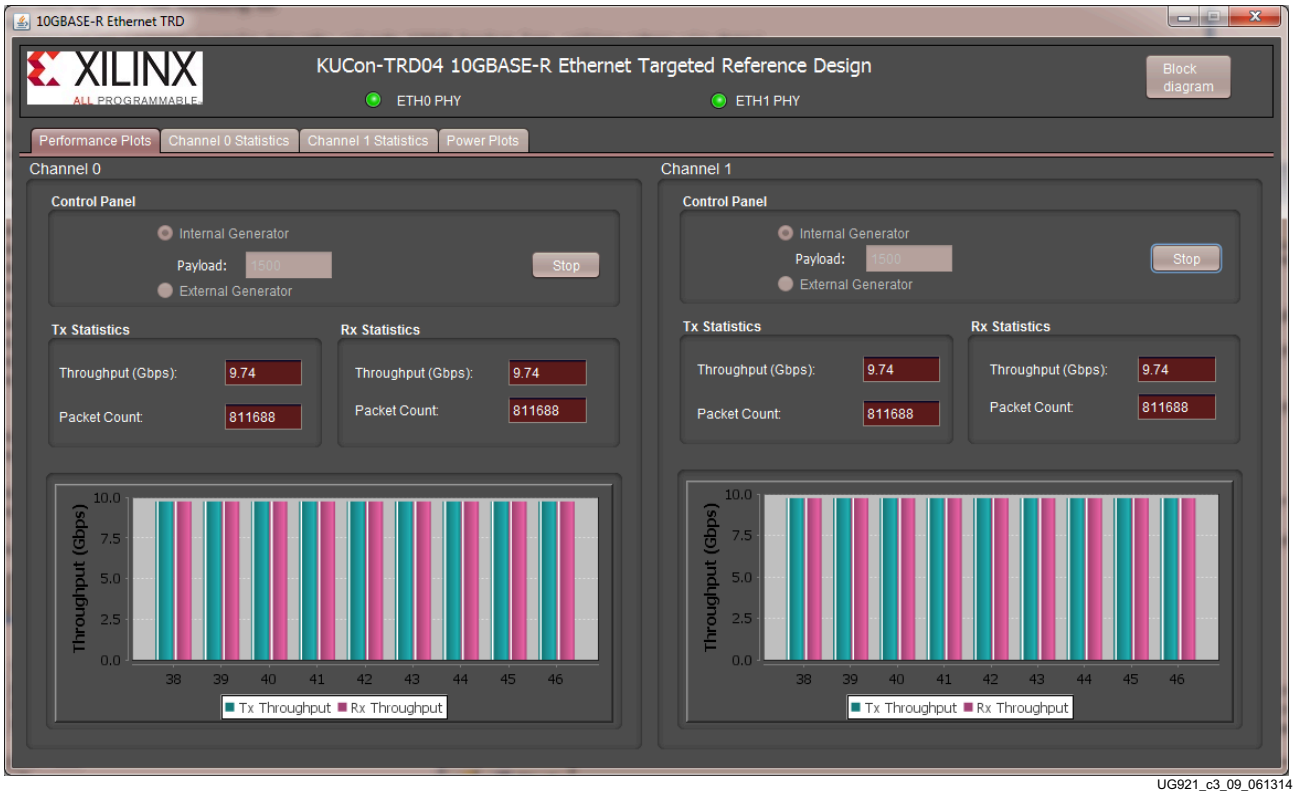


Figure 3-9: **Throughput Performance Plots**

TIP: *The relationship between payload size and throughput can be demonstrated by changing the payload size. Reducing the payload size causes a dip in performance. Refer to Appendix B, Performance Estimates for performance estimation on 10G Ethernet protocol.*

2. Stop traffic generation by clicking **Stop** for both channels.
3. Select the **Channel 0 Statistics** tab and verify if any packets were in error or were dropped (Figure 3-10).

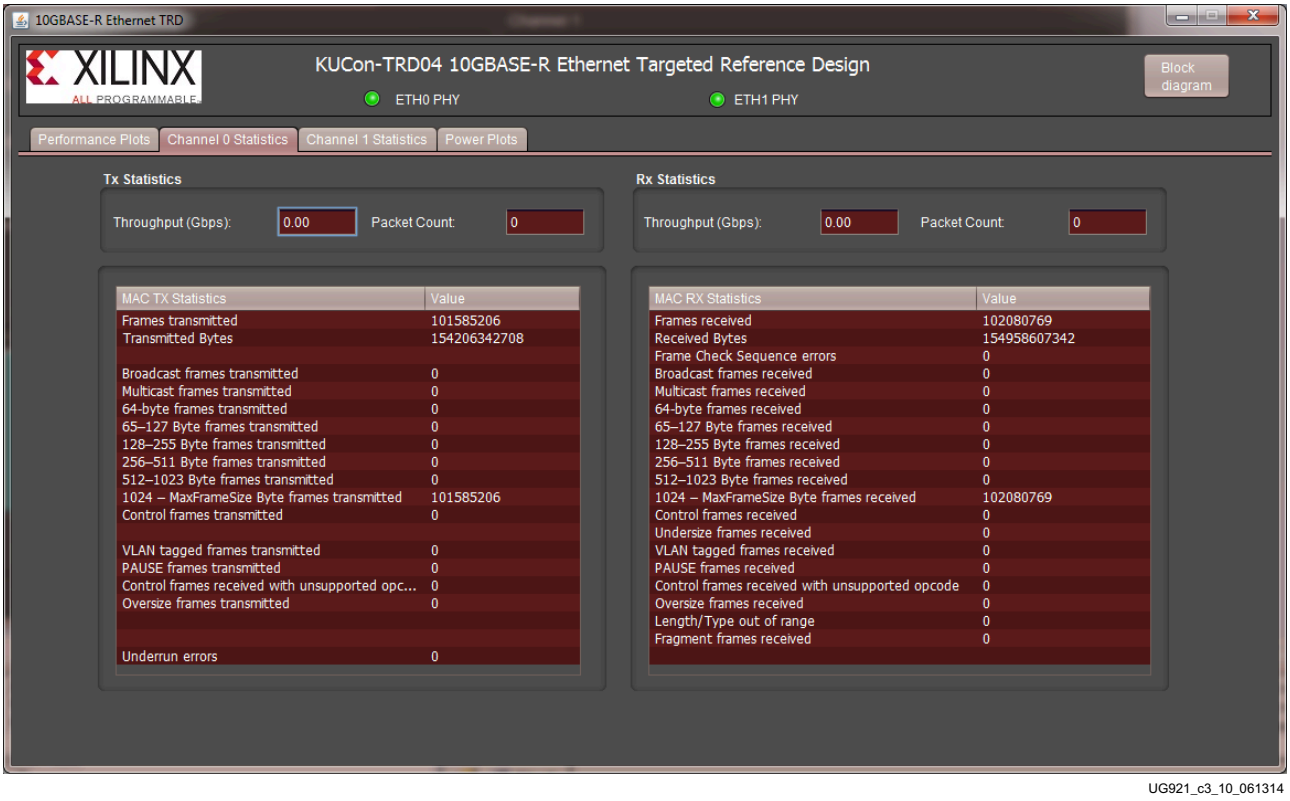


Figure 3-10: **Channel 0 MAC Statistics**

4. Next, select the **Channel 1 Statistics** tab and verify if any packets were in error or were dropped:
   - The TX MAC statistics for Channel 0 should match the RX MAC statistics of Channel 1.
   - The TX MAC statistics for Channel 1 should match the RX MAC statistics of Channel 0.

IMPORTANT: *When running traffic on both channels, If internal generator mode is selected on one channel, the internal generator mode must be selected on the other channel. If external generator mode is selected on one channel, the external generator mode must be selected on the other channel.*

TIP: *Appendix D, Testing with an External Traffic Generator describes how to setup and test the 10GBASE-R TRD using an external Ixia NGY-NP4-01 10 Gigabit Application Network Processor Load Module.*

5. Select the **Power Plot** tab to view the FPGA power consumption and die temperature (Figure 3-11).

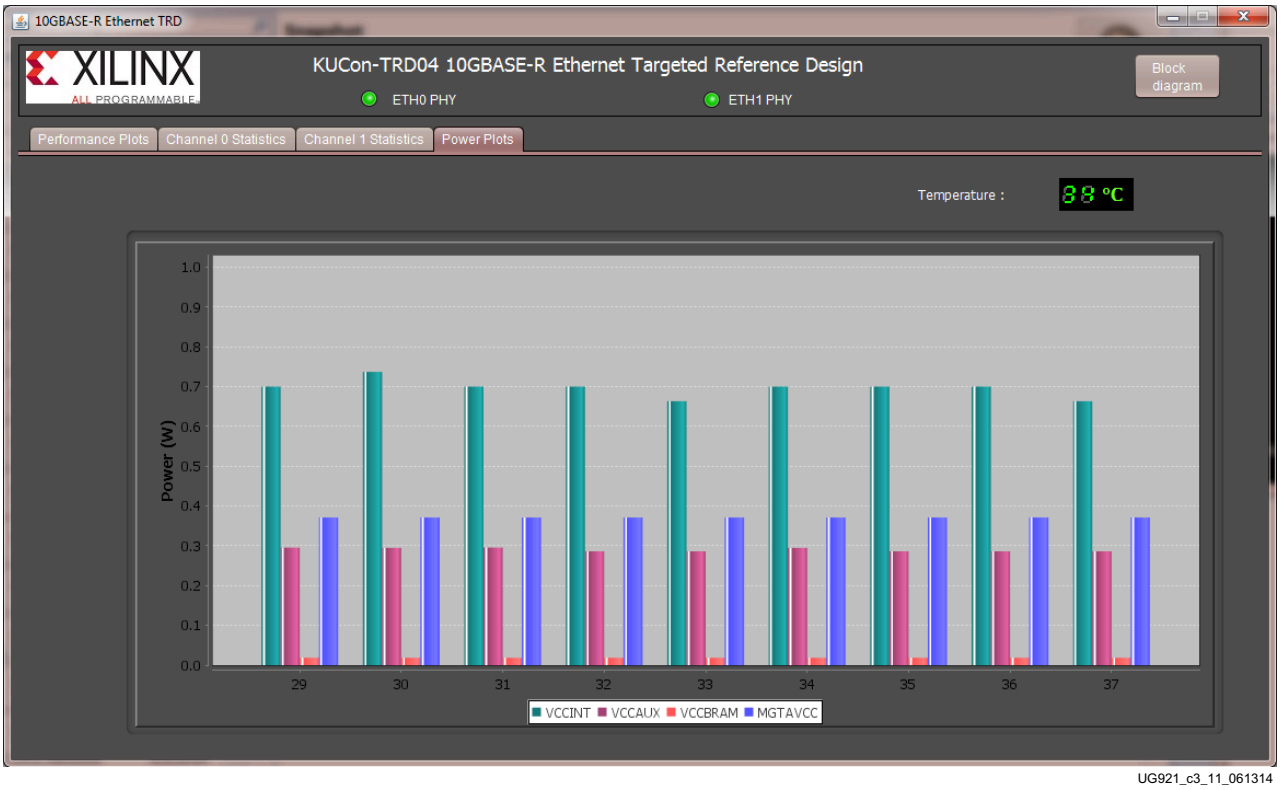


Figure 3-11: **Power Plots**

Implementing and Simulating the Design

This chapter describes how to implement and simulate the 10GBASE-R TRD.

Implementing the Design

The 10GBASE-R TRD is implemented using Vivado® Design Suite.

Download the Reference Design Files

See Download Targeted Reference Design Files, page 10 for instructions.

TIP: *The Reference Design directory structure is described in Appendix A, Directory Structure.*

IMPORTANT: *The 10-Gigabit Ethernet MAC IP core requires a license to build the design. Obtain the license at the 10 Gigabit Ethernet Media Access Controller (10GEMAC) webpage [Ref 4]. Click **Evaluate** or **Order** to access the license.*

Generate the Hardware Bitstream

The reference design can be implemented or simulated on a Windows 7 or a Linux computer. The computer should have Vivado Design Suite 2015.4 installed on it.

1. Launch the Vivado Integrated Design Environment (IDE) on the control computer and set up the reference design project.

- On Windows 7:
  a. Click **Start > All Programs > Xilinx Design Tools > Vivado 2015.4 > Vivado 2015.4**.

b. On the getting started page, click **Window > Tcl Console** (Figure 4-1).

c. In the Tcl console type (Figure 4-1):

```
cd <working_dir>/kcu105_10gbaser_trd/hardware/vivado

source scripts/kcu105_10GBASER_trd.tcl
```

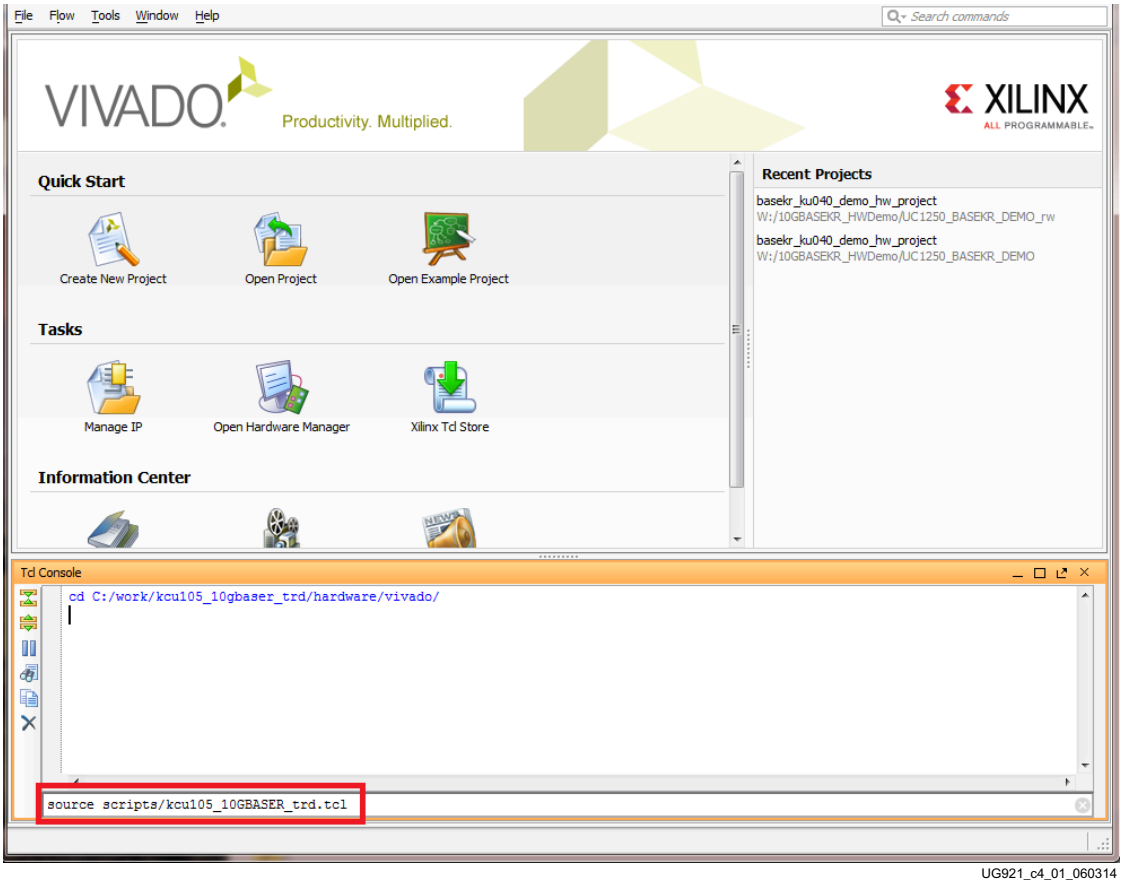


Figure 4-1: **Vivado IDE Getting Started Page, Open Hardware Manager**

- On Linux:

  a. On a terminal window, change directory to

  ```
  <working_dir>/kcu105_10gbaser_trd/hardware/vivado
  ```

  b. At the command prompt enter:

  ```
  vivado -source scripts/kcu105_10GBASER_trd.tcl
  ```

  The Vivado IDE will display the `10gbaser_trd` project populated with sources (Figure 4-2).

IMPORTANT: *When building the design on Windows, if this error occurs:* ***Path Length Exceeds 260-Bytes maximum allowed by Windows****, move the* `kcu105_10gbaser_trd` *directory directly under* `C:\`.

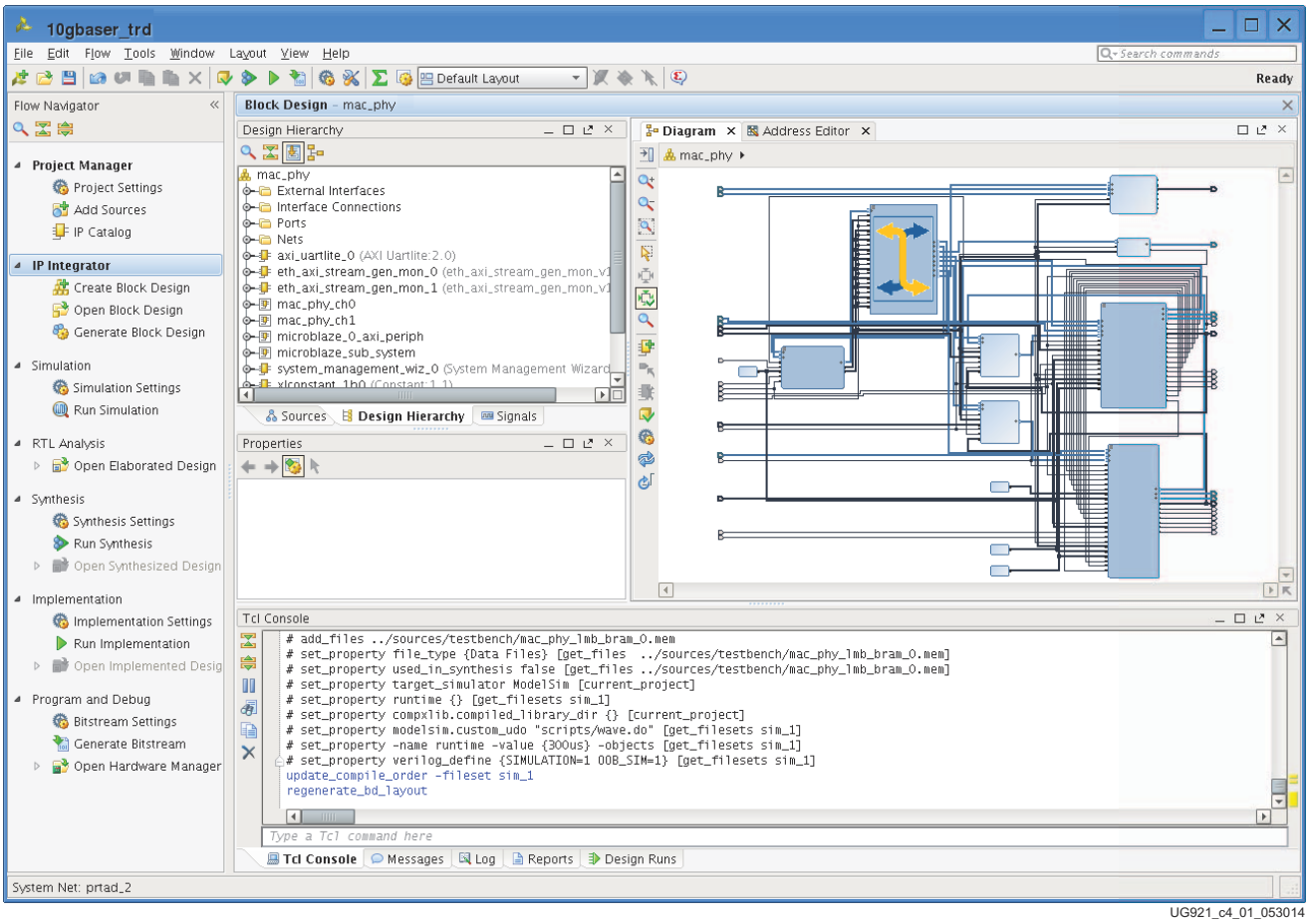


Figure 4-2: **10gbaser_trd Vivado project**

2. Select the **Sources** tab.

   In the Hierarchy window, an IP Integrator (IPI) block design (`mac_phy.bd`) is referenced that contains the 10-Gigabit Ethernet MAC IP core (10G MAC), 10-Gigabit Ethernet PCS/PMA IP core (10GBASE-R), the Traffic Generator and Monitor, and the MicroBlaze™ processor subsystem (Figure 4-3).

   The design top level file `kcu105_10gbaser_top.v` instantiates the block design.

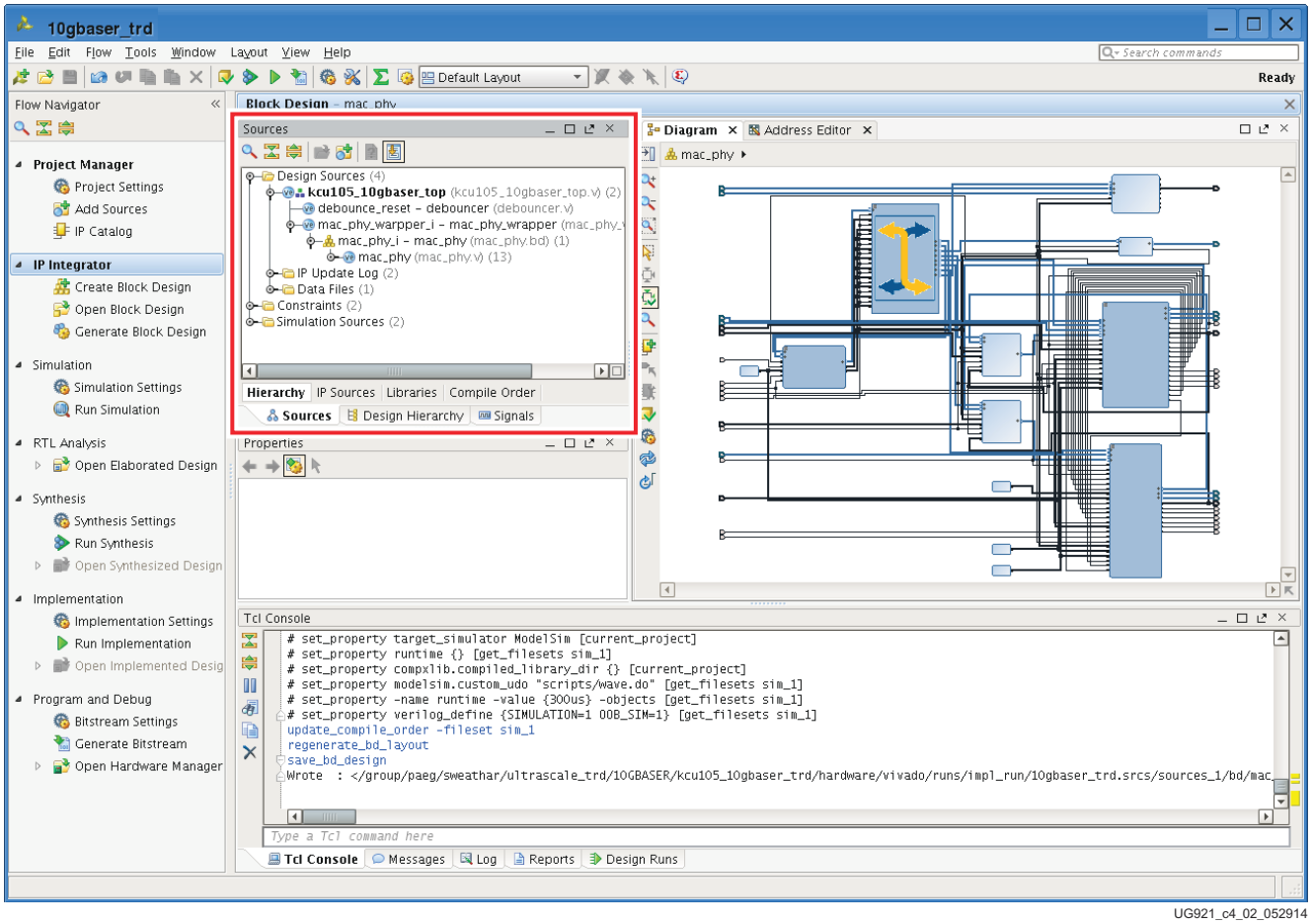


Figure 4-3: Vivado Project, Sources View

3. In the Flow Navigator, click **Generate Bitstream**.
4. In the No Implementation Results Available window, Click **Yes**. The bitstream will be generated and available at:

   `<working_dir>/kcu105_10gbaser_trd/hardware/vivado/runs/impl_run/10gbaser_trd.runs/impl_1/kcu105_10gbaser_top.bit`

Generate ELF file for the MicroBlaze Controller

To generate the MicroBlaze Controller ELF file, the design is exported to the Software Development Kit (SDK). The SDK is an IDE for software developers built on the Eclipse IDE.

