KCU105 PCI Express Control Plane TRD User Guide

KUCon-TRD01

Vivado Design Suite

UG918 (v2015.2) June 30, 2015

XILINX
ALL PROGRAMMABLE™

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/30/2015	2015.2	Updated for Vivado Design Suite 2015.2.
05/05/2015	2015.1	Updated for Vivado Design Suite 2015.1. TRD ZIP file changed to `rdf0305-kcu105-trd01-2015-1.zip`. Updated Information about resource utilization for the base design and the user extension design in Table 1-1 and Table 1-2. Added information about Windows 7 driver support of the reference design, updating these: sections: Components, Features, and Functions, Computers, Software, and Appendix A, Directory Structure. Updated Figure 5-3, Figure A-1, and Table A-1 to include Windows information. The section Set Up and