1. From the Vivado IDE menu bar, select **File > Export > Export Hardware** (Figure 4-4).

IMPORTANT: *The IPI Block Design, `mac_phy.bd` must be open to successfully export the design to the SDK.*

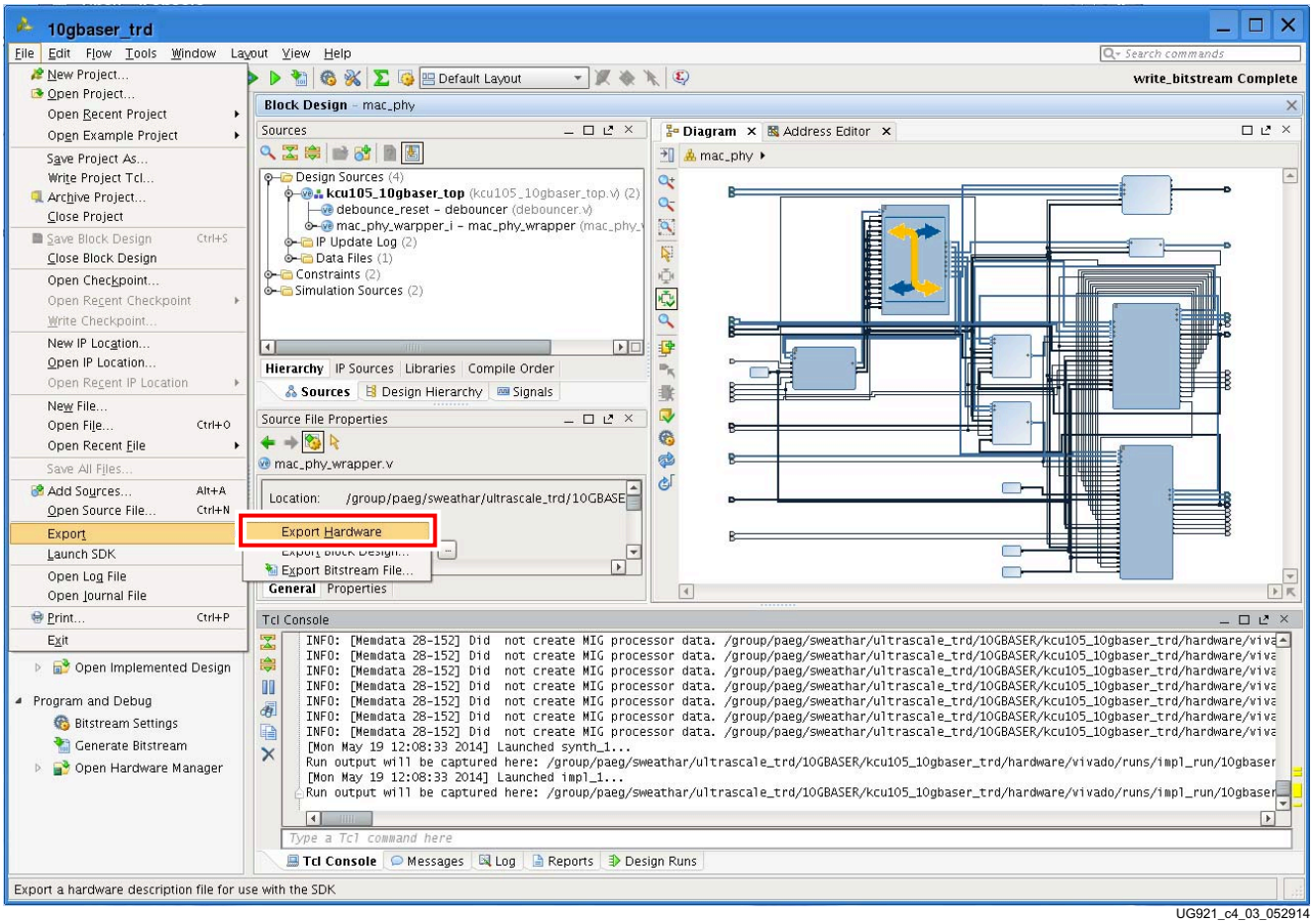


Figure 4-4: **Export Hardware to the SDK from Vivado**

2. In the Export Hardware Platform for SDK window (Figure 4-5) click **OK**. The hardware platform will be exported to:

   ```
   <working_dir>/kcu105_10gbaser_trd//hardware/vivado/runs/impl_run
   /10gbaser_trd.sdk
   ```

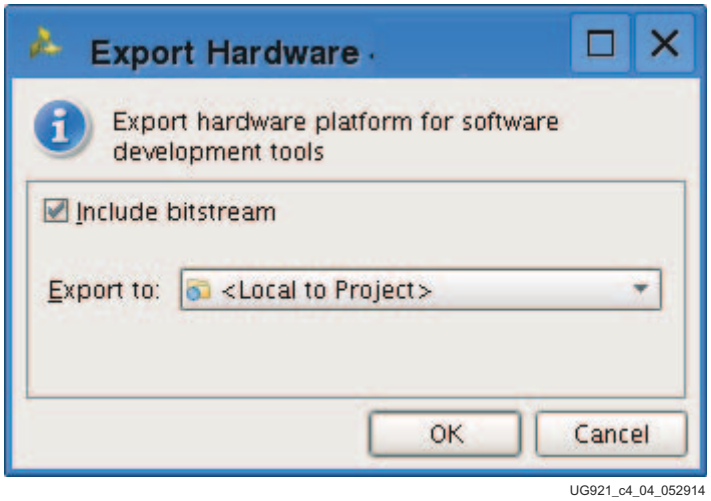


Figure 4-5: **Export Hardware**

3. To launch SDK from the Vivado IDE menu bar, select **File > Launch SDK**. In the launch SDK window click **OK**, SDK window will open with the hardware platform populated (Figure 4-6).

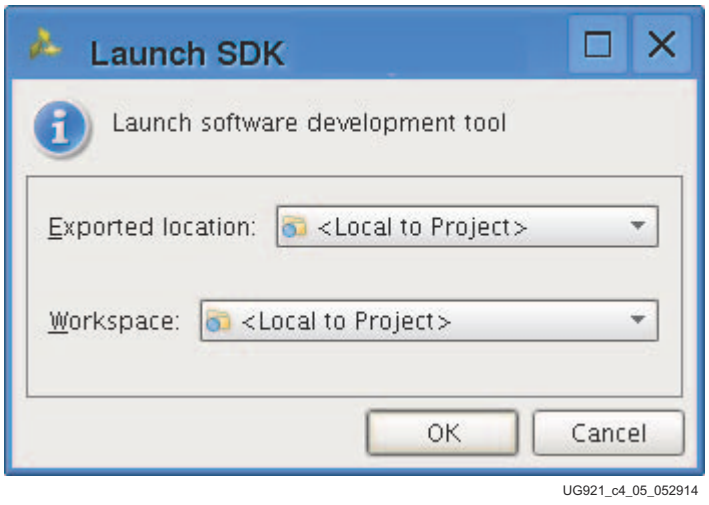


Figure 4-6: **Launch SDK**

4. In the SDK window (Figure 4-7) select **File > New > Application Project** to build an application.

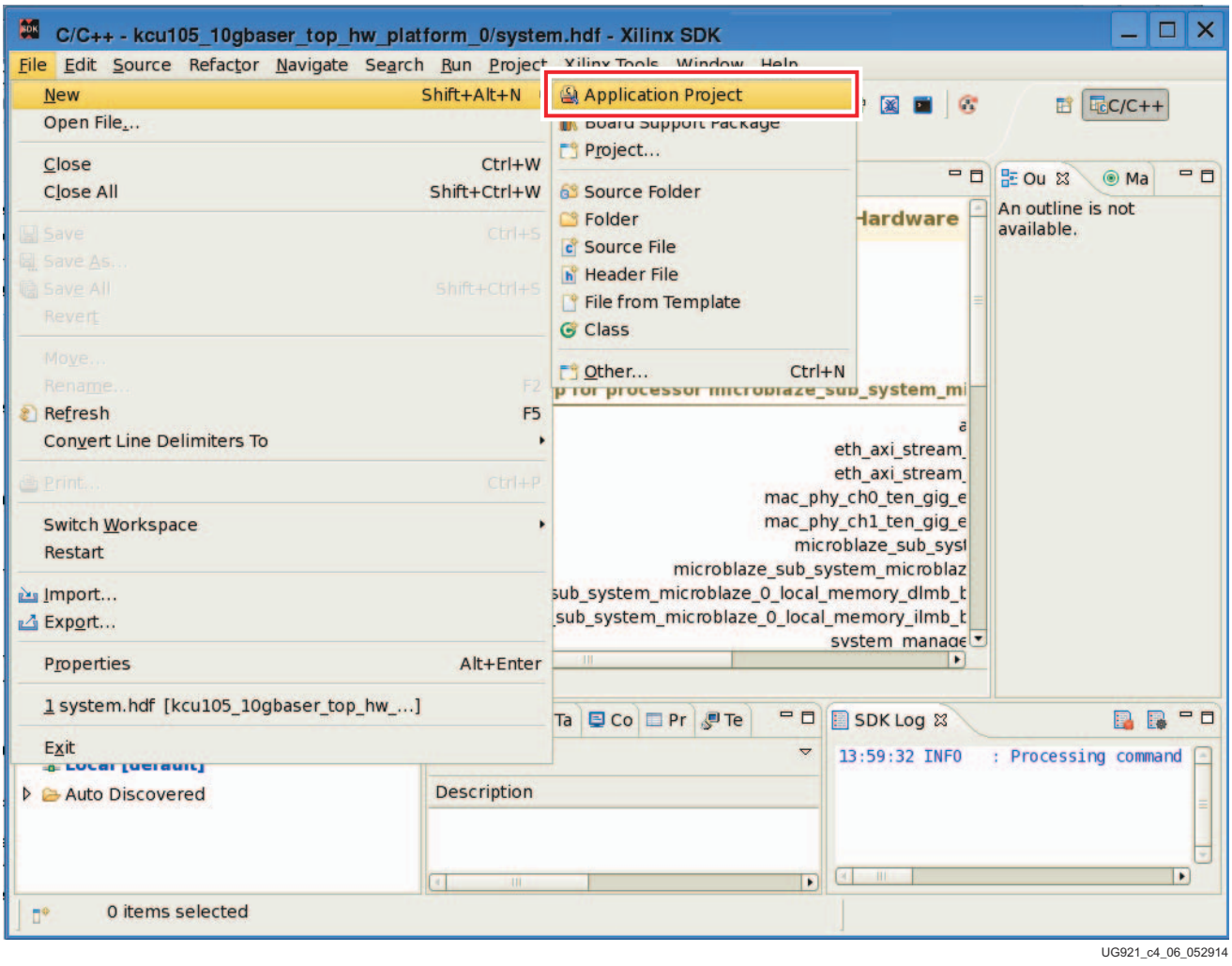


Figure 4-7: **Creating an Application Project in the SDK**

5. In the Application Project window (Figure 4-8) enter the project name as **kcu105_10gbaser_top** and click **Next**.

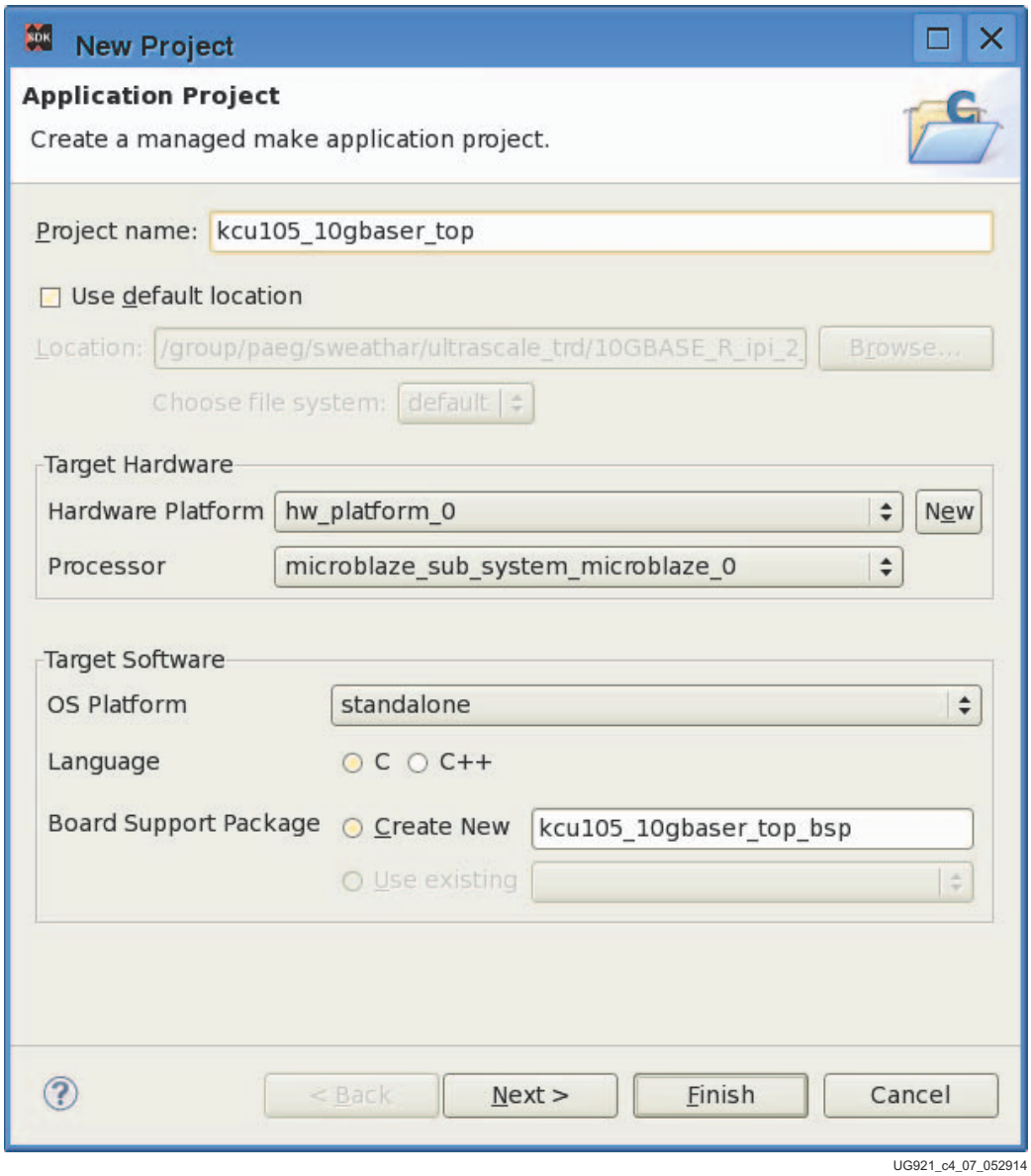


Figure 4-8: **Assign Project Name**

6. Select **Empty Application** and click Finish (Figure 4-9).

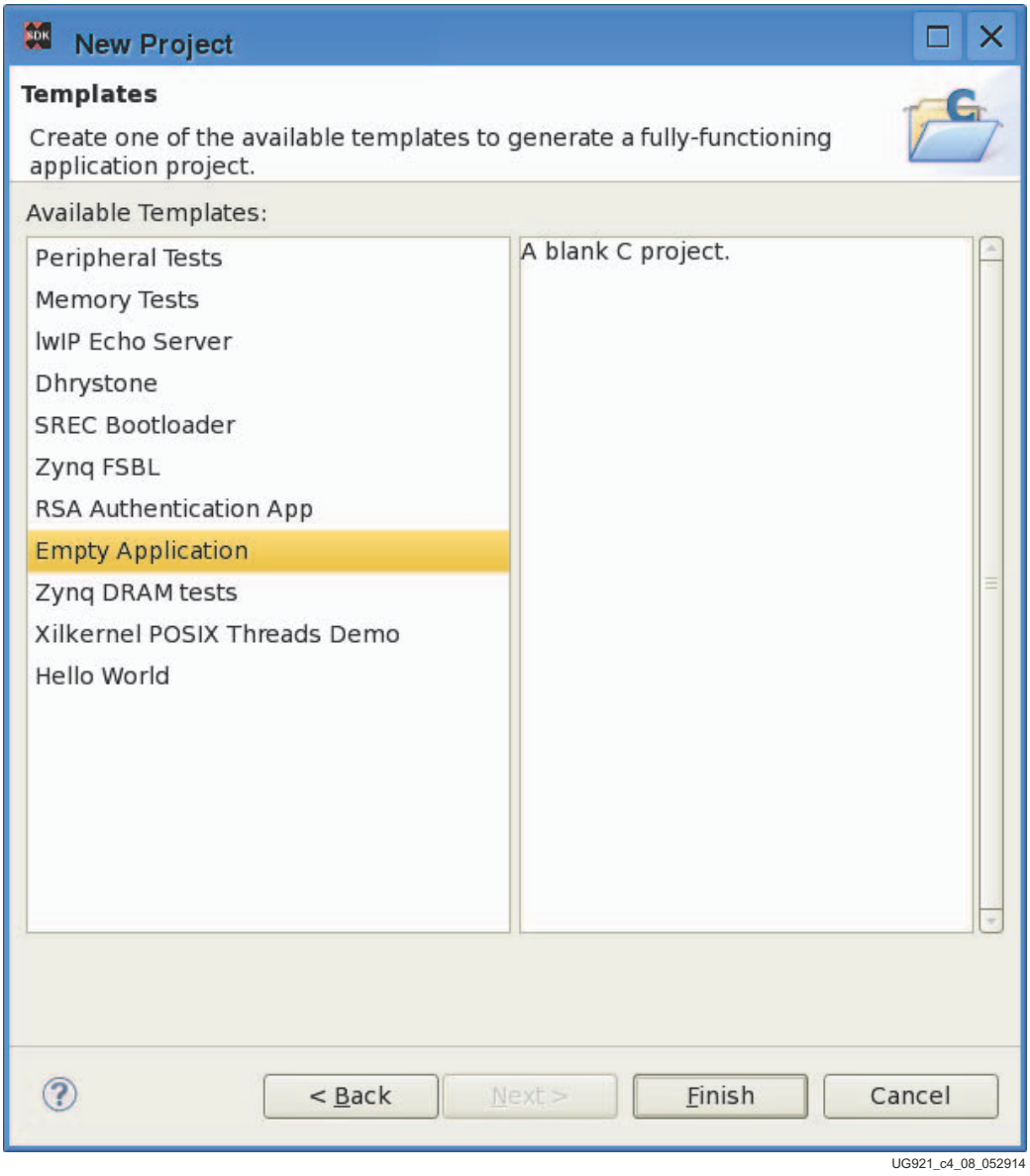


Figure 4-9: **Select Empty Application**

7. In the project explorer tab (Figure 4-10), right-click **kcu105_10gbaser_top**, select **Import**, and under the General tab select **File System**. Click **Next**.

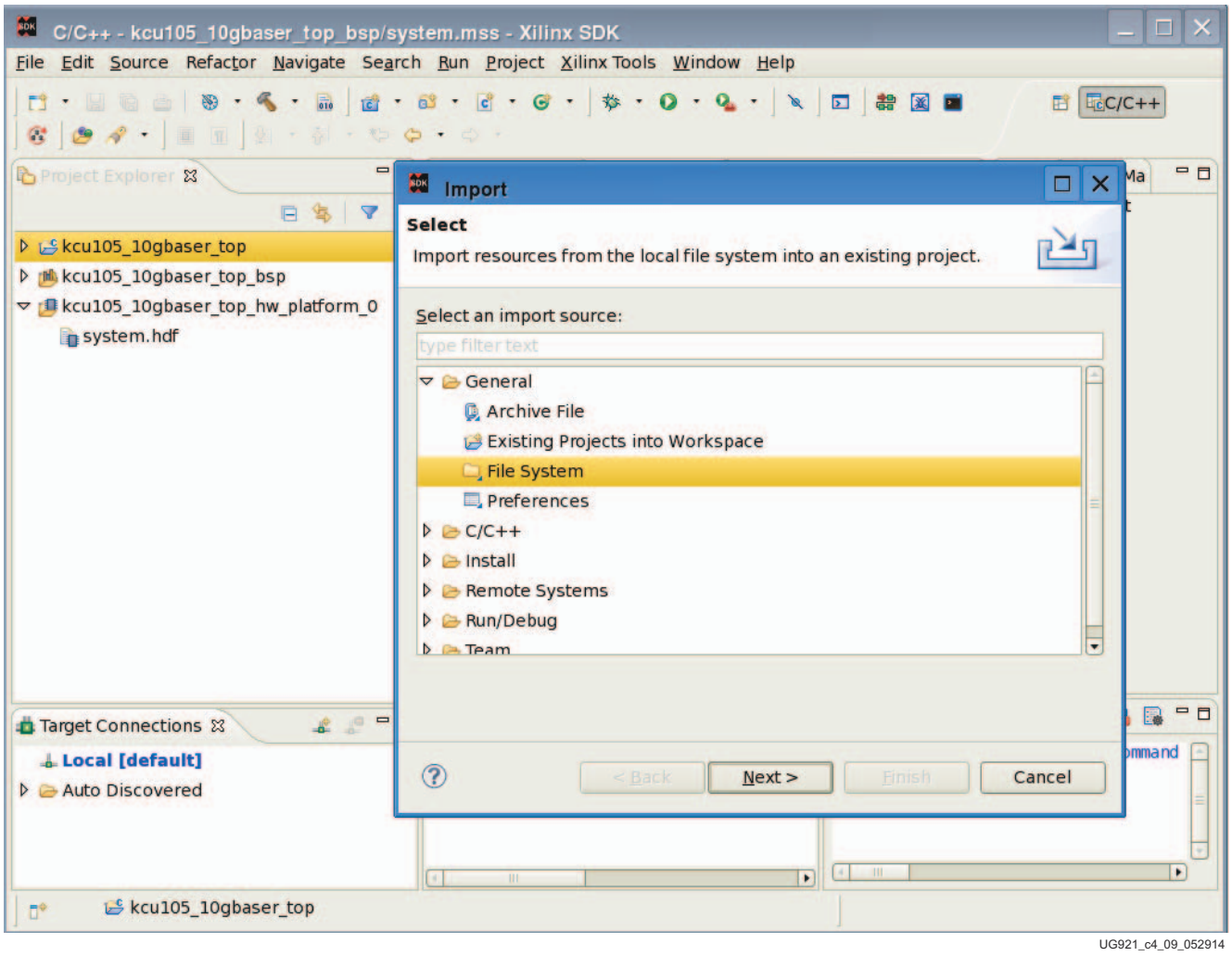


Figure 4-10: Importing File System

8. Browse to source folder:

   `<working_dir>/kcu105_10gbaser_trd/software/source`

   Select the source directory in the left pane and click **Finish** (Figure 4-11).

   The application ELF file will be generated and available at:

   `<working_dir>/kcu105_10gbaser_trd/hardware/vivado/runs/impl_run/10gbaser_trd.sdk/kcu105_10gbaser_top/Debug/kcu105_10gbaser_top.elf`

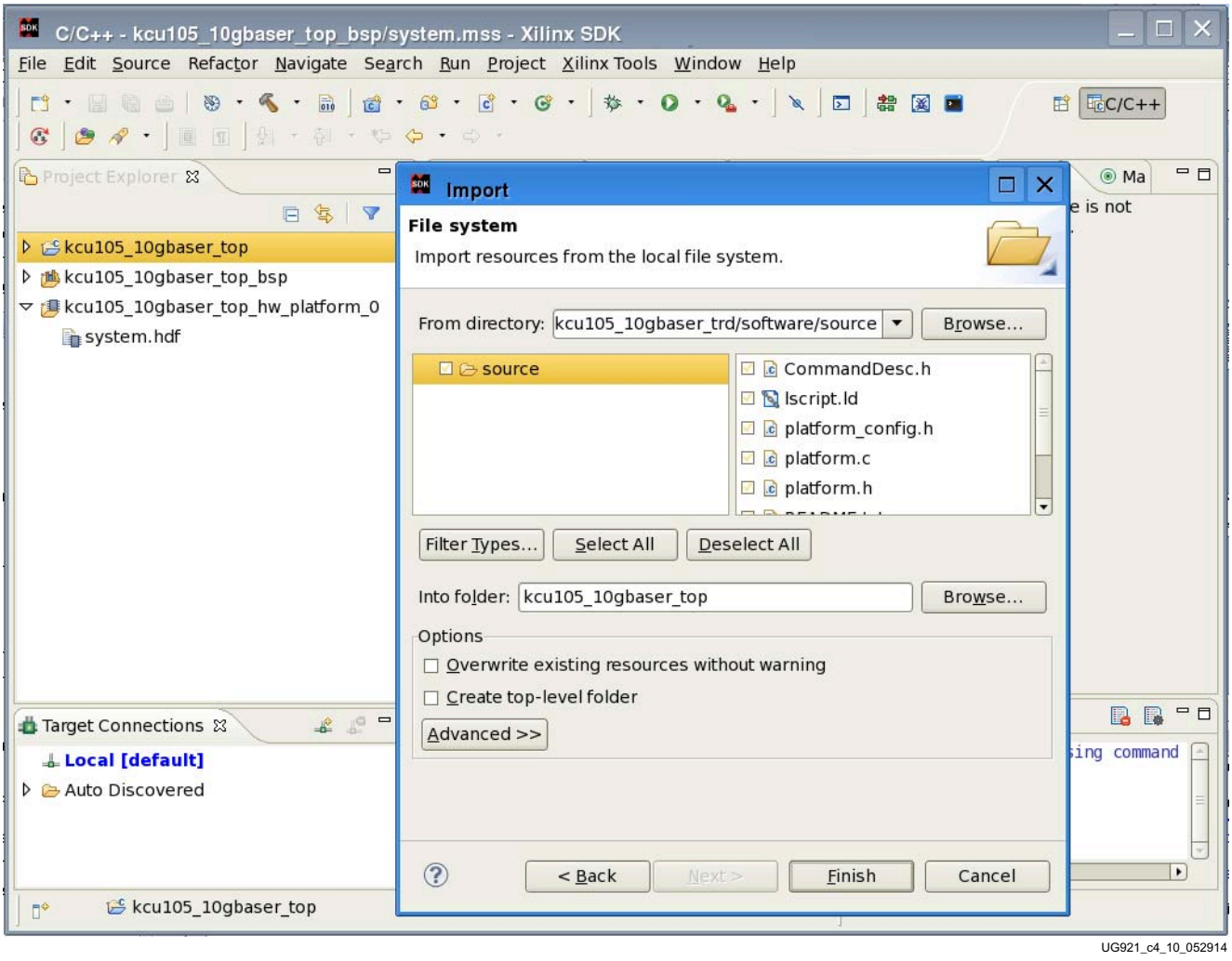


Figure 4-11: **Importing software/source Directory**

Generate Bitstream For Download

1. Create a bitstream for download. In the vivado Tcl Console (Figure 4-12) run the command:

```
source scripts/create_download_bit.tcl
```

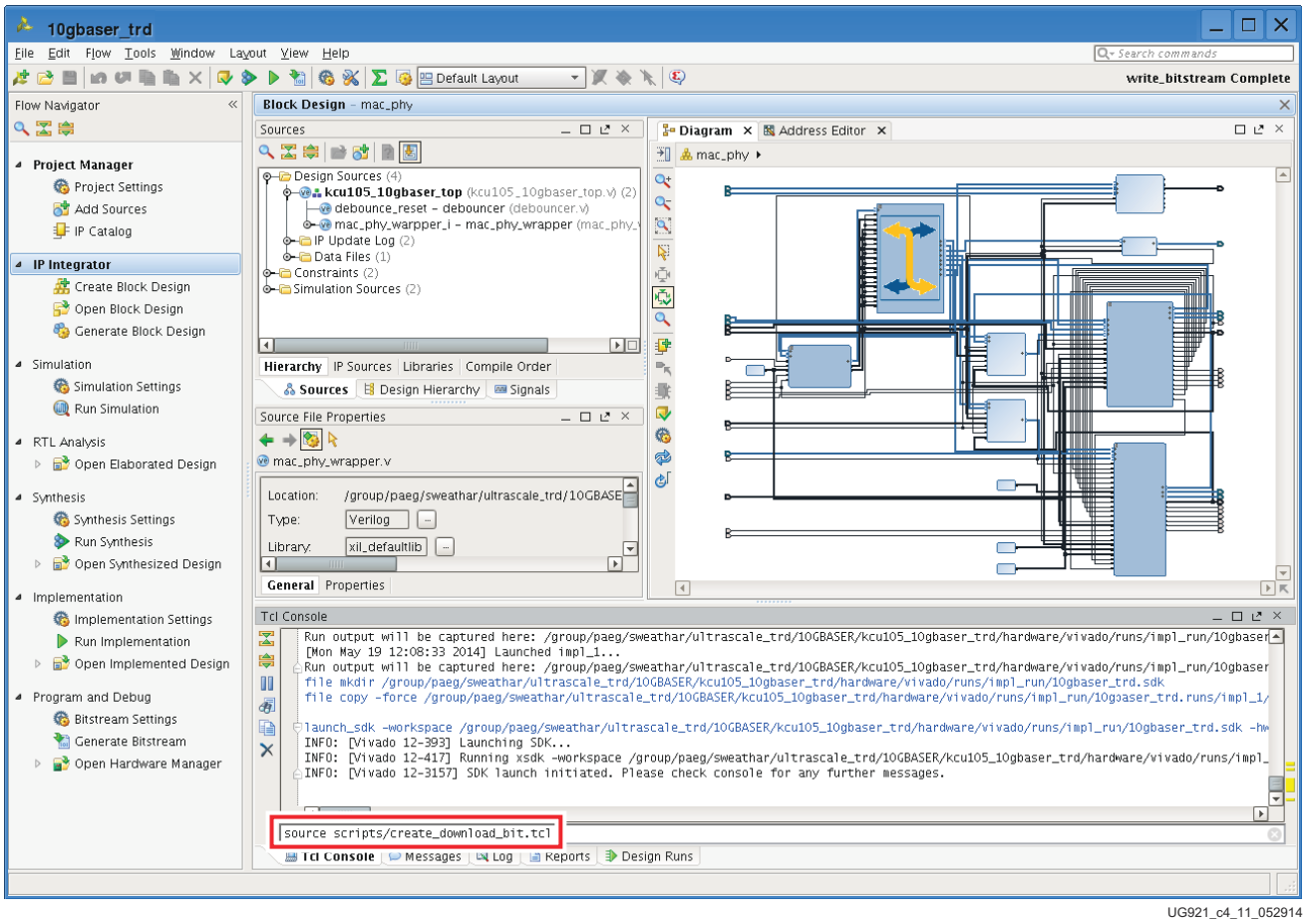


Figure 4-12: **Run Script to Create download.bit**

The `create_download_bit.tcl` script runs the update_mem command and combines `kcu105_10gbaser_top.bit` and `kcu105_10gbaser_top.elf` into single bitfile available at:

```
<working_dir>/kcu105_10gbaser_trd/hardware/vivado/runs/impl_run/10g
baser_trd.runs/impl_1/kcu105_10gbaser_download.bit
```

Simulating the Design

The 10GBASE-R TRD can be simulated using the Vivado Design Suite simulator. Refer to *Vivado Design Suite User Guide Logic Simulation* (UG900) [Ref 7], for information describing how to run simulation with different simulators.

The simulation environment sets up the Traffic Generator and Monitor blocks of the TRD to operate in internal generator mode. The Traffic Generator for channel 0 generates 10 packets which are transmitted to the 10-Gigabit Ethernet MAC IP core. The packets are looped back on the PHY and become receive packets on channel 1. Similarly, the Traffic Generator for channel 1 generates 10 packets which are transmitted to the 10-Gigabit Ethernet MAC IP core. The packets are looped back on the PHY and become receive packets on Channel 0. The testbench waits to receive 10 packets on each channel without errors and then ends the simulation with a `Simulation Passed` message.

Simulating the AXI UART Lite IP and MicroBlaze processor subsystem takes a lot of time. In order to speed up simulation the Traffic Generator and Monitor block is not configured using its AXI Lite interface connected to the MicroBlaze processor subsystem. The Traffic Generator and Monitor block provides a port tg_config to configure the block. This port is used only for simulation. Table 4-1 shows the bitmap for this port.

Table 4-1: **Traffic Generator Configuration Port**

Bit Position	Description
0	Enable loopback for external generator mode.
1	Enable generator for internal generator mode.
31:16	Ethernet frame data payload size. Allowed values (46 bytes to 1,500 Bytes).

The testbench for the 10GBASE-R TRD is available at:

`<working_dir>/kcu105_10gbaser_trd/hardware/sources/testbench/tb.v`.

IMPORTANT: *Before running a simulation the 10gbaser_trd project must be open and step 1 under Generate the Hardware Bitstream should have been executed.*

To run simulation in Modelsim/Questa:

1. In the flow navigator panel, under Simulation, Click **Run Simulation > Run Behavioral Simulation** (Figure 4-13).

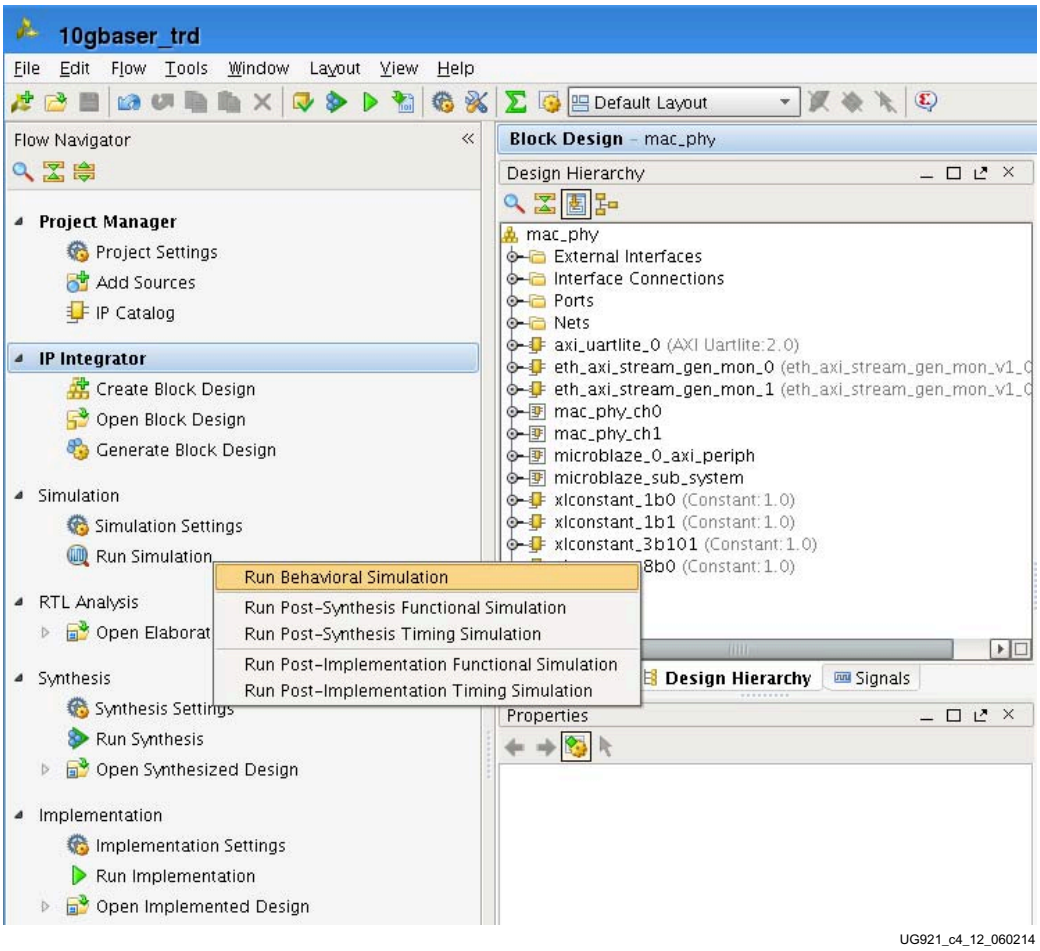


Figure 4-13: **Run Modelsim simulation**

To run a simulation in the Vivado Design Suite Simulator:

1. In the Flow Navigator Panel, under Simulation, Click **Simulation Settings**.
2. In the Project Settings window, select **Vivado Simulator** in the Target simulator field and click **Yes** when asked if it is OK to change your target simulator to Vivado Simulator. Click **OK** in the Project Settings window (Figure 4-14).

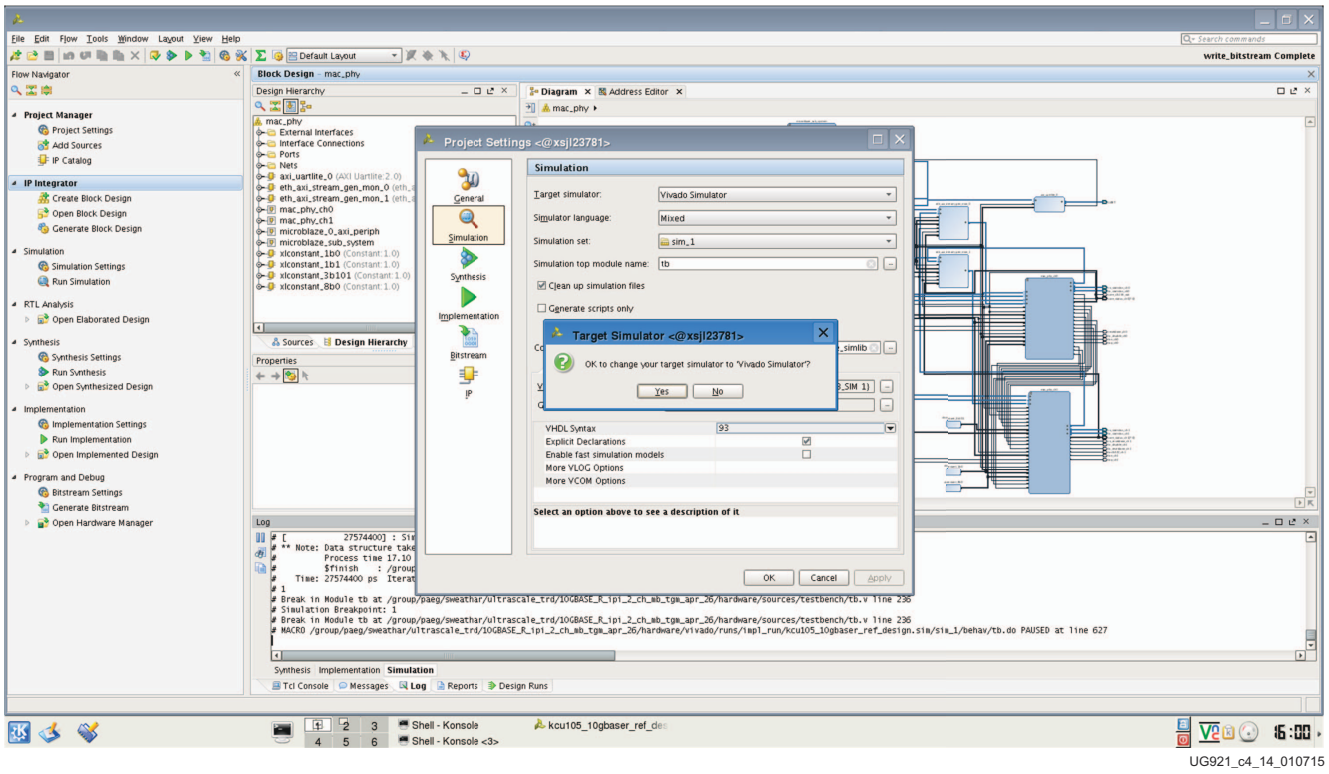


Figure 4-14: Set simulator to Vivado Simulator

3. In the Flow Navigator Panel, under Simulation, **Click Simulation > Run behavioral simulation**.

IMPORTANT: *When simulating the design on Windows, use this command to prevent the path length from exceeding 260 bytes:*

```
exec subst A:<working dir>\rdf0308-kcu105-trd04-2015-4\kcu105_10gbaser_trd\hardware
```

Reference Design Details

This chapter describes the hardware design and software components.

Hardware

Figure 5-1 shows a block-level overview of the 10GBASE-R TRD.

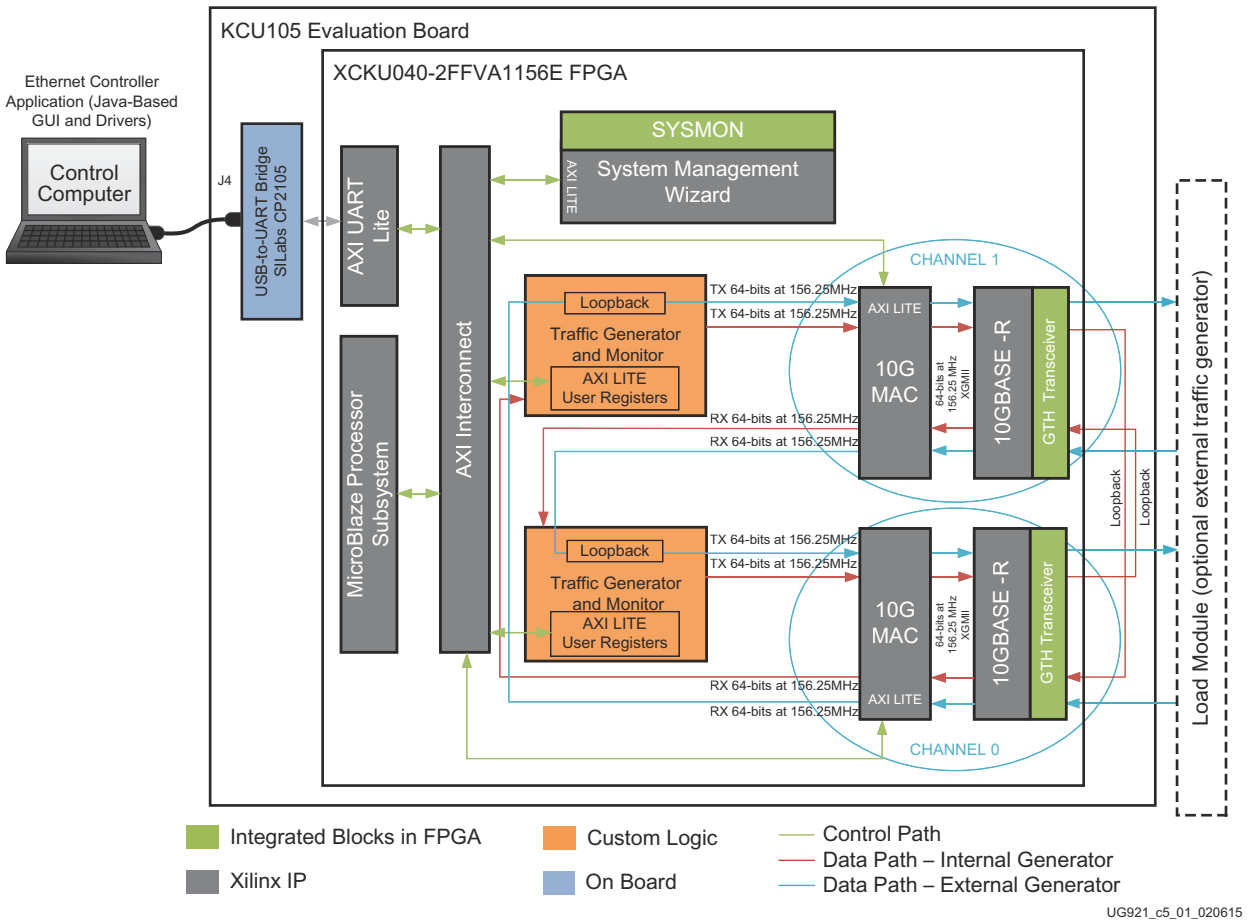


Figure 5-1: **10GBASE-R TRD Block Diagram**

The details of the hardware architecture are provided in three sections:

- Data Plane Components: Describes the 10-Gigabit Ethernet PCS/PMA IP core (10GBASE-R), 10-Gigabit Ethernet MAC IP core (10G MAC) and the Traffic Generator and Monitor.
- Control Plane Components: Describes the MicroBlaze™ processor subsystem and the peripherals connected to it.
- Clocking and Reset: Describes how clocks and resets are distributed to the different components in the 10GBASE-R TRD.

Data Plane Components

The 10-Gigabit Ethernet PCS/PMA IP (10GBASE-R) and 10-Gigabit Ethernet MAC IP (10G MAC) cores constitute a 10 Gb/s Ethernet channel. There are two channels in the 10GBASE-R TRD; channel 0 and channel 1. The data source for these channels can be configured to be either an internal or external Traffic Generator.

10-Gigabit Ethernet PCS/PMA IP Core

The 10-Gigabit Ethernet PCS/PMA IP core provides an XGMII interface to connect to a 10-Gigabit Ethernet MAC IP core and implements a 10.3125 Gb/s serial single-channel PHY providing a direct connection to an SFP+ optical transceiver module [Ref 18] using the SFI electrical specification. The SFP+ optical transceiver module plugs into an SFP cage on the KCU105 evaluation board. The external Traffic Generator communicates with this IP via the SFP+ interface.

More information is available at the *10 Gigabit Ethernet PCS/PMA (10GBASE-R) webpage* [Ref 3] and in the *LogiCORE IP 10-Gigabit Ethernet PCS/PMA Product Guide* (PG068) [Ref 5].

10-Gigabit Ethernet MAC IP Core

The 10-Gigabit Ethernet MAC IP core is a single-speed, full-duplex, 10-Gb/s Ethernet Media Access Controller. This 10G MAC connects to the PHY layer through the XGMII interface. The internal Traffic Generator drives data on the AXI4-Stream ports of this IP.

A license can be obtained at the *10 Gigabit Ethernet Media Access Controller (10GEMAC) webpage* [Ref 4]. More information is available in the *LogiCORE IP* 10-Gigabit Ethernet MAC *Product Guide* (PG072) [Ref 6].

Traffic Generator and Monitor

The data source for either of the two 10 Gb/s Ethernet channels can be configured to be from an internal or external Traffic Generator. The internal Traffic Generator consists of custom logic implemented in the FPGA that drives the 10-Gigabit Ethernet MAC. The external traffic generator is an off-the-shelf Ethernet traffic generator/checker like the

Ixia load module that can drive the 10-Gigabit Ethernet PCS/PMA IP core as described in Appendix D, Testing with an External Traffic Generator.

Figure 5-2 shows the block diagram of the Traffic Generator and Monitor.

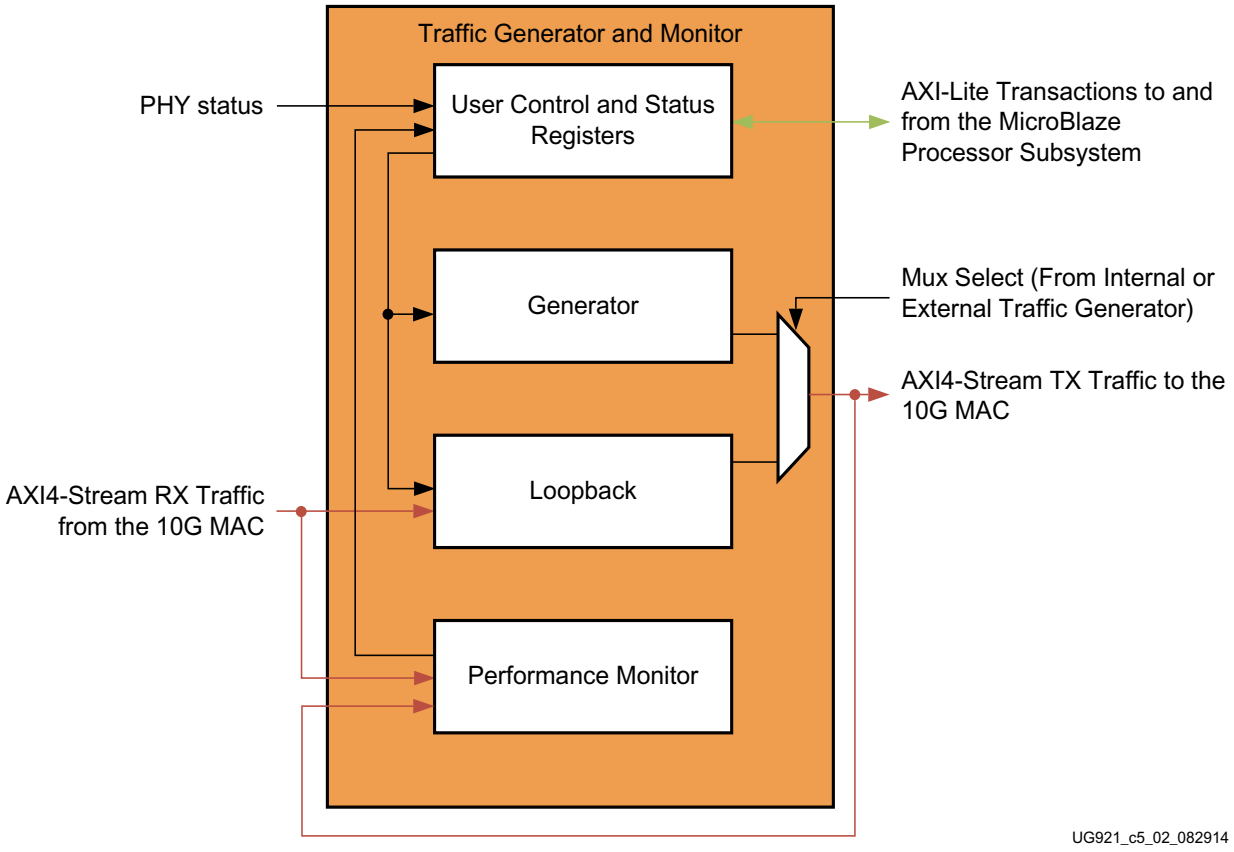


Figure 5-2: **Traffic Generator Block Diagram**

The Generator block generates Ethernet traffic when internal Traffic Generator mode is selected by the user.

The Loopback module loops back the data received from an external Ethernet traffic generator when external Traffic Generator mode is selected by the user.

The Performance Monitor block monitors the AXI4-Stream ports of the 10-Gigabit Ethernet MAC IP core and reports throughput.

The User Control and Status Registers block passes information to and from the Ethernet Controller application using the MicroBlaze processor subsystem.

Internal Traffic Generator, Generator Module

The Generator module generates Ethernet packets based on user inputs provided from the Ethernet Controller application running on the control computer. Data payload size can be

from 46 bytes to 1,500 bytes. Table 5-1 shows the packet format generated by the Generator module.

Table 5-1: **Packet Format of Generated Packets**

Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Source Address		Destination Address					
Sequence Number		Length		Source Address			
. . . . .				Sequence Number		Sequence Number	
Sequence Number		. . . . .					

The source and destination MAC addresses are parameters into this block. The length field is the data payload value. The actual length of a packet generated by this block is:

- 14 bytes of header (Destination Address + Source Address + Length/Type Field) + Data payload

The sequence number field indicates packet count and increments by one every packet.

The generated packets are transmitted on AXI4-Stream interface to the 10-Gigabit Ethernet MAC IP core. Table 5-2 shows the parameters and ports on the generator module.

Table 5-2: **Generator Module Parameters and Ports**

Port/Parameter Name	Type	Description
XIL_MAC_ID_THIS	Parameter	Source MAC Address For channel 0 = `48'h111100000000` For channel 1 = `48'h222200000000`
XIL_MAC_ID_OTHER	Parameter	Destination MAC Address For channel 0 = `48'h222200000000` For channel 1 = `48'h111100000000`
Clock and reset ports		
reset	Input	Synchronous reset.
tx_axis_clk	Input	156.25 MHz clock transmit ports on the AXI4-Stream interface.
Transmit Ports on the AXI4-Stream Interface		
tx_axis_tdata[63:0]	Output	Data to be transmitted to the 10-Gigabit Ethernet MAC IP core.
tx_axis_tkeep[7:0]		The transmit keep signal is used to determine which data bytes are valid on tx_axis_tdata during a given beat (this signal is valid only if tx_axis_tvalid and tx_axis_tready are both asserted). Bit 0 corresponds to the least significant byte on tx_axis_tdata and bit 7 corresponds to the most significant byte. When tx_axis_tlast is not asserted, the only valid value is `0xFF`. When tx_axis_tlast is asserted, valid values are `0x01` to `0xFF`.
tx_axis_tlast	Output	End of frame indicator on transmit packets. Valid only along with assertion of tx_axis_tvalid.

Table 5-2: **Generator Module Parameters and Ports *(Cont'd)***

Port/Parameter Name	Type	Description
tx_axis_tvalid	Output	Source ready to provide transmit data. Indicates that the generator is presenting valid data on tx_axis_tdata.
tx_axis_tuser	Output	If asserted indicates an underrun frame. This is tied to `1'b0`.
tx_axis_tready	Input	Destination ready for transmit. Indicates that the 10-Gigabit Ethernet MAC IP core is ready to accept data on tx_axis_tdata. The simultaneous assertion of tx_axis_tvalid and tx_axis_tready marks the successful transfer of one data beat on tx_axis_tdata.
Control Ports		
enable_gen	Input	Enable internal generator.
data_payload	Input	Size of the payload (46 bytes to 1,500 Bytes).

The data flow with internal generator mode enabled on the Traffic Generator for channel 0 is:

Generator module CH0 → CH0 TX AXI4-Stream interface of the 10-Gigabit Ethernet MAC IP → CH0 TX XGMII interface of the 10GBASE-R → CH0 TXN/TXP serial lines → loopback to CH1 RXN/RXP serial lines → CH1 RX XGMII interface of the 10GBASE-R → CH1 RX AXI4-Stream interface of the 10-Gigabit Ethernet MAC IP core

The data flow with internal generator mode enabled on Traffic Generator for channel 1 is:

Generator module CH1 → CH1 TX AXI4-Stream interface of the 10-Gigabit Ethernet MAC IP core → CH1 TX XGMII interface of the 10GBASE-R → CH1 TXN/TXP serial lines → loopback to CH0 RXN/RXP serial lines → CH0 RX XGMII interface of the 10GBASE-R → CH0 RX AXI4-Stream interface of the 10-Gigabit Ethernet MAC IP core

External Traffic generator, Loopback Module

The Traffic Generator is put into loopback mode when the user selects **External Generator** in the Ethernet Controller application control panel.

In this mode data is generated by an external generator like the Ixia load module. The data received on the AXI4-Stream RX port of the 10-Gigabit Ethernet MAC IP core is looped back to the other channel's AXI4-Stream TX port. The loopback module changes the source address and destination address on the received data before passing it to the 10-Gigabit Ethernet MAC IP core. The source and destination MAC addresses are parameters into the loopback module. Table 5-3 shows the parameters and ports on the loopback module.

Table 5-3: **Loopback Module Generator Parameters and Ports**

Port/Parameter Name	Type	Description
XIL_MAC_ID_THIS	Parameter	Source MAC Address For Channel 0 = `48'h111100000000` For Channel 1 = `48'h222200000000`
EXT_MAC_ID	Parameter	Destination MAC Address (External Generator) For Channel 0 = `48'h333300000000` For Channel 1 = `48'h444400000000`
Clock and reset ports		
reset	Input	Synchronous reset.
tx_axis_clk	Input	156.25 MHz clock Transmit Ports on the AXI4-Stream interface.
Transmit Ports on the AXI4-Stream Interface		
tx_axis_tdata[63:0]	Output	Data to be transmitted to 10-Gigabit Ethernet MAC IP core.
tx_axis_tkeep[7:0]		The transmit keep signal is used to determine which data bytes are valid on tx_axis_tdata during a given beat (this signal is valid only if tx_axis_tvalid and tx_axis_tready are both asserted). Bit 0 corresponds to the least significant byte on tx_axis_tdata and bit 7 corresponds to the most significant byte. When tx_axis_tlast is not asserted, the only valid value is `0xFF`. When tx_axis_tlast is asserted, valid values are `0x01` to `0xFF`.
tx_axis_tlast	Output	End of frame indicator on transmit packets. Valid only along with asser-tion of tx_axis_tvalid.
tx_axis_tvalid	Output	Source ready to provide transmit data. Indicates that the generator is presenting valid data on tx_axis_tdata.
tx_axis_tuser	Output	If asserted indicates an underrun frame. This is tied to `1'b0`
tx_axis_tready	Input	Destination ready for transmit. Indicates that the 10-Gigabit Ethernet MAC IP core is ready to accept data on tx_axis_tdata. The simultaneous assertion of tx_axis_tvalid and tx_axis_tready marks the successful transfer of one data beat on tx_axis_tdata.
Receive Ports on the AXI4-Stream Interface		
rx_axis_tdata[63:0]	Input	Data received from the 10-Gigabit Ethernet MAC IP core.
rx_axis_tkeep[7:0]		The receive keep signal is used to determine which data bytes are valid on rx_axis_tdata during a given beat (this signal is valid only if rx_axis_tvalid and rx_axis_tready are both asserted).
rx_axis_tlast	Input	End of frame indicator on received packets. Valid only along with assertion of rx_axis_tvalid.
rx_axis_tvalid	Input	Source ready to provide data. Indicates that the MAC is presenting valid data on rx_axis_tdata. rx_axis_tuser input if asserted indicates a good packet is received.

Table 5-3: **Loopback Module Generator Parameters and Ports *(Cont'd)***

Port/Parameter Name	Type	Description
rx_axis_tready	Output	Destination ready for receive. Indicates that the loopback module is ready to accept data on rx_axis_tdata. The simultaneous assertion of rx_axis_tvalid and rx_axis_tready marks the successful transfer of one data beat on rx_axis_tdata. The 10-Gigabit Ethernet MAC IP core doesn't look at this signal and sends received data whenever available.
Control Ports		
enable_loopback	Input	Enable external generator.

The 10-Gigabit Ethernet MAC IP core AXI4-Stream RX interface doesn't allow back-pressure i.e., after a packet reception has started it completes the entire packet (rx_axis_tready is disregarded). If transmit throttles, the receive side cannot stop. AXI4-Stream Data FIFO IP is added in the loopback path to counter this. The FIFO size (8 bytes wide x 256 depth) should accommodate maximum payload size (1,500 bytes). Packet mode is also selected on the FIFO which ensures that a full packet is present in the FIFO before transmission to the 10-Gigabit Ethernet MAC IP core.

The Data flow with loopback mode enabled on Traffic Generator for Channel 0 is:

IXIA TX CH1 → CH1 RXN/RXP serial lines → CH1 RX XGMII interface of the 10GBASE-R → CH1 RX AXI4-Stream interface of the 10G MAC → Loopback Module CH0 → CH0 TX AXI4-Stream interface of the 10G MAC → CH0 TX XGMII interface of the 10GBASE-R → CH0 TXN/TXP serial lines → IXIA RX CH0

The Data flow with loopback mode enabled on Traffic Generator for Channel 1 is:

IXIA TX CH0 → CH0 RXN/RXP serial lines → CH0 RX XGMII interface of the 10GBASE-R → CH0 RX AXI4-Stream interface of the 10G MAC → Loopback Module CH1 → CH1 TX AXI4-Stream interface of the 10G MAC → CH1 TX XGMII interface of the 10GBASE-R → CH1 TXN/TXP serial lines → IXIA RX CH1

Ethernet Performance Monitor

The Ethernet performance monitor block snoops for valid transactions on the AXI4-Stream interface ports of the 10-Gigabit Ethernet MAC IP core and keeps track of bandwidth utilization. A timer within this block counts the clocks until one second has elapsed, during which time counters have collected data about link performance.

Four counters collect information on the transactions on the AXI4-Stream interface:

- TX Payload Byte Count. This counter counts bytes transferred when tx_tvalid and tx_tready signals are asserted between the Traffic Generator block and the 10G MAC. At the end of the packet (tx_tlast) 14 bytes of header are subtracted from the count to get payload count.

- TX Packet Count. This counter counts the number of transmitted packets. The counter increments when tx_tvalid and tx_tready and tx_tlast signal are asserted.
- RX Payload Byte Count. This counter counts bytes transferred when rx_tvalid and rx_tready signals are asserted between the Traffic Generator block and the 10G MAC. At the end of the packet (rx_tlast) 14 bytes of header are subtracted from the count to get payload count.
- RX Packet Count. This counter counts the number of received packets. The counter increments when rx_tvalid and rx_tready and rx_tlast signal are asserted.

The counts are truncated to a four-byte resolution, and the last two bits of the register indicate the sampling period. The last two bits transition every second from `00` to `01` to `10` to `11`. The software polls the performance registers every second. If the sampling bits are the same as the previous read, then the software needs to discard the second read and try again. When the one-second timer expires, the new byte counts are loaded into the registers, overwriting the previous values. Table 5-4 shows the parameters and ports on this module

Table 5-4: **Ethernet Performance Monitor Parameters and Ports**

Port/Parameter Name	Type	Description
ONE_SEC_CLOCK_COUNT	Parameter	Defines the number of 156.25 MHz clock cycles equivalent to 1 sec. Default value is `32'h9502F90`.
Clock and reset ports		
reset	Input	Synchronous reset.
clk	Input	156.25 MHz clock.
Transmit ports on the AXI4-Stream interface		
tx_axis_tdata[63:0]	Input	Data to be transmitted to the 10-Gigabit Ethernet MAC IP core.
tx_axis_tkeep[7:0]		The transmit keep signal is used to determine which data bytes are valid on tx_axis_tdata during a given beat (this signal is valid only if tx_axis_tvalid and tx_axis_tready are both asserted). Bit 0 corresponds to the least significant byte on tx_axis_tdata and bit 7 corresponds to the most significant byte. When tx_axis_tlast is not asserted, the only valid value is `0xFF`. When tx_axis_tlast is asserted, valid values are `0x01` to `0xFF`.
tx_axis_tlast	Input	End of frame indicator on transmit packets. Valid only along with assertion of tx_axis_tvalid.
tx_axis_tvalid	Input	Source ready to provide transmit data. Indicates that the generator is presenting valid data on tx_axis_tdata.
tx_axis_tuser	Input	If asserted indicates an underrun frame. This is tied to `1'b0`.
tx_axis_tready	Input	Destination ready for transmit. Indicates that the 10-Gigabit Ethernet MAC IP core is ready to accept data on tx_axis_tdata. The simultaneous assertion of tx_axis_tvalid and tx_axis_tready marks the successful transfer of one data beat on tx_axis_tdata.
Receive Ports on the AXI4-Stream Interface		

Table 5-4: **Ethernet Performance Monitor Parameters and Ports *(Cont'd)***

Port/Parameter Name	Type	Description
rx_axis_tdata[63:0]	Input	Data received by the 10-Gigabit Ethernet MAC IP core.
rx_axis_tkeep[7:0]		The receive keep signal is used to determine which data bytes are valid on rx_axis_tdata during a given beat (this signal is valid only if tx_axis_tvalid and tx_axis_tready are both asserted).
rx_axis_tlast	Input	End of frame indicator on received packets. Valid only along with asser-tion of rx_axis_tvalid.
rx_axis_tvalid	Input	Source ready to provide data. Indicates that the MAC is presenting valid data on rx_axis_tdata.
rx_axis_tuser	Input	If asserted indicates a good packet is received.
rx_axis_tready	Output	Destination ready for receive. Indicates that the loopback is ready to accept data on rx_axis_tdata. The simultaneous assertion of rx_axis_tvalid and rx_axis_tready marks the successful transfer of one data beat on rx_axis_tdata. The 10-Gigabit Ethernet MAC IP core doesn't look at this signal and sends received data whenever available.
Performance Statistics Ports		
tx_byte_count	Output	Number of bytes transmitted in one second.
tx_pkt_count	Output	Number of packets transmitted in one second.
rx_byte_count	Output	Number of bytes received in one second.
rx_pkt_count	Output	Number of packets received in one second.

User Control and Status Registers

The user selections made in the Ethernet controller application are passed to the Traffic Generator and Monitor using this block. An AXI4-Lite interface is required for the MicroBlaze processor subsystem to execute reads (status) and writes (control) to this block. The AXI4-Lite to the AXI4-Lite IP Interface IP core (IPIF) [Ref 9] is instantiated in the design to read and write to a register map file.

Providing an AXI4-Lite slave interface provides the flexibility of using this module in other designs. To reuse this block, the control and status signals into the register map must be changed. Appendix C, User-Space Registers describes the registers implemented in the Traffic Generator and Monitor block. Figure 5-3 shows the user register interface.

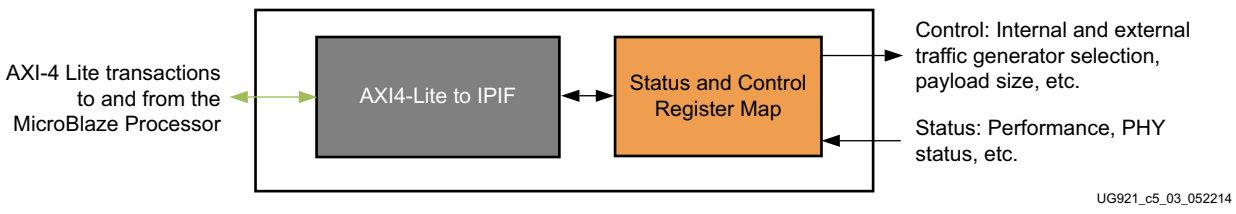


Figure 5-3: **User Register Interface**

SYSMON for Power and Temperature Monitoring

All UltraScale™ devices contain a System Monitor (SYSMON). SYSMON is used for monitoring die temperature and voltage and current on different power supply rails which are used to calculate system power.

For more information about SYSMON see *UltraScale Architecture System Monitor* (UG580) [Ref 8].

The application driver running on the MicroBlaze processor subsystem sets up SYSMON to read the temperature, voltage, and current data periodically. The System Management Wizard for SYSMON is configured with AXI4-Lite interface. This interface is used for communication with the MicroBlaze controller.

For more information on the System Management Wizard see *LogiCORE IP System Management Wizard Product Guide* (PG185) [Ref 11].

Figure 5-4 shows the power and temperature monitoring through SYSMON.

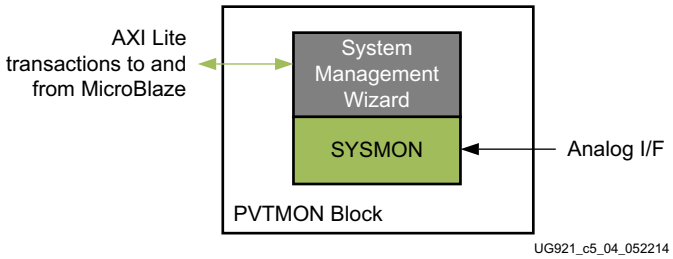


Figure 5-4: **Power and Temperature Monitor**

Control Plane Components

The Ethernet Controller application running on the control computer sends control information and receives status to and from different components of the 10GBASE-R TRD using the MicroBlaze processor subsystem.

MicroBlaze Processor Subsystem and AXI Interconnect

The IP cores required to support the MicroBlaze processor and create a subsystem are:

- MicroBlaze local memory
- Processor system reset
- MicroBlaze debug module
- AXI Interrupt controller

In Vivado IPI, adding the MicroBlaze IP core and running the connection automation creates the MicroBlaze processor system shown in Figure 5-5.

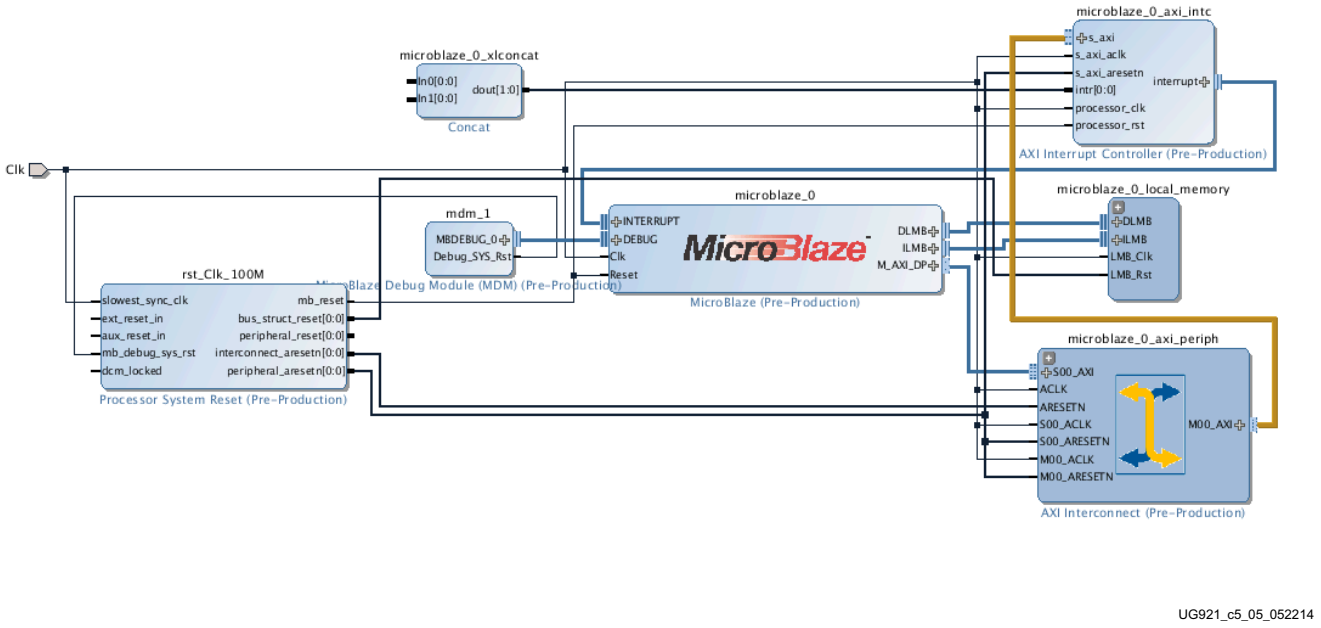


Figure 5-5: **IPI-Generated MicroBlaze Processor System**

The AXI interconnect IP allows the MicroBlaze processor subsystem to communicate with the AXI Interrupt controller IP using the AXI4 memory mapped interface.

The 10GBASE-R TRD AXI interconnect IP is reconfigured to have seven master ports instead of one. The seven master ports connect to seven AXI slaves:

- AXI interrupt controller
- AXI UART Lite which communicates with the Ethernet Controller application running on the control computer
- System Management Wizard for SYSMON
- Channel 0, the 10-Gigabit Ethernet MAC IP core
- The Traffic Generator and Monitor connected to channel 0
- Channel 1, the 10-Gigabit Ethernet MAC IP core
- The Traffic Generator and Monitor connected to channel 1

The AXI interconnect enables communication between the MicroBlaze processor subsystem (Master) and seven Peripherals (Slaves).

The address range assigned to each peripheral is shown in Figure 5-6. The application driver running on the MicroBlaze processor subsystem will use these addresses to map read/write transactions from the Ethernet Controller application to the AXI UART Lite to the MicroBlaze processor subsystem to other peripherals and back.

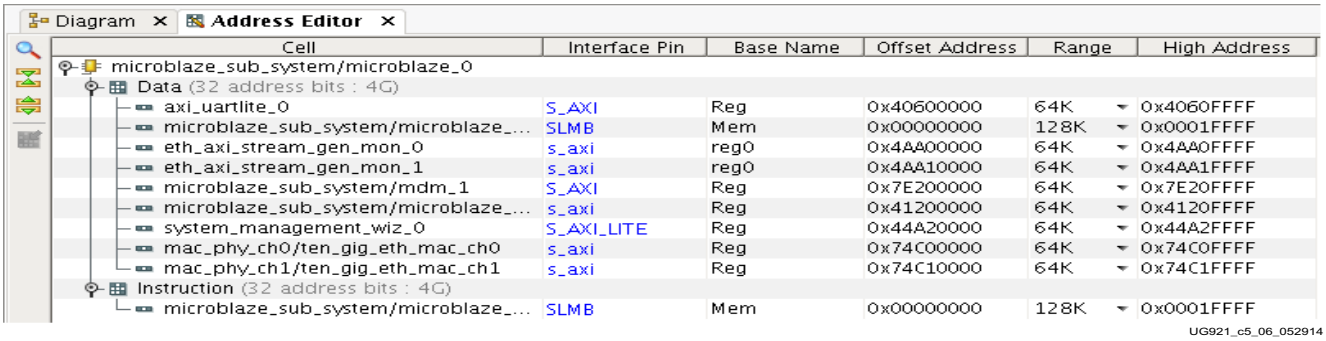


Figure 5-6: **Peripheral Address Map**

For more details on the MicroBlaze processor core see [Ref 13].

For more details on the AXI Interconnect see [Ref 14].

AXI UART Lite

The AXI UART Lite IP core provides the controller interface for asynchronous serial data transfer. The Ethernet Controller application running on the control computer communicates with this serial interface.

The AXI UART Lite IP core also connects to the MicroBlaze processor subsystem through the AXI interface and passes information to and from the Ethernet Controller application to the different components of the design.

For more details on the AXI UART Lite see *LogiCORE IP AXI UART Lite Product Guide* (PG142) [Ref 10] and the *AXI UART Lite webpage* [Ref 15].

Clocking and Reset

The 10-Gigabit Ethernet PCS/PMA core requires a 156.25 MHz differential reference clock for transceivers. The shared logic (clocking and reset logic) within the channel 0 10-Gigabit Ethernet PCS/PMA IP core produces a single ended 156.25 MHz clock. This clock is used for all of the blocks in the design including the MicroBlaze processor subsystem and SYSMON. Figure 5-7 shows the clock distribution.

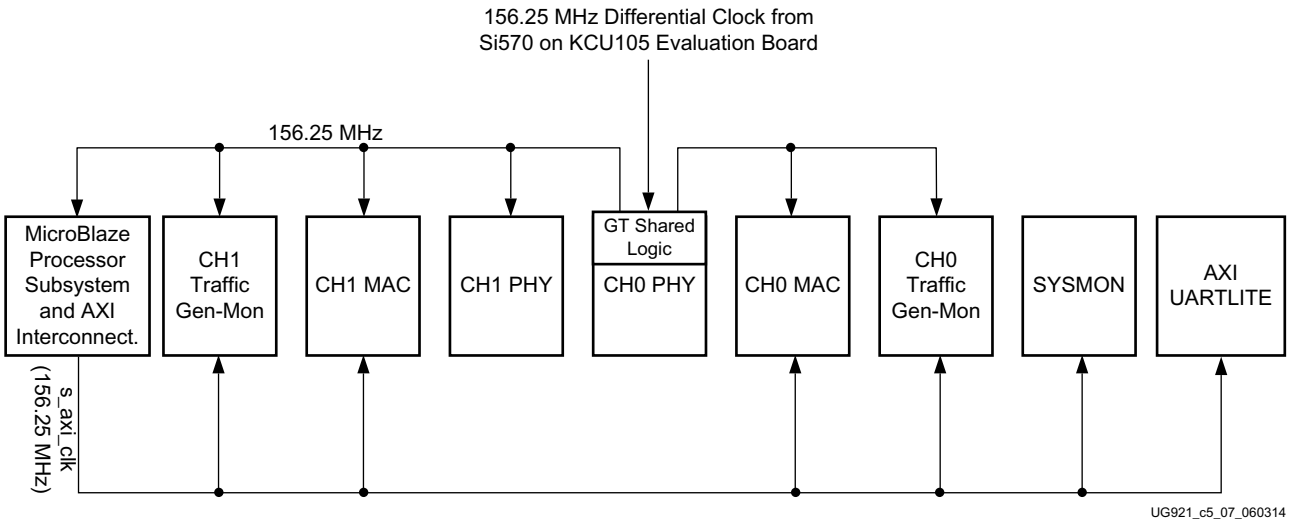


Figure 5-7: **Clock Distribution**

An external reset (a debounced pushbutton switch) drives the 10-Gigabit Ethernet PCS/PMA IP cores, the 10-Gigabit Ethernet MAC IP cores and the Processor Subsystem Reset IP core in the MicroBlaze processor System after being debounced.

The Processor System Reset Module provides resets for the MicroBlaze processor subsystem components and resets to the AXI Interconnect and peripherals (AXI4-Lite interfaces on AXI UART Lite, SYSMON, Traffic Generator and Monitor, and 10-Gigabit Ethernet MAC IP). Figure 5-8 shows the reset connections.

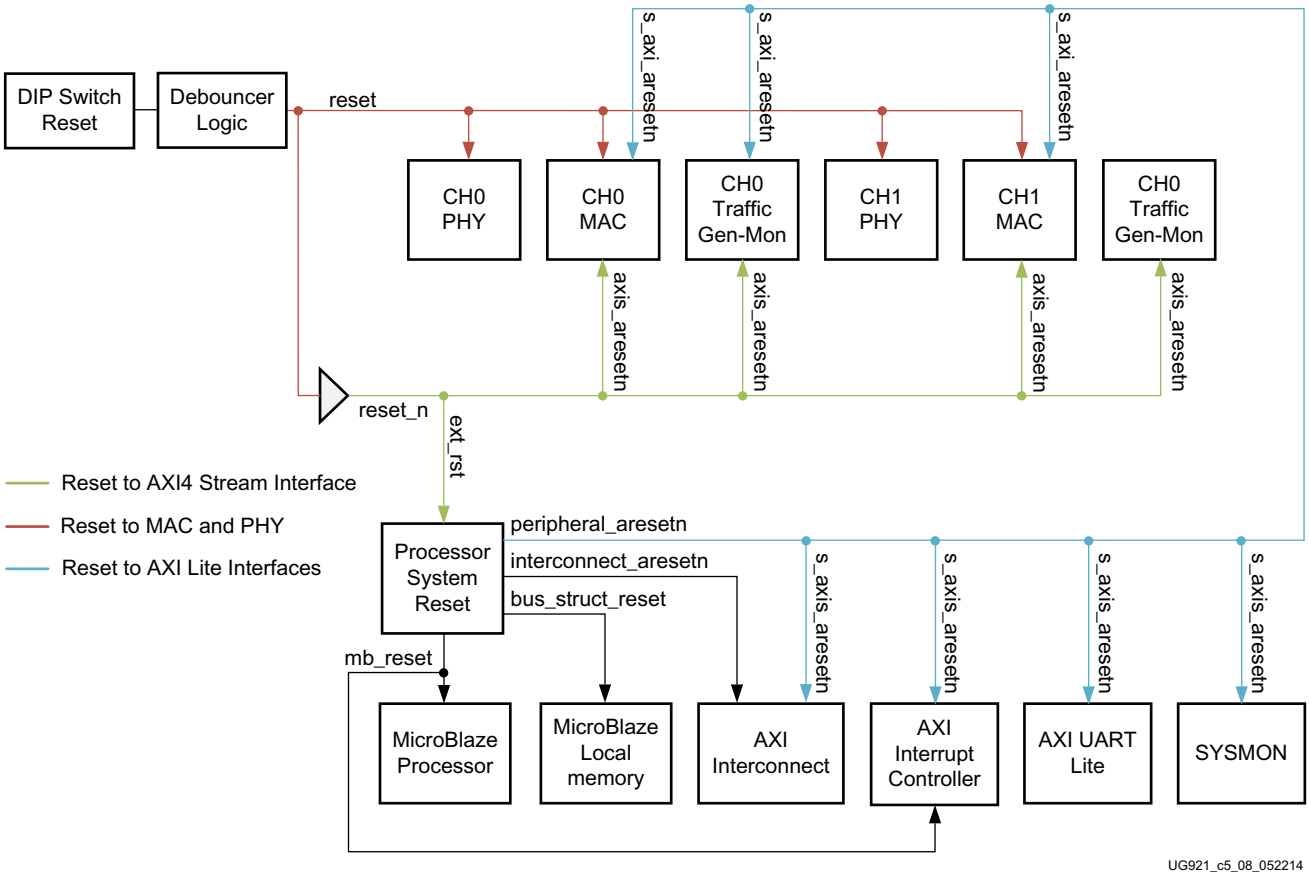


Figure 5-8: **Resets**

For details on the Processor System Reset Module see the *Processor System Reset Module webpage* [Ref 16].

Software

There are two major software components to monitor and control the Ethernet Reference Design System:

- GUI/Client Application
- MicroBlaze Processor Server Application

Figure 5-9 shows these components.

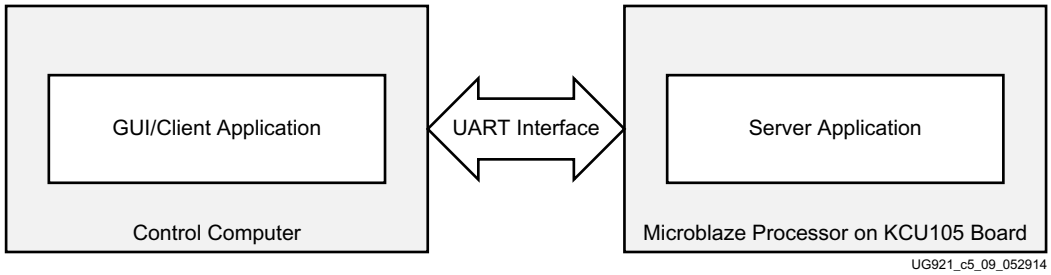


Figure 5-9: **Software Components**

GUI/Client Application

A Java-based GUI/Client application running on the control computer communicates with the MicroBlaze Processor Server Application through a UART interface to control test parameters, collect statistics and display current status of the design.

The GUI displays the following information:

- Current mode of operation
- Payload size
- Throughput numbers and graphs when a test is executing
- 10-Gigabit Ethernet MAC IP statistics
- Power consumption and temperature for the FPGA device
- Block diagram of the design

Figure 5-10 shows the GUI.

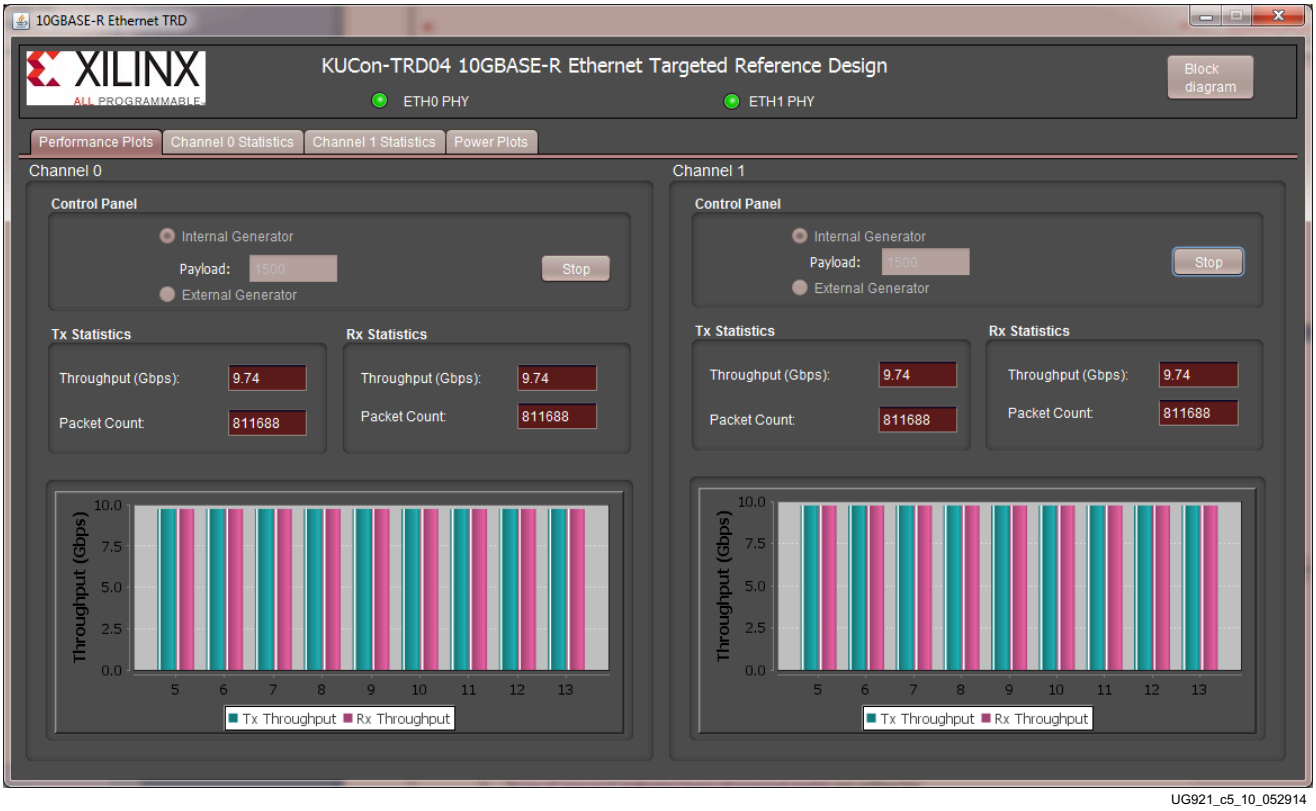


Figure 5-10: **GUI (Ethernet Controller Application)**

The GUI provides control of:

- Test mode: Select use of the internal traffic generator or an external generator to generate Ethernet traffic.
- Payload size: Specify the size of packets when running in internal generator mode.

The GUI interacts with the KCU105 board through the UART COM port exposed by the Silicon Labs UART driver. All transmitted and received data adheres to a custom command model followed by the client and server.

Command Format

The command format used by the GUI for reading and writing 32-bit values to the registers in the design is described in this section.

Read Command

```
R <type of request> <output type> <Command number, denoted as a
4-character hexadecimal numeric string, AAAA>
```

- `R` denotes a Read command
- `type of request` indicates the type of request and it can either be:
  - `s` which specifies a single register read request, or
  - `b` which specifies a bulk read request

Use a single register read for debugging purposes or for reading registers which are not monitored by the GUI.

Use a bulk read command to make the GUI constantly poll all test parameters and statistics. This eliminates the need to send multiple commands for each individual value. A bulk read command reads the registers in a predefined order.

- `output type` indicates output type and it can be:
  - `h` specifies the value sent out by the server will be the actual data width i.e., 4 bytes
  - `a` specifies the value sent out by the server will be an 8-byte hexadecimal string
- `AAAA` denotes four character hexadecimal numeric string representation of the command number which will eventually be mapped to the actual register offset in the server application. Using different command numbers instead of the actual register values allows the GUI to remain constant in spite of changes in the hardware design or the application.

Example Read command:

`R s a 0001`: Read a single register corresponding to command number `0001` and the reported value from the server should be in 8-byte hexadecimal format.

Read command response:

1. The command number is read by the server and the appropriate register is identified, its value is read, and is stored in a 4-byte data value.
2. Based on the output type specified in the read command, the data is either directly transmitted to the client by the server, or it is converted to an 8-character string denoting an 8-character hexadecimal number and then sent out to the client on the control computer side.

Currently the application only supports output type of 8-character hexadecimal string.

An example Read command output response : `F000000F`.

Write Command

```
W <Command Number denoted as a 4-character hexadecimal numeric string
AAAA> < Data, represented as an 8-character hexadecimal numeric
string DDDDDDDD>
```

- `W` denotes a Write command
- `AAAA` denotes a 4-character hexadecimal string representation of the command number which will eventually be mapped to the actual register offset in the server application.
- `DDDDDDDD` denotes an 8-character hexadecimal string.

Example Write command:

- `W 0001 007D0002`: Write a value of `007D0002` to the register corresponding to command value 0001.

Write command is always targeted towards a single register.

The mapping of command numbers to the corresponding register values is the same for both Read and Write commands, i.e., if a write command with a certain command number is used to update a register value, the same command number can be used with a read command to retrieve the value.

MicroBlaze Processor Server Application

MicroBlaze Processor Server application running on the FPGA takes care of interpreting the read and write commands sent from the client application, and acts accordingly. The software layers are as shown in Figure 5-11.

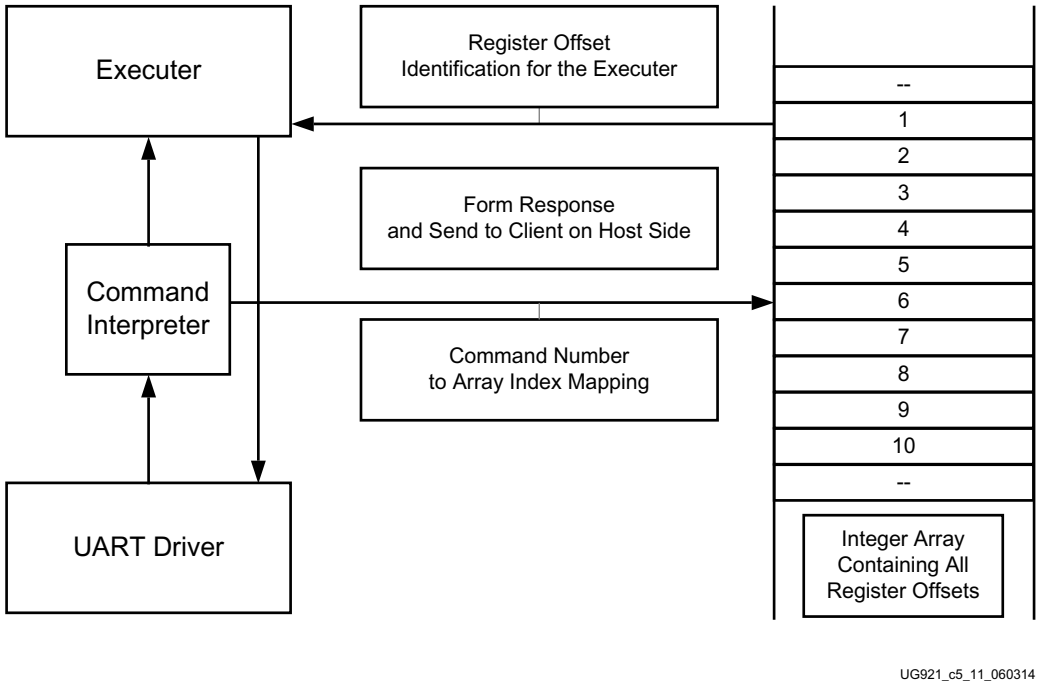


Figure 5-11: **Software Layers in the MicroBlaze Processor Server application**

When the hardware platform is exported to Software Development Kit (SDK) as described in step 2, page 32:

- The drivers for standard peripherals (like UART, Interrupt controller, etc.) that are connected to the MicroBlaze processor are already available in the SDK.
- The base address of each peripheral assigned in the hardware platform as shown in Figure 5-6 and is also accessible by the SDK.

The register offsets required by the server application to access relevant registers (TRD configuration registers, performance registers, MAC Statistics, etc.) are available in the IP core product guides being used in this design [Ref 6], and the user space registers as described in *Appendix C, User-Space Registers*.

The UART driver is responsible for transferring and receiving data from the UART interface. The server application uses the Application Programming Interface (API) provided by the UART driver and communicates with the client application running on the control computer. The main task of the server is to read/write values from/to the registers specified in the READ/WRITE commands issued by the GUI.

The various steps undertaken by the server in servicing requests from the client are:

1. Register offsets to access registers are put in an integer array during the initialization of MicroBlaze processor server application. The value placed in the integer array is:
    - Base address for the component + register offset
2. The order in which these register offset values are placed is made aware to the client application on the host side.
3. Each register is identified using a unique command number. This command number is same for both read and write commands.
4. When a client initiates a Read/Write command, the data is obtained by the server application through the UART driver.
5. The Command Interpreter part of the server application then interprets the data obtained from UART.
6. The Command Interpreter identifies the register offset value in the integer array, with respect to the command number associated with the request.
7. The Command Interpreter functionality is different for Single Read and Bulk Read commands.
8. For a single read request, command number shall give the index of the integer array where the referenced register offset is placed.

Example Single Read Command:

```
R s a 0001
```

Here the command number is translated to 1 and the offset value of the register being requested is placed in the zero index of the array.

9. For a bulk read request the command number represents a set of registers with a clearly defined start and end index positions in the array.

Example Bulk Read Command:

```
R b a 0001
```

Here the command number 1 represents a set of register offsets in the array, 2 to 17. The values placed at these array indices are the Ethernet performance registers for Channel 0 and Channel 1.

10. The Executor part of the server application then initiates either AXI Read/Write request to the register offset values identified.
11. The Executer then sends an appropriate response to the client on control computer through the UART driver.

    Note: The bulk Read commands from GUI are serviced by the Server application by going through all the registers associated with the command in a predefined order and transferring the data with a ' ' (white space) as delimiter between each register's value. This mode greatly reduces the number of commands the GUI has to send to obtain the same amount of information.

The source code for the MicroBlaze application is available under the `software/source directory` (see Appendix A, Directory Structure). The command mapping is defined in CommandDesc.h in the source directory

Directory Structure

The directory structures for the 10GBASE-R TRD is shown in Figure A-1.

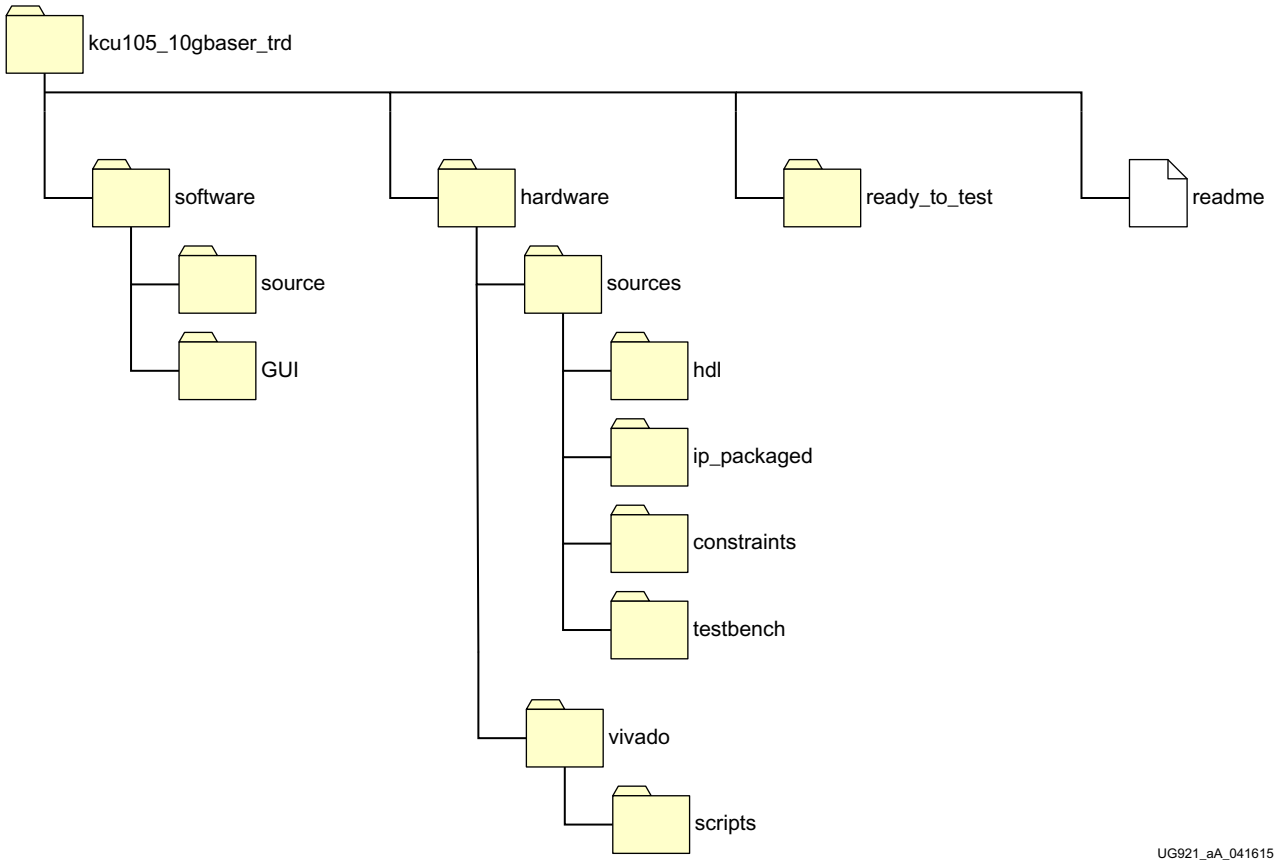


Figure A-1: **Reference Design Directory Structure**

Directory Content Summary

The files and folders contained in the 10GBASE-R TRD are described here. The top-level folder is `kcu105_10gbaser_trd`.

`hardware` Folder

The `hardware` folder contains all the required sources needed to generate a bitstream.

sources Folder

The `sources` folder contains subfolders that contain HDL files, custom IP that is packaged, constraint files, and testbench files.

vivado Folder

The `vivado` folder contains files to create a Vivado® Design Suite project and outputs of vivado runs.

scripts Folder

The `scripts` folder contains Tcl scripts to create a Vivado project.

runs Folder

The `runs` folder is created when the Tcl script file is sourced. The runs folder contains the output of simulation, synthesis and implementation processes.

ready_to_test Folder

The `ready_to_test` folder contains the bit file to program the KCU105 evaluation board.

software Folder

The `software` folder contains the software design deliverables.

source Folder

The `source` folder contains the source code used for creating the ELF file application that runs on the MicroBlaze™ processor and communicates with the Ethernet Controller application.

GUI Folder

The `GUI` folder contains the EXE file used to install the Ethernet Controller application.

readme File

The readme file is a TXT file that describes the 10GBASE-R TRD and includes revision history information.

Performance Estimates

The 10-Gigabit Ethernet MAC IP core operates at a clock rate of 156.25 MHz using a 64-bit data-path width (64 x 156.25 x 10^6 = 10 Gb/s).

For the XGMII, the minimum required interframe gap is 12 bytes. Header overhead consists of a preamble (7 bytes) + Start of frame delimiter (1 byte) + MAC destination address (6 bytes) + MAC source address (6 bytes) + Length/Type field (2 bytes) + FCS (4 bytes). This gives a total overhead of 38 bytes per Ethernet packet.

Table B-1 shows the effective throughput and percentage of maximum bandwidth used for four different payload sizes.

Table B-1: **Effective Throughput as a Function of Payload**

Ethernet Payload Size, Bytes	Percentage of Bandwidth = Payload size/Packet size * 100	Effective Throughput, Gb/s
64	64/(38 + 64) = 62.7%	6.27
512	512/(38 + 512) = 93.1%	9.31
1024	1024/(38 + 1024) = 96.3%	9.63
1500	1500/(38 + 1500) = 97.5%	9.75

User-Space Registers

User-space registers are user defined registers implemented in the Traffic Generator and Monitor block shown in Figure 5-1, page 42. These registers can be accessed by the MicroBlaze processor subsystem via the AXI4-Lite interface.

Table C-1 through Table C-14 describe the custom registers implemented in the 10GBASE-R TRD. All registers are 32 bits wide. Register bit positions are to be read from bit 31 to bit 0 from left to right. All bits that are undefined in this section are reserved and will return zero when read. Address holes will also return a value of zero when read.

Each peripheral connected to the MicroBlaze processor subsystem is assigned an offset address which is the base address for that peripheral. Figure 5-6, page 53 shows the addresses assigned to the Traffic Generator and Monitor blocks (eth_axi_stream_gen_mon_0 and eth_axi_stream_gen_mon_1). The Traffic Generator and Monitor base addresses are:

- Traffic Generator and Monitor channel 0 is `0x4AA0_0000`
- Traffic Generator and Monitor channel 1 is `0x4AA1_0000`

Control and Status Registers

Traffic Generator—Monitor Channel 0

Table C-1: **Design Version Register (0x4AA0_0000)**

Bit Position	Mode	Default Value	Description
3:0	Read Only	`4'h1`	Ethernet reference design1.
15:4		`12'h141`	Software version: Indicates the Vivado® Design Suite version used when developing this reference design. For example, Vivado Design Suite 2014.1 is indicated by 141.
31:16		`16'h0105`	Target Board: KCU105 board.

Table C-2: **Ethernet Performance Monitor, Transmit Payload Byte Count Register (0x4AA0_0004)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Transmit payload byte count. This field contains the interface utilization count for active beats (tx_axis_tready = 1 and tx_axis_tvalid = 1) on channel 0 10G Ethernet MAC AXI4-Stream interface for transmit.

Table C-3: **Ethernet Performance Monitor, Transmit Packet Count Register (0x4AA0_0008)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Transmit packet count. This field contains the count for the event when there is an active beat on channel 0 10G Ethernet MAC AXI4-Stream interface and end of packet (tx_axis_tlast) is asserted for transmit.

Table C-4: **Ethernet Performance Monitor, Received Payload Byte Count Register (0x4AA0_000C)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Receive payload byte count. This field contains the interface utilization count for active beats (rx_axis_tready = 1 and rx_axis_tvalid = 1) on channel 1 10G Ethernet MAC AXI4-Stream interface for receive.

Table C-5: **Ethernet Performance Monitor, Received Packet Count Register (0x4AA0_0010)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Receive packet count. This field contains the count for the event when there is an active beat on channel 1 10G Ethernet MAC AXI4-Stream interface and end of packet (rx_axis_tlast) is asserted for receive.

Table C-6: **Traffic Generator Configuration Register (0x4AA0_0014)**

Bit Position	Mode	Default Value	Description
0	Read or Write	0	Enable loopback if external generator is selected.
1		0	Enable generator if internal generator is selected.
31:16		d'125	Ethernet frame data payload size. Allowed values (46 bytes to 1,500 Bytes).

Table C-7: **Loopback module and PHY status Register (0x4AA0_00x18)**

Bit Position	Mode	Default Value	Description
0	Read Only	0	PHY is up.
31		0	Packets dropped in loopback mode when external traffic generator is selected.

Traffic Generator—Monitor Channel 1

Table C-8: **Design Version Register (0x4AA1_0000)**

Bit Position	Mode	Default Value	Description
4:0	Read Only	4'h1	Ethernet reference design1.
14:5		12'h141	Design was developed in which Vivado version 141 = Vivado Design Suite 14.1
31:16		16'h0105	Target Board. KCU105 board.

Table C-9: **Ethernet Performance Monitor, Transmit Payload Byte Count Register (0x4AA1_0004)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Transmit payload byte count. This field contains the interface utilization count for active beats (tx_axis_tready = 1 and tx_axis_tvalid = 1) on channel 1 10G Ethernet MAC AXI4-Stream interface for transmit.

Table C-10: **Ethernet Performance Monitor, Transmit Packet Count Register (0x4AA1_0008)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Transmit packet count. This field contains the count for the event when there is an active beat on channel 1 10G Ethernet MAC AXI4-Stream interface and end of packet (tx_axis_tlast) is asserted for transmit.

Table C-11: **Ethernet Performance Monitor, Received Payload Byte Count Register (0x4AA1_000C)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Receive payload byte count. This field contains the interface utilization count for active beats (rx_axis_tready = 1 and rx_axis_tvalid = 1) on channel 0 - 10G Ethernet MAC AXI4-Stream interface for receive.

Table C-12: **Ethernet Performance Monitor, Received Packet Count Register (0x4AA1_0010)**

Bit Position	Mode	Default Value	Description
1:0	Read Only	00	Sample count. Increments once every second.
31:2		0	Receive packet count. This field contains the count for the event when there is an active beat on channel 0 10G Ethernet MAC AXI4-Stream interface and end of packet (rx_axis_tlast) is asserted for receive.

Table C-13: **Traffic Generator Configuration Register (0x4AA1_0014)**

Bit Position	Mode	Default Value	Description
0	Read or Write	0	Enable loopback if external generator is selected.
1		0	Enable generator if internal generator is selected.
31:16		d'125	Ethernet frame. Data payload size allowed values = 46 bytes to 1,500 Bytes.

Table C-14: **Loopback module and PHY status Register (0x4AA1_00x18)**

Bit Position	Mode	Default Value	Description
0	Read Only	0	PHY is up.
31		0	Packets dropped in loopback mode when external traffic generator is selected.

Testing with an External Traffic Generator

This appendix describes how to setup and test the 10GBASE-R TRD running on the KCU105 board using an Ixia NGY-NP4-01 10 Gigabit Application Network Processor Load Module (Ixia load module).

Requirements

Hardware

- KCU105 evaluation board with the Kintex® UltraScale™ XCKU040-2FFVA1156E FPGA
- Two USB cables, standard-A plug to micro-B plug
- Power Supply: 100 VAC-240 VAC input, 12 VDC 5.0A output
- Four SFP+ 10GBASE-SR/SW transceiver modules [Ref 18]
- Four Multimode fiber optic patch cables [Ref 19]
- Ixia XM2 portable chassis with NGY-NP4-01 10-Gigabit application network processor load module (Ixia load module) [Ref 17].

Computer

The control computer is required for running Vivado® Design Suite, configuring the FPGA, and running the Ethernet Controller application to control and monitor the reference design. It can be a laptop or desktop computer with Microsoft Windows 7 Operating system.

Targeted Reference Design ZIP file

The TRD ZIP file contains:

- Design source files
- Ethernet Controller application files
- Board design file (bitfile)
- Documentation

These files are required to demonstrate the 10GBASE-R TRD on the KCU105 evaluation board.

Software

- Vivado Design Suite version 2015.4
- USB UART drivers (Silicon Laboratories CP210x VCP drivers)
- Java version 1.7
- Ethernet Controller application (Included with the 10GBASE-R TRD)

Software installation instructions are described in Chapter 2, Setup.

- Ixia software packages installed on the IXIA XM2 portable chassis. Software CDs and installation instructions are provided by IXIA [Ref 17].

TIP: *The IXIA XM2 portable chassis with a load module has windows OS installed on it. Connect a mouse, keyboard and monitor to the chassis to run IxExplorer (The IXIA controller application)*

Program the KCU105 Evaluation Board

This section describes how to setup the KCU105 board with the Ixia load module and control computer and how to program the FPGA with the 10GBASE-R TRD BIT file.

1. Connect the KCU105 board to the Ixia load module, control computer, and power supply as shown in Figure D-1:
   a. Connect the power supply to the KCU105 board.
   b. Connect the USB cable having a micro-B plug to the micro-B receptacle located on U115 on the KCU105 board. Connect the standard-A plug to a USB receptacle on the control computer.
   c. Connect the USB cable having a micro-B plug to the micro-B receptacle located at J4 on the KCU105 board. Connect the standard-A plug to a USB receptacle on the control computer.
   d. Insert two SFP+ transceiver modules into the SFP cage on the KCU105 board.
   e. Insert two SFP+ transceiver modules into the port 01 and port 02 cages on the Ixia load module.
   f. Connect the four fiber optic patch cables between the two SFP+ transceiver modules on the KCU105 board and the port 01 and port 02 SFP+ transceiver modules on the Ixia load module transmit to receive and receive to transmit.

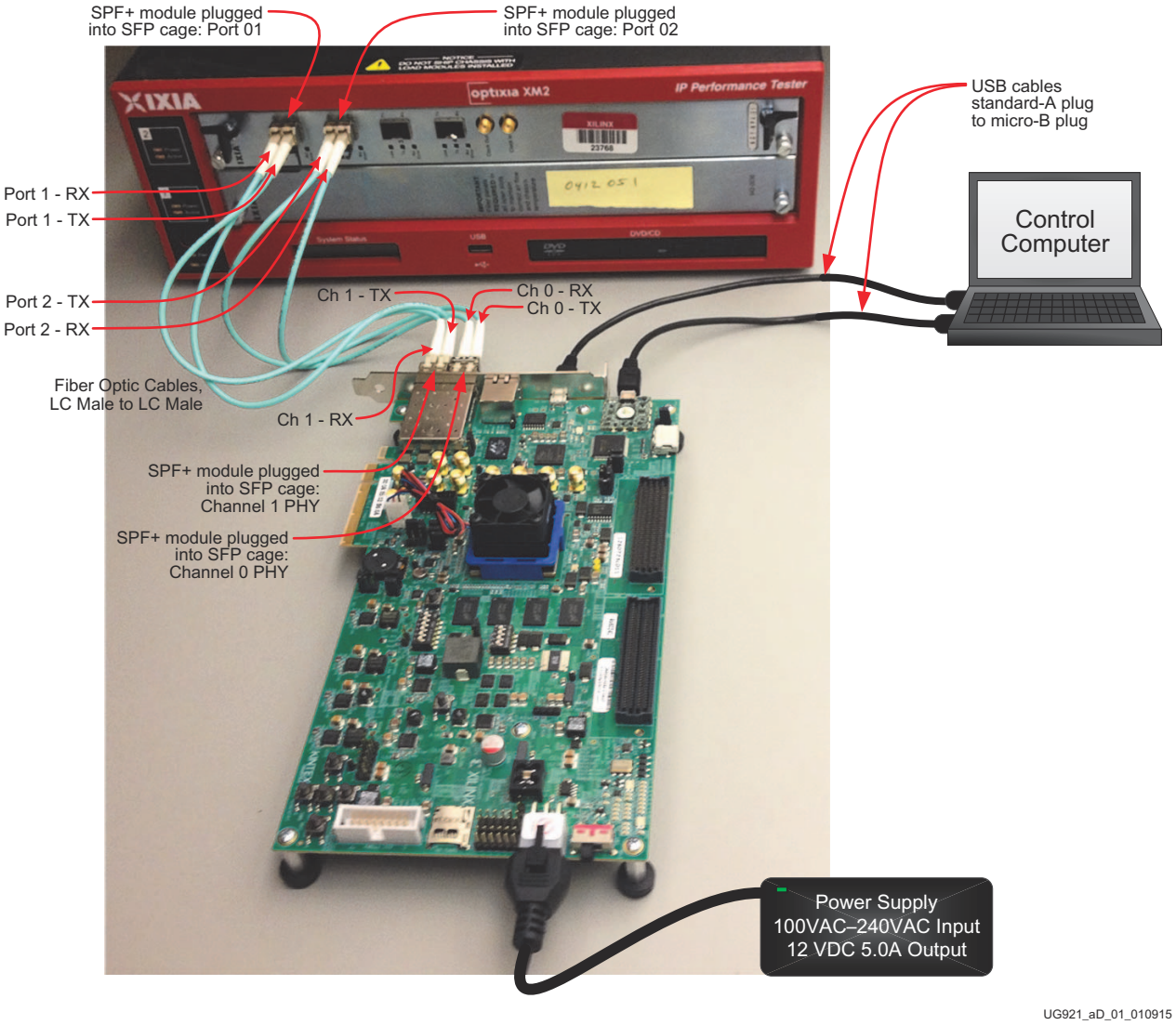


Figure D-1: **KCU105 Board and Ixia Module Connections**

IMPORTANT: *Port 1 of the Ixia load module is connected to connector SPF 1 on the KCU105 board which is Channel 1 in the reference design. Port 2 of the Ixia tester is connected to connector SPF 0 on the KCU105 board which is Channel 0 in the reference design.*

2. Power on the KCU105 board by placing switch SW1 to the ON position.

3. Launch the Vivado Integrated Design Environment (IDE) on the control computer:
   a. Select **Start > All Programs > Xilinx Design Tools > Vivado 2015.4 > Vivado 2015.4**.
   b. On the getting started page, click **Open Hardware Manager** (Figure D-2).

Figure D-2: **Vivado IDE Getting Started Page, Open Hardware Manager**

4. Open the connection wizard to initiate a connection to the KCU105 board:
   a. Click **Open a new hardware target** (Figure D-3).

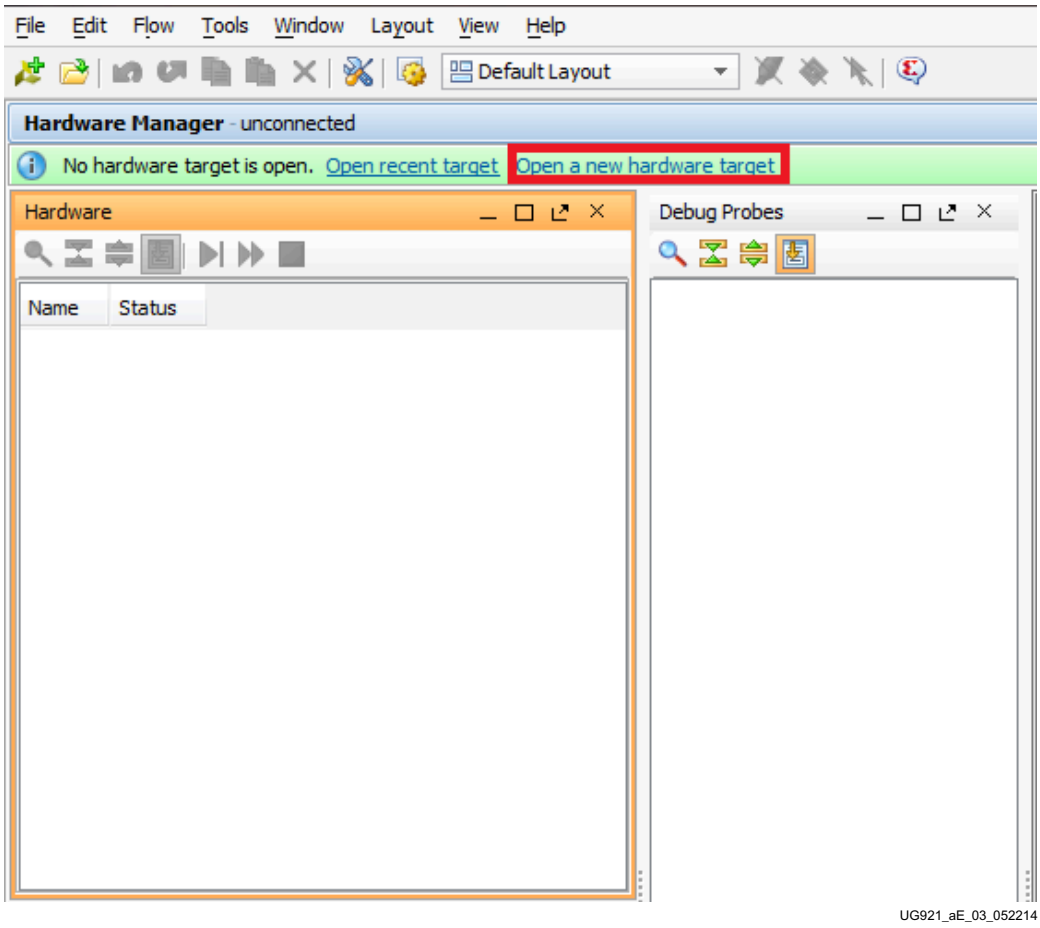


Figure D-3: **Using the User Assistance Bar to Open a Hardware Target**

   b. Configure the wizard to establish connection with the KCU105 board by selecting the default value on each wizard page. Click **Next > Next > Next > Next > Finish**.

c. In the hardware view, right-click on **xcku040** and click **program device...** (Figure D-4).

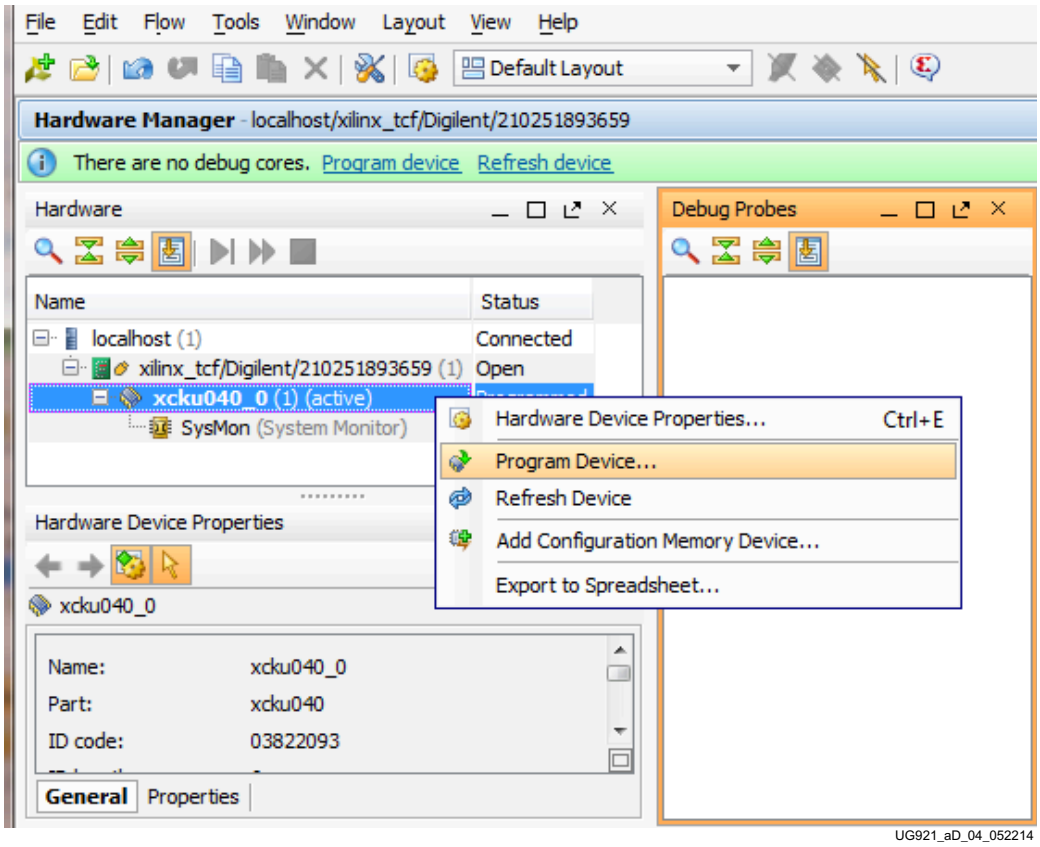


Figure D-4: **Select Device to Program**

d. In the **Bitstream file** field, enter:

```
<working_dir>/kcu105_10gbaser_trd/ready_to_test/kcu105_10gbase
r_download.bit
```

and click **Program** (Figure D-5).

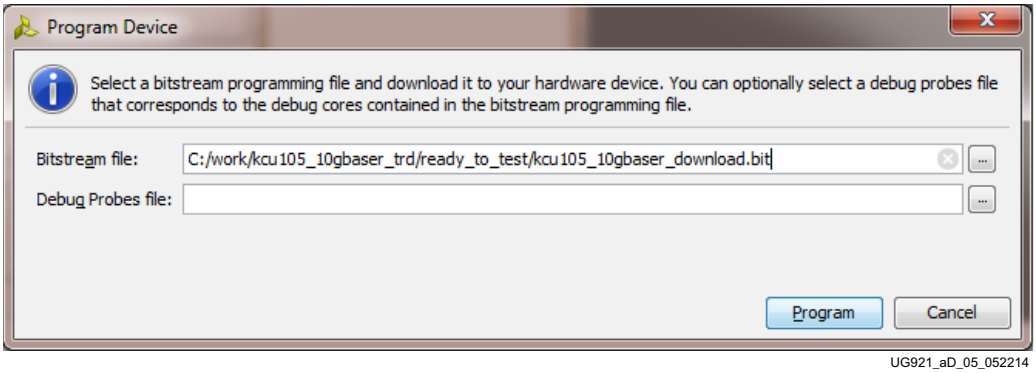


Figure D-5: **Program Device Window**

After completing these steps, continue on to Set Up the Ixia Load Module Parameters.

Set Up the Ixia Load Module Parameters

This section describes how to setup the Ixia Load Module Ethernet parameters using IxExplorer.

1. Power on the Ixia XM2 portable chassis.
2. Launch Ixia IxExplorer on the Ixia chassis. Click **Start > All Programs > Ixia > Ixia Application Selector > IxOS > IxOS 6.10.750.5 EA > Ixia IxExplorer** (Figure D-6).

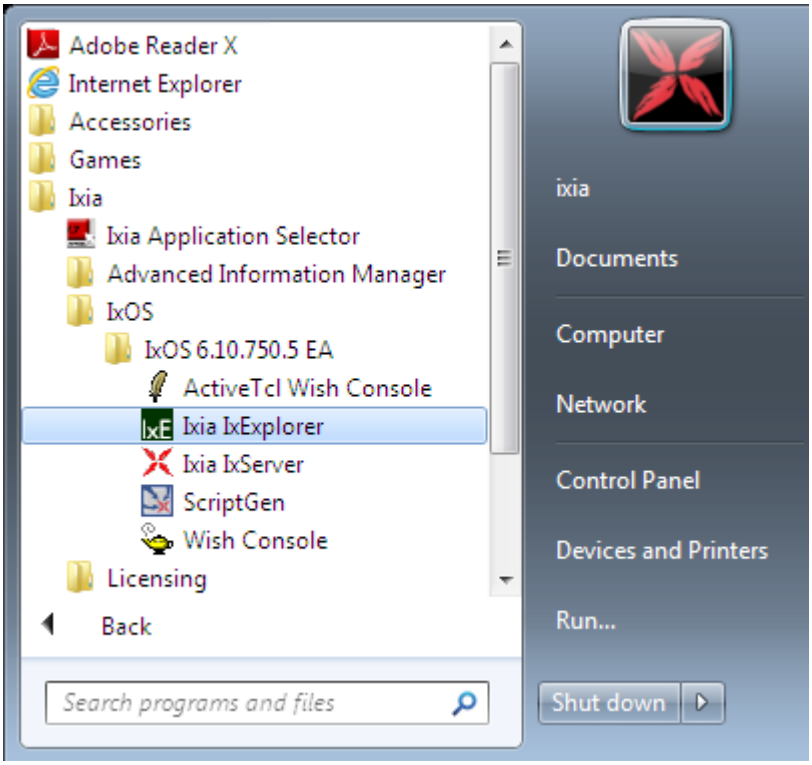


Figure D-6: **Open Ixia IxExplorer**

3. Confirm the link between the KCU105 board and the Ixia load module is up. The green icons shown Figure D-7 shows that Port 01 and Port 02 are connected and operating.

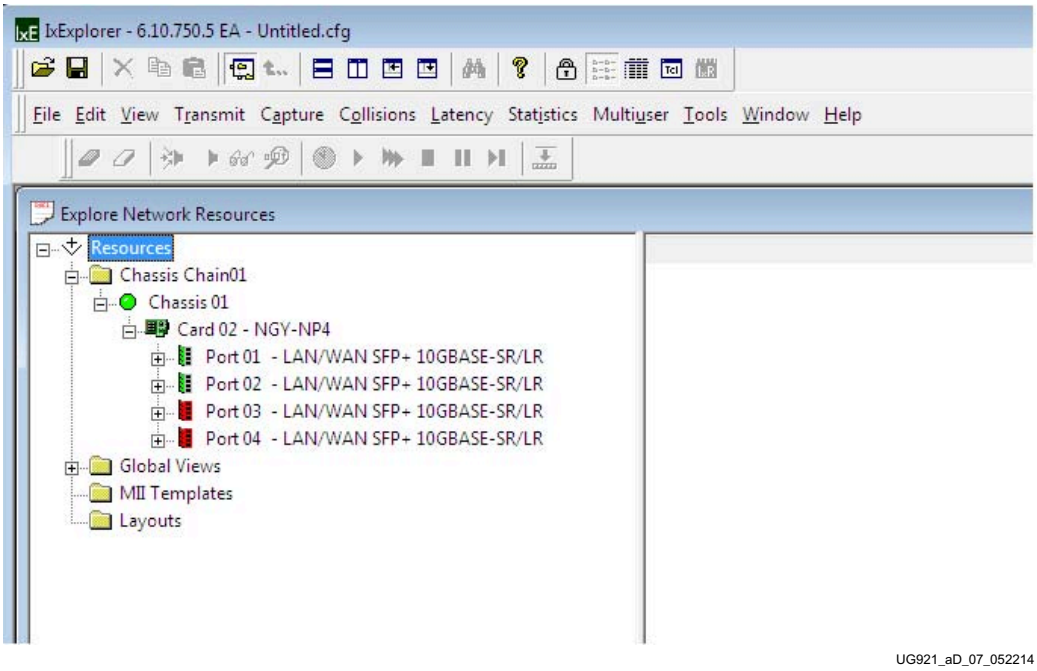


Figure D-7: **IxExplorer showing E-7 Link is Up Between the KCU105 board and Ixia Load Module**

4. Click the + next to Port 01 and Port 02 to expand both ports (Figure D-8).

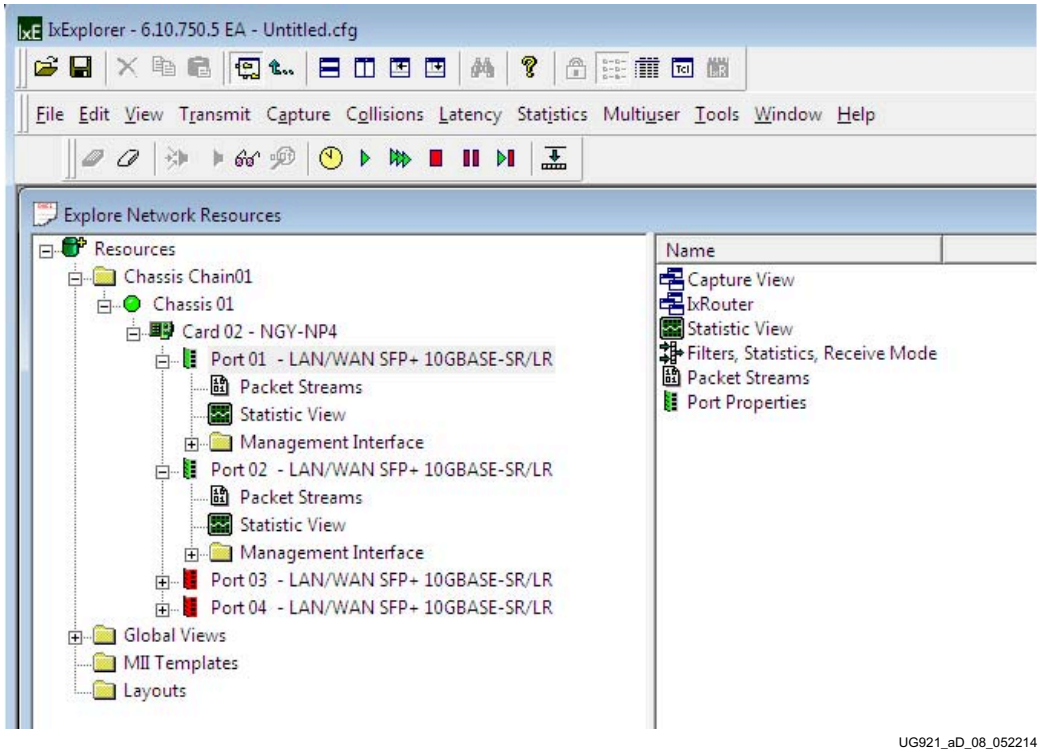


Figure D-8: **Expanded Port 01 and Port 02**

5. Click **Packet Streams** on Port 01 to display Ethernet packet parameters (Figure D-9).

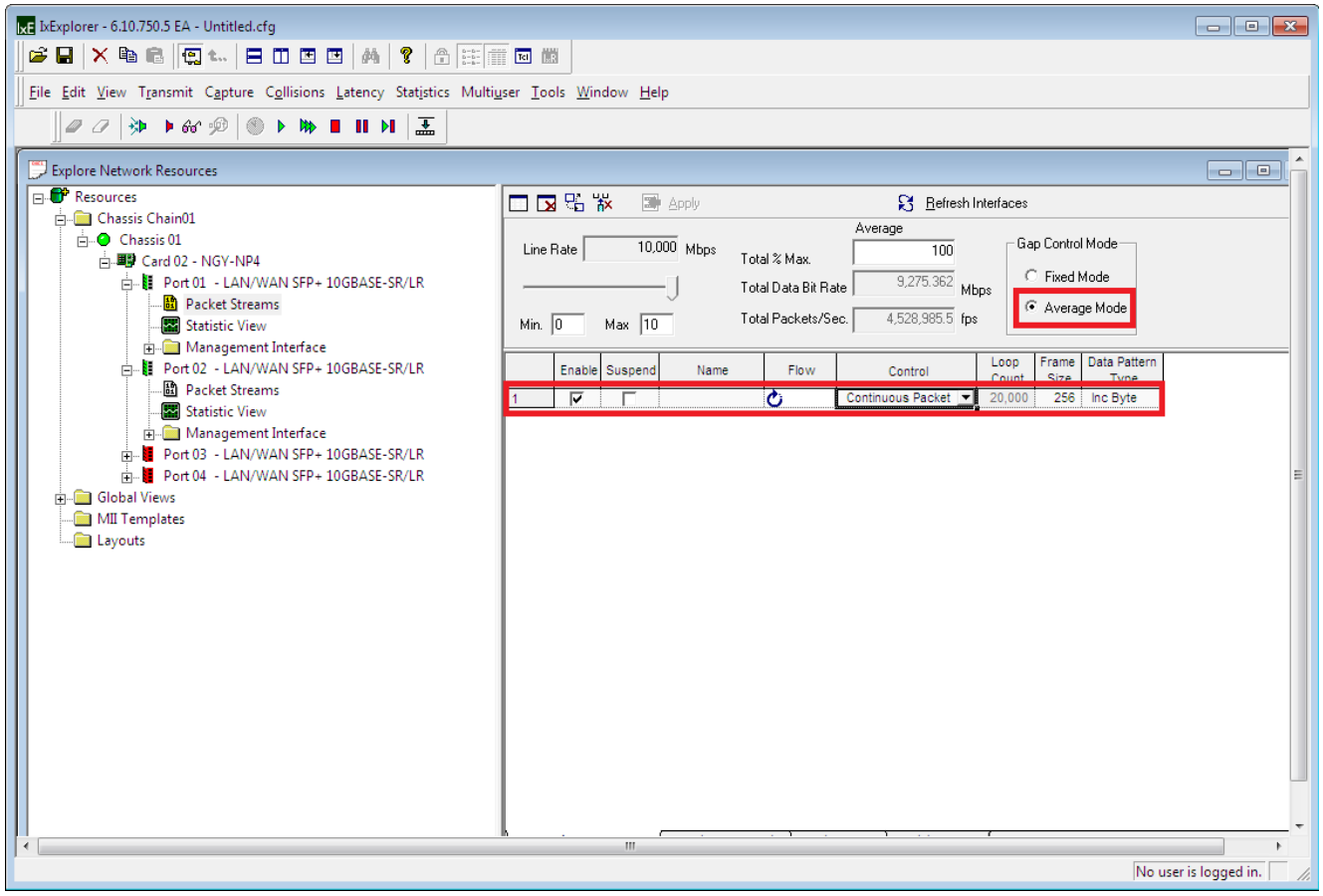


Figure D-9: **Open Packet Streams on Port 01**

6. Configure frame size, destination, source MAC addresses, and frame type:
   a. Click **line 1** (highlighted in Figure D-9) to open the stream properties window.
   b. Select the **DA/SA** tab (Figure D-10) and set the parameters as described here:
      - Frame Size: **1,518** Bytes (Includes Ethernet header and CRC)
      - Destination Address: **00 00 00 00 22 22**
      - Source Address: **00 00 00 44 44**
      - All other parameters should match values and selections shown in Figure D-10.

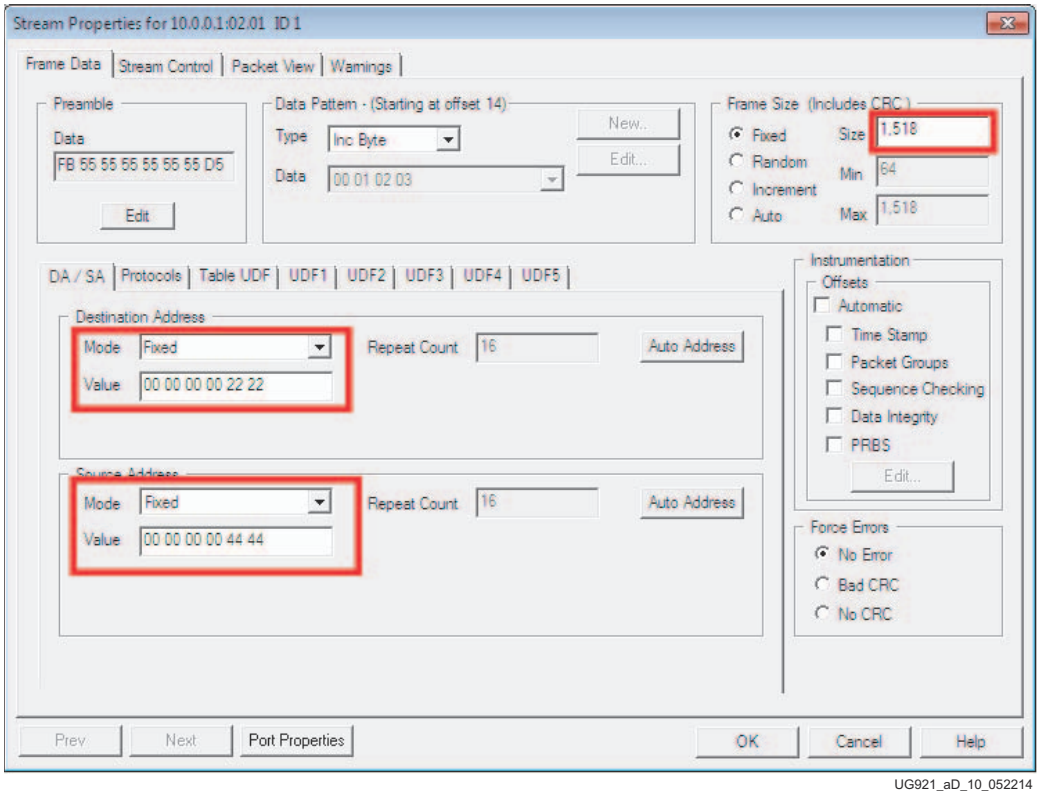


Figure D-10: **Set Frame Size and Addresses (DA/SA Tab)**

7. Select the **Protocols** tab (Figure D-11) and set the parameters as described here:
   - Select **802.3 Raw** frame type
   - All other parameters should match values and selections shown in Figure D-11.

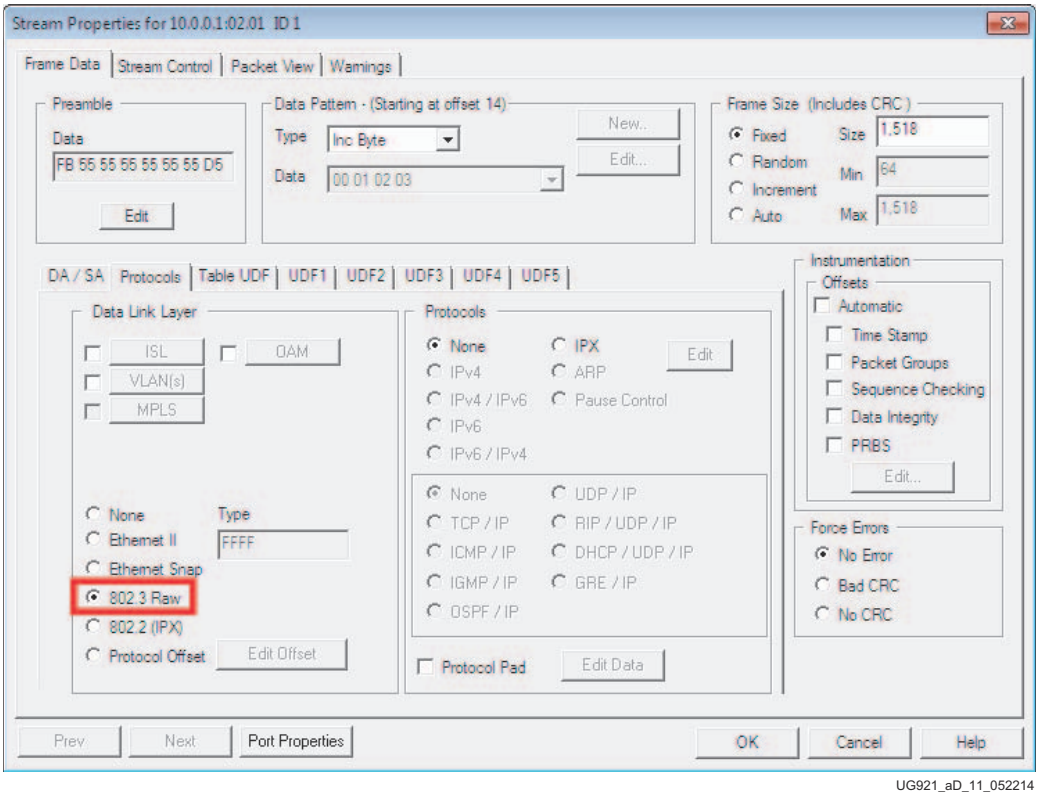


Figure D-11: **Set Frame Type to 802.3 (Protocols Tab)**

8. Select the **Stream** tab (Figure D-12) and set the parameters as described here:
   - Set the Inter Packet gap to **16** bytes
   - Set enforce minimum to **12** bytes
   - All other parameters should match values and selections shown in Figure D-12.
9. Click **OK**.

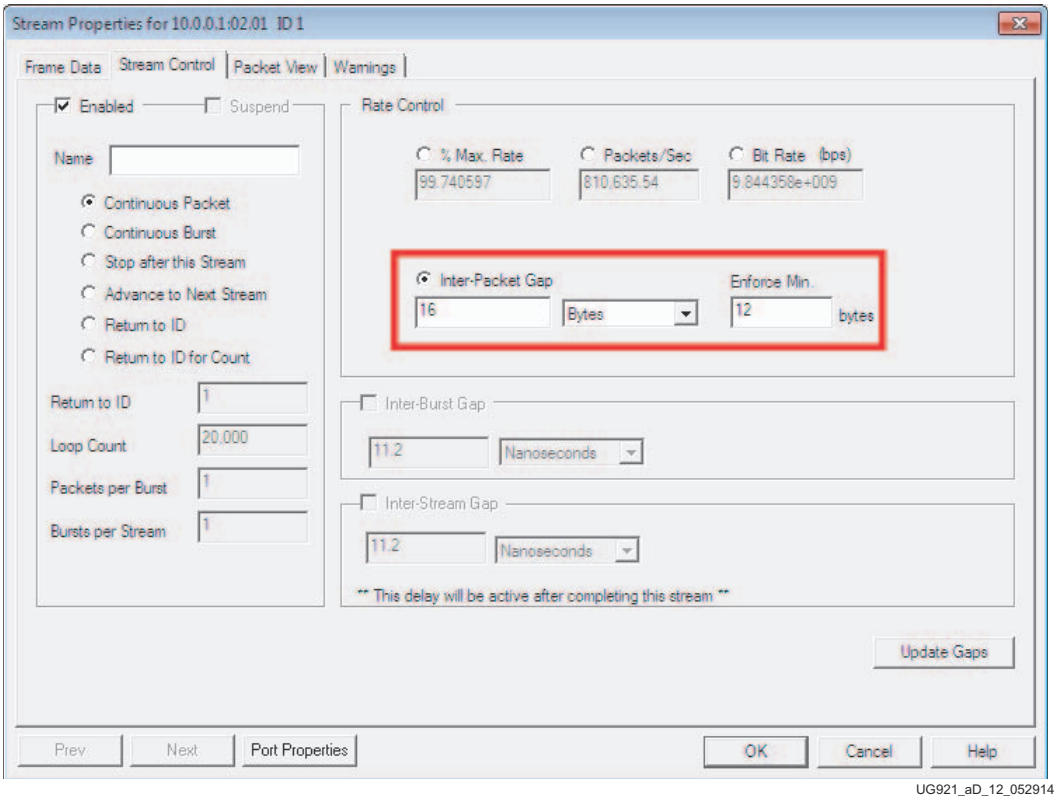


Figure D-12: **Stream Control Tab Settings**

IMPORTANT: *The inter-packet gap is set to 16 bytes to accommodate the parts per million (ppm) difference between the Ixia load module clock source and the KCU105 board clock source. In this lab setup, the Ixia load module clock is running faster than the KCU105 board clock. Because the TRD loops back the data coming from the Ixia load module through a FIFO, this FIFO will overflow at some point of time if the IPG is set to 12 on both the Ixia load module and the 10GBASE-R TRD. This is because the receive clock (from the Ixia load module) is faster than the transmit clock (from the KCU105 board). If the transmit data was not dependent on the receive data then the IPG could be set to 12 bytes.*

10. Repeat step 5 through step 7 for Port 02 but set the destination address to **00 00 00 00 11 11** and source address to **00 00 00 33 33**.

After completing these steps, continue on to Launch the Ethernet Controller Application.

Running the Design

Launch the Ethernet Controller Application

1. Launch the Ethernet Controller application on the control computer. Click **Start > All Programs > Xilinx > EthernetController** (Figure D-13).

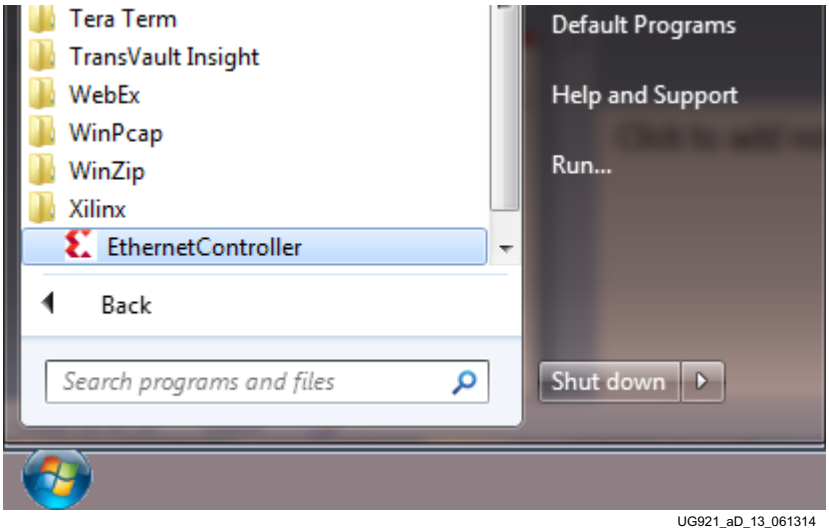


Figure D-13: **Ethernet Controller Application**

2. Select the COM port associated with the Silicon Labs CP210x USB to UART Bridge and click **Connect** (Figure D-14) to open the Ethernet Controller application for the 10GBASE-R TRD.

TIP: *The COM port associated with the Silicon Labs CP210x USB to UART Bridge can be identified using the Windows control panel. See step 3, page 11.*

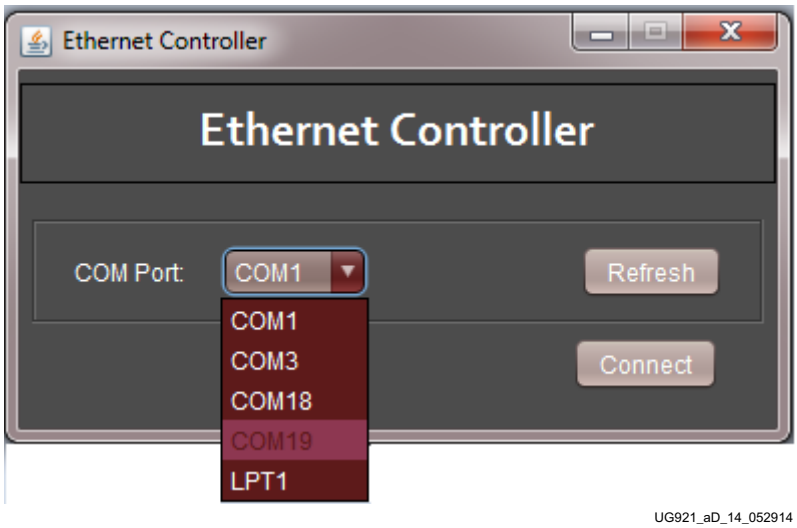


Figure D-14: **Select COM Port Associated with the USB to UART Bridge**

3. Ethernet channel 0 and channel 1 are up and ready when the ETH0 PHY and ETH1 PHY indicators are green. For both channels select **External Generator** and click **Start** (Figure D-15).

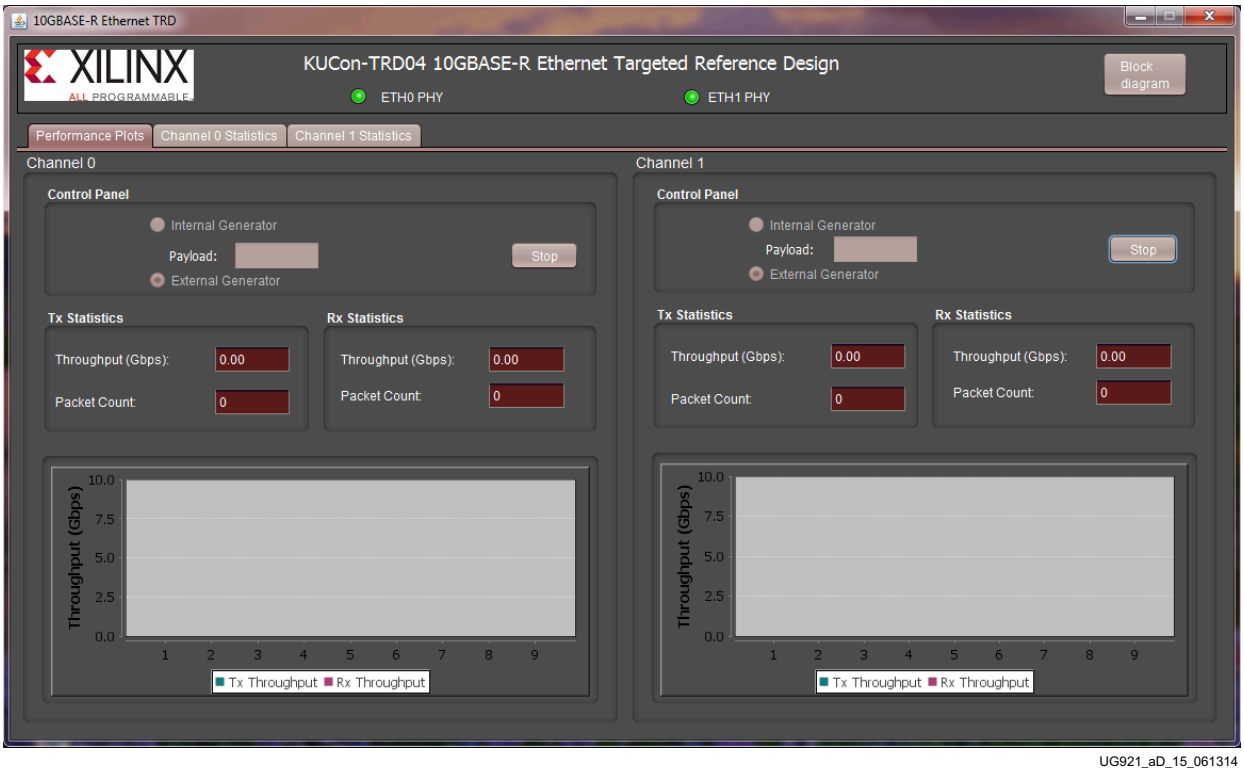
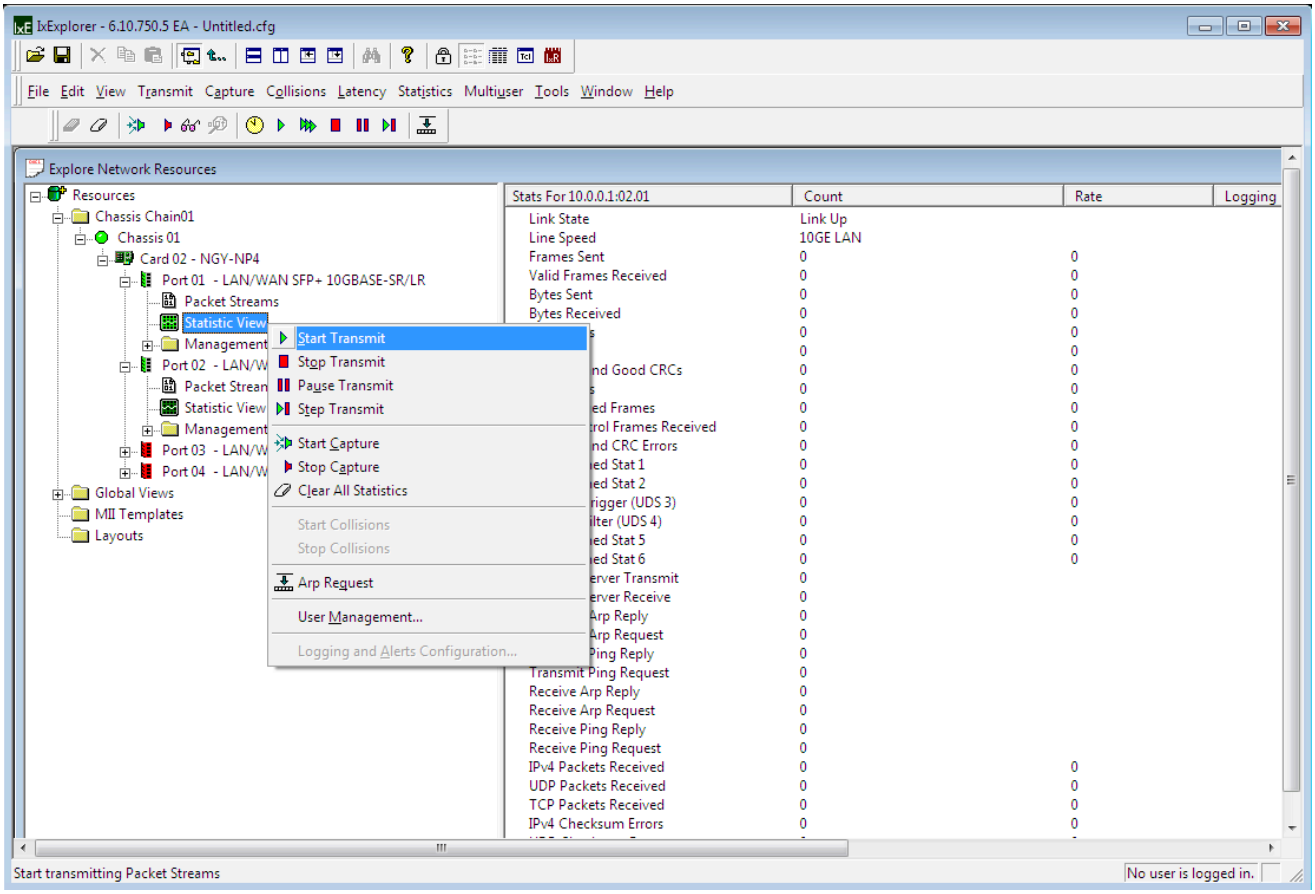


Figure D-15: **Set External Generator on the Channels**

Start Traffic Generation

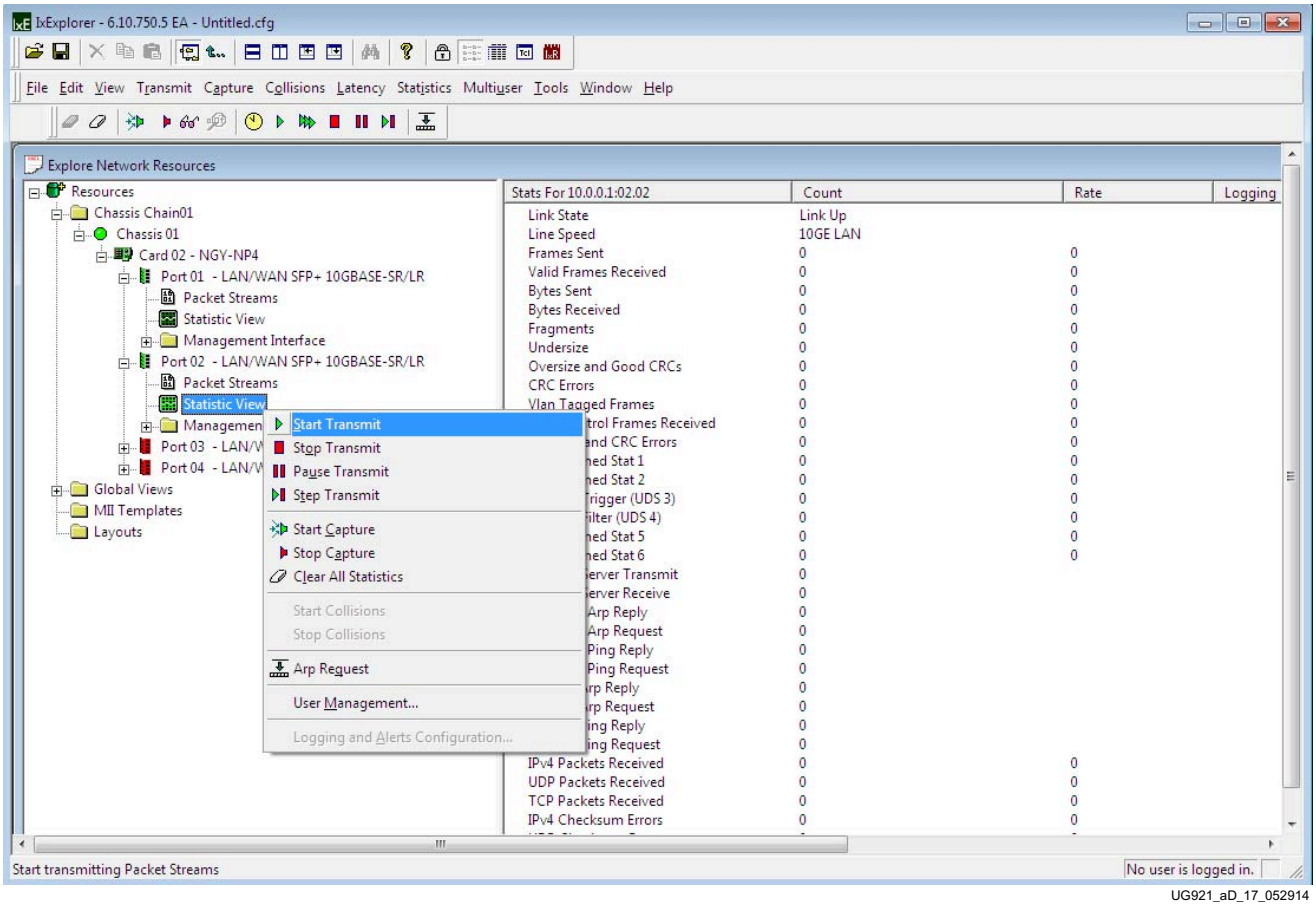
1. To start traffic generation by the Ixia load module, switch to IxExplorer on the Ixia chassis, and right-click on **Port 01 Statistics view** and click on **Start Transmit** (Figure D-16).



UG921_aD_16_052914

Figure D-16: **Start Traffic on Port 01**

2. Repeat step 1 on Port 02 (Figure D-17).



UG921_aD_17_052914

Figure D-17: **Start Traffic on Port 02**

3. The Ethernet Controller application (Figure D-18) shows the performance achieved with a packet size of 1,518 bytes is 9.74 Gb/s per channel per direction.

TIP: *The relationship between payload size and throughput can be demonstrated by changing the payload size. Reducing the payload size set in IxExplorer will cause a dip in performance. Refer to Appendix B, Performance Estimates for performance estimation on 10G Ethernet protocol.*

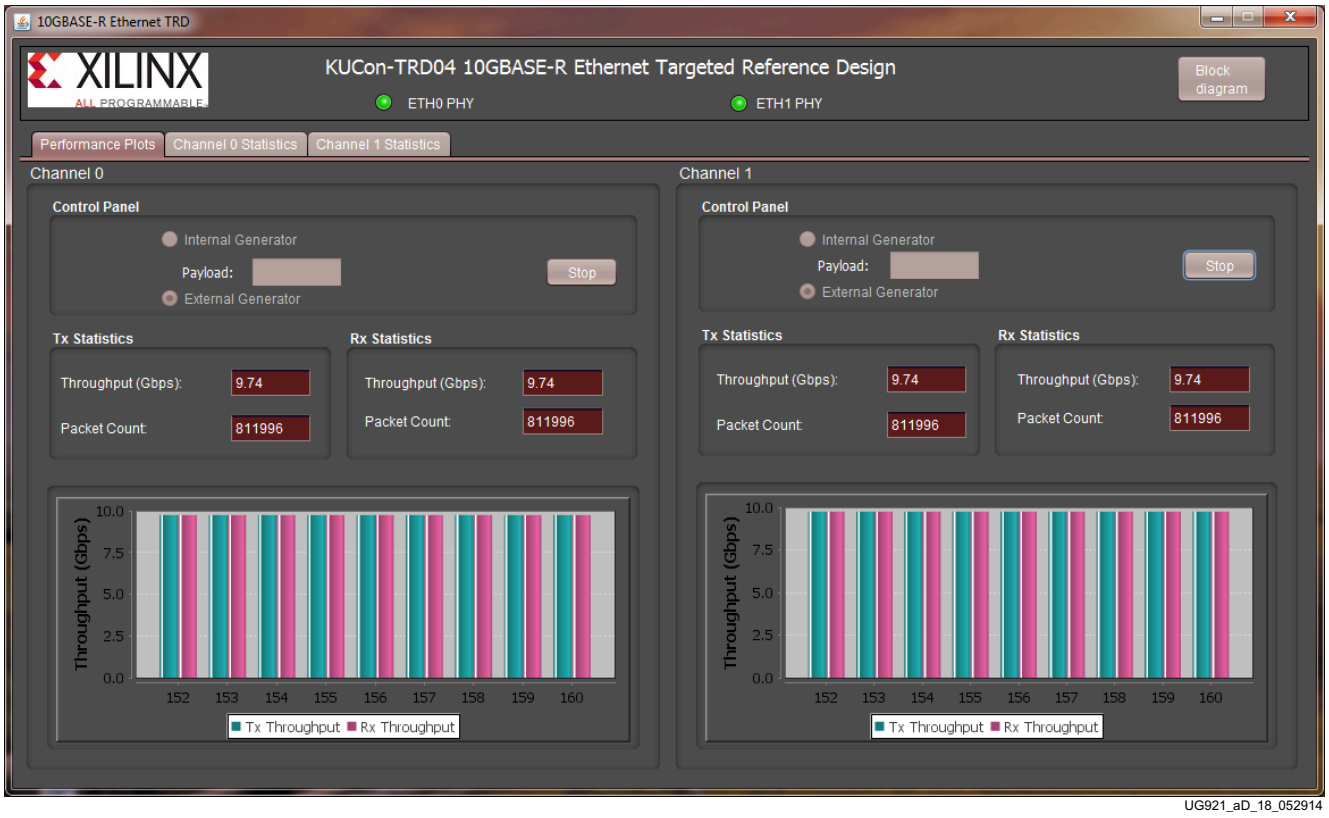


Figure D-18: **Throughput Performance Plots, External Generator**

4. Using IxExplorer (Figure D-19), stop traffic generation on Port 01 by right-clicking on **Port 01 Statistics view** then click on **Stop Transmit**.

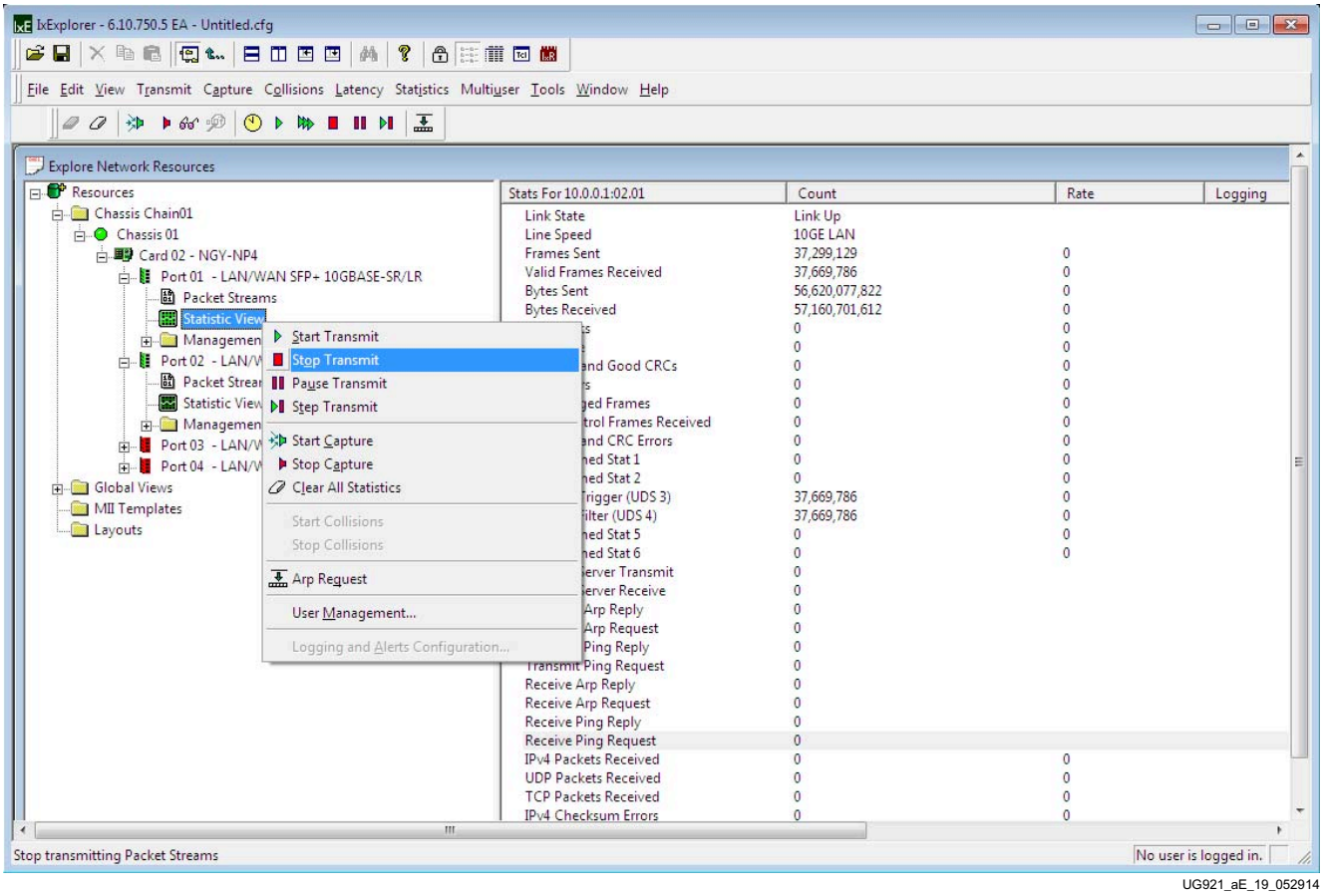
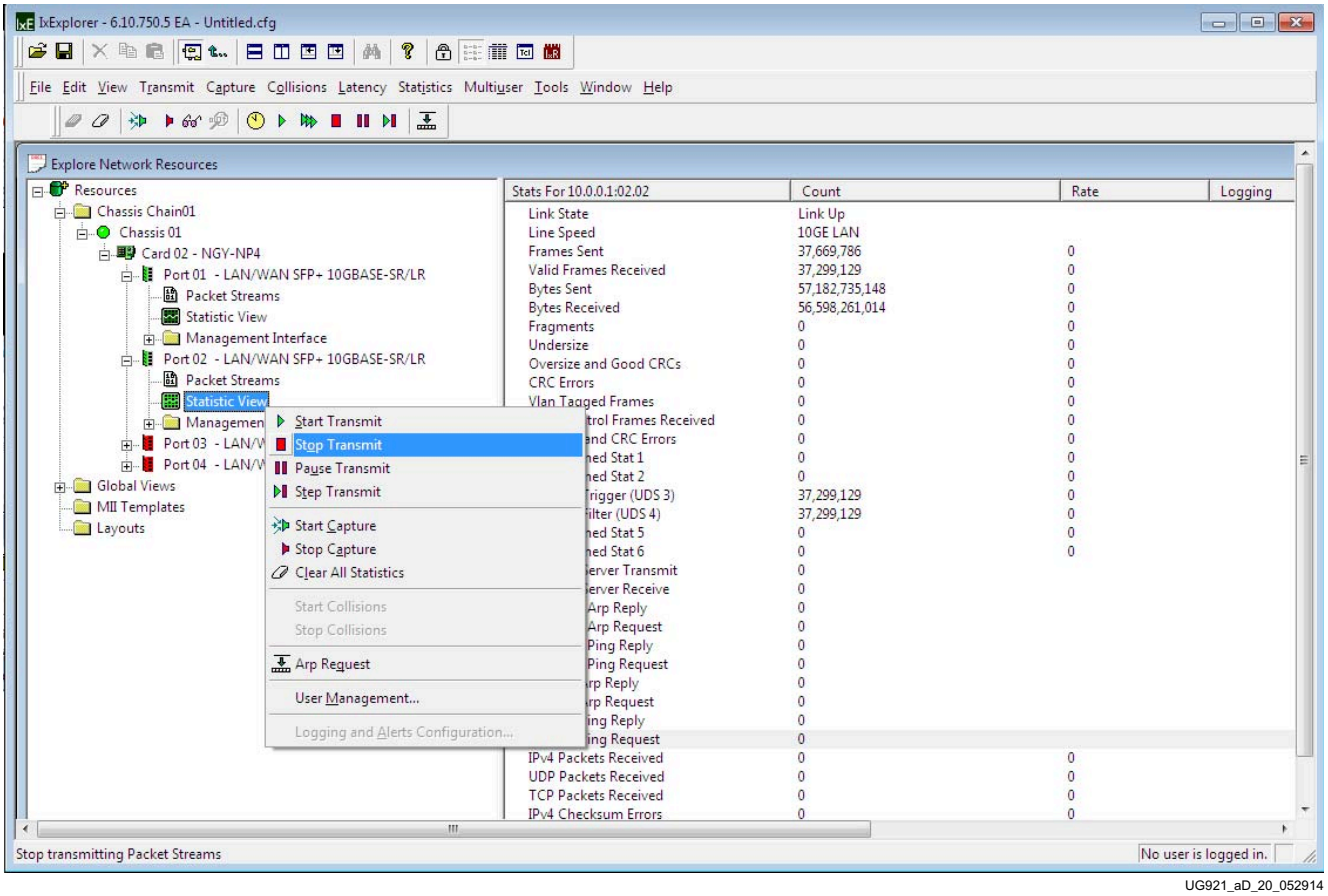


Figure D-19: **Stop Traffic on Port 01**

5. Repeat step 4 on Port 02 (Figure D-20).



UG921_aD_20_052914

Figure D-20: **Stop Traffic on Port 02**

6. Select **Statistics View** for Port 01 (Figure D-21) and verify if any packets were in error or were dropped.

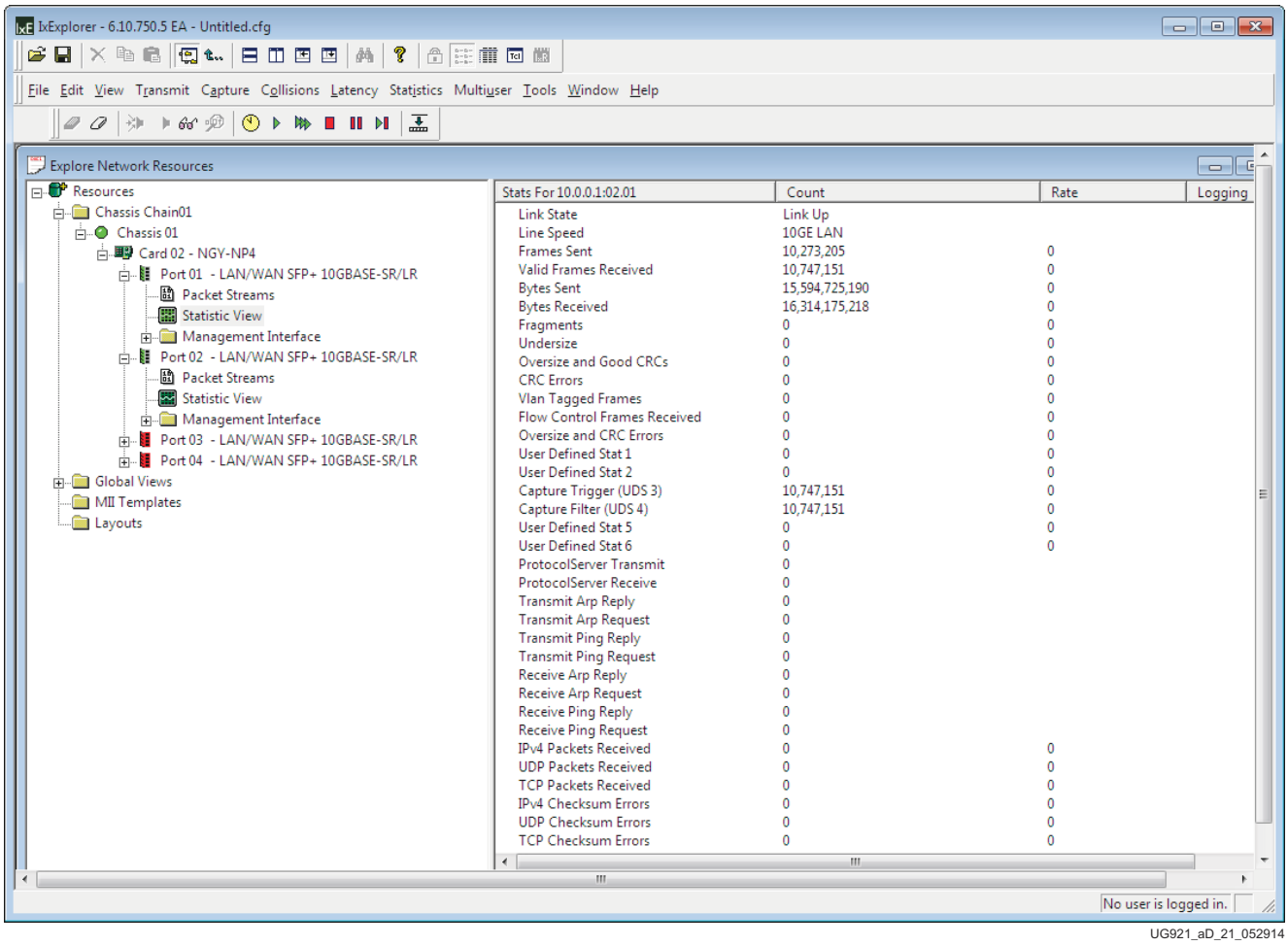


Figure D-21: **Traffic Statistics on Port 01**

7. Select **Statistics View** for Port 02 (Figure D-22) and verify if any packets were in error or were dropped. The transmit frame count for Port 02 should match the receive frame count for Port 01. The transmit frame count for Port 01 should match the receive frame count for Port 02.

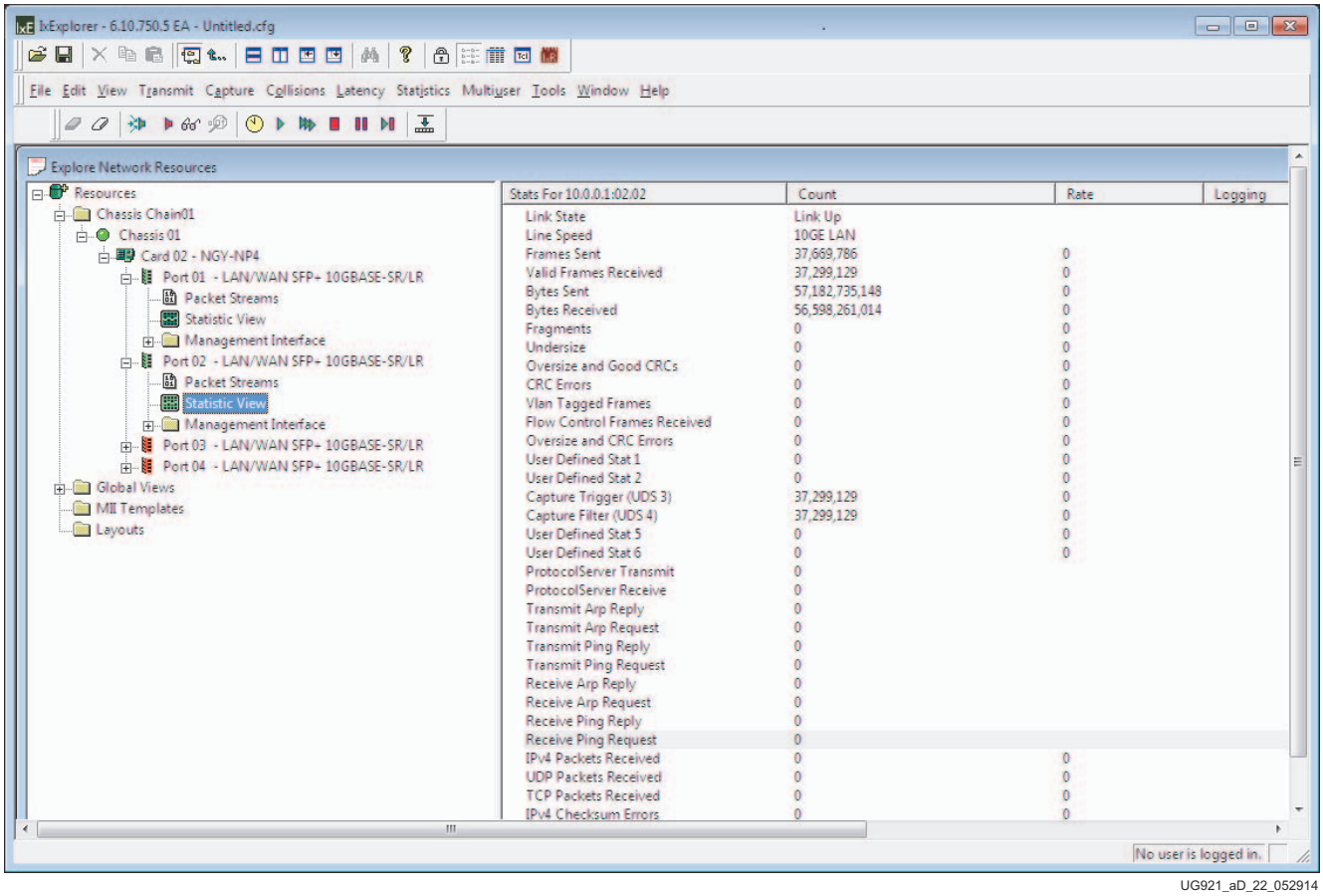


Figure D-22: **Traffic Statistics on Port 02**

8. Using the Ethernet Controller application, select the **Channel 0 Statistics** tab and verify if any packets were in error or were dropped (Figure D-23).

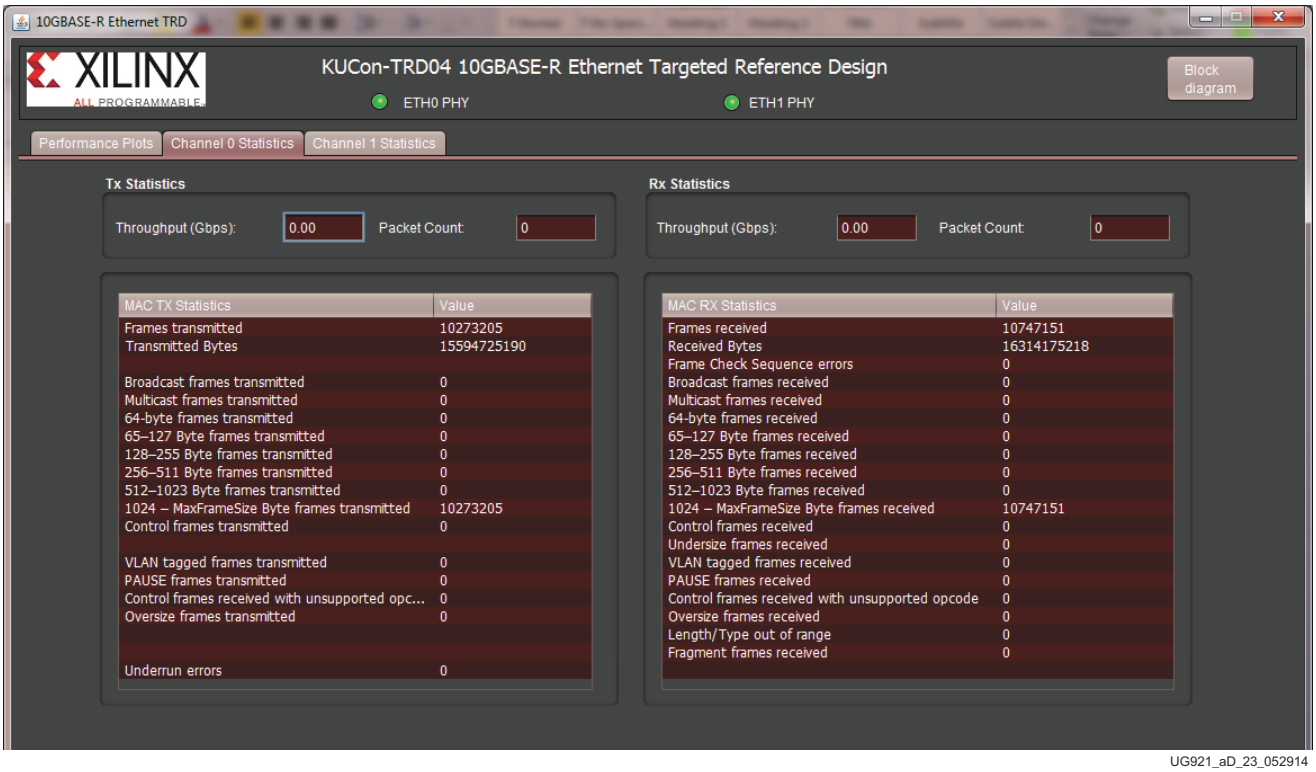


Figure D-23: **Channel 0 MAC statistics**

9. Using the Ethernet Controller application, select the **Channel 1 Statistics** tab and verify if any packets were in error or were dropped (Figure D-24). The TX MAC statistics for Channel 1 should match the RX MAC statistics of Channel 0. The TX MAC statistics for Channel 0 should match the RX MAC statistics of Channel 1. The numbers should also match up with the statistics reported by the Ixia load module in IxExplorer.

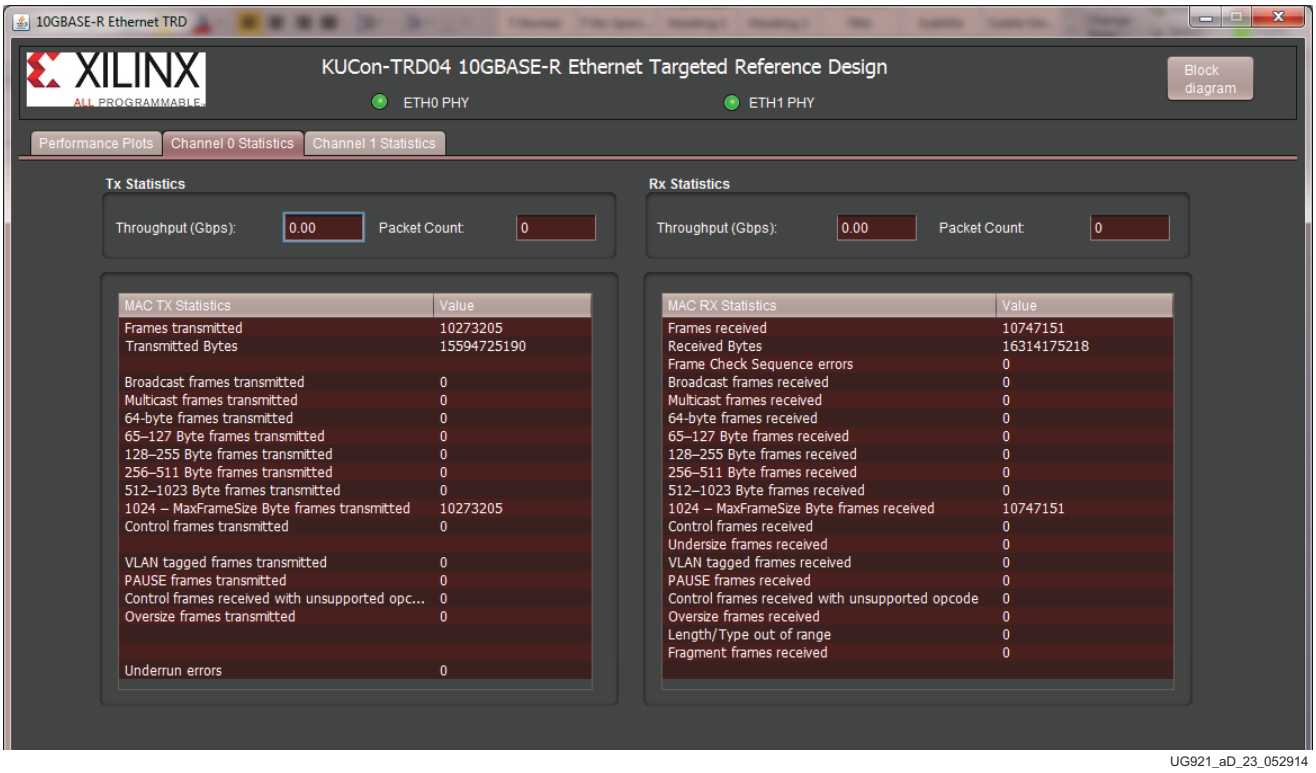


Figure D-24: **Channel 1 MAC Statistics**

IMPORTANT: *When running traffic on both channels, If external generator mode is selected on one channel, then external generator mode should also be selected on the other channel.*

TIP: *Other IxExplorer traffic options can be used to test the 10GBASE-R Ethernet targeted reference design.*

Additional Resources and Legal Notices

Xilinx Resources

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

For a glossary of technical terms used in Xilinx documentation, see the Xilinx Glossary.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

The most up-to-date information for this design is available on these websites:

KCU105 Evaluation Kit

KCU105 Evaluation Kit Documentation

KCU105 Evaluation Kit Answer Record (AR 63175)

These documents and sites provide supplemental material:

1. *Vivado Design Suite User Guide Release Notes, Installation, and Licensing* (UG973)
2. *Silicon Labs CP210x USB-to-UART Installation Guide* (UG1033)
3. 10 Gigabit Ethernet PCS/PMA (10GBASE-R) webpage
4. 10 Gigabit Ethernet Media Access Controller (10GEMAC) webpage
5. *LogiCORE IP 10-Gigabit Ethernet PCS/PMA Product Guide* (PG068)

6. *LogiCORE IP 10-Gigabit Ethernet MAC Product Guide* (PG072)
7. *Vivado Design Suite User Guide Logic Simulation* (UG900)
8. *UltraScale Architecture System Monitor* (UG580)
9. *LogiCORE IP AXI4-Lite IPIF Product Guide for Vivado Design Suite* (PG155)
10. *LogiCORE IP AXI UART Lite Product Guide* (PG142)
11. *LogiCORE IP System Management Wizard Product Guide* (PG185)
12. *KCU105 Evaluation Board for the Kintex UltraScale FPGA* (UG917)
13. MicroBlaze Soft Processor Core
14. AXI Interconnect
15. AXI UART Lite
16. Processor System Reset Module

These external websites provide supplemental material and information:

17. Ixia
    XM2 Portable Chassis, NGY-NP4-01 10-Gigabit Application Network Processor Load Module and software CD
18. Avago Technologies
    AFBR-709SMZ 10 Gb/s Ethernet, 850 nm, 10GBASE-SR, SFP+ Transceiver
19. Amphenol Corporation
    Amphenol part number FO-LCX2SIMP00-003, LC-LC Simplex Single-Mode 9/125 Fiber Optic Patch Cable (OFNR Riser) - LC Male to LC Male (Figure D-1, page 72 shows cable application)
20. Amphenol Corporation
    Amphenol part number FO-10GGBLCX20-001, LC-LC Duplex 10Gb Multimode 50/125 OM3 Fiber Optic Patch Cable - 2 x LC Male to 2 x LC Male (Figure 3-1, page 16 shows cable application)
21. Silicon Labs
    CP210x USB to UART Bridge VCP Drivers
22. Oracle
    Java SE Runtime Environment 7 Downloads

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos.

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.

This document contains preliminary information and is subject to change without notice. Information provided herein relates to products and/or services not yet available for sale, and provided solely for information purposes and are not intended, or to be construed, as an offer for sale or an attempted commercialization of the products and/or services referred to herein.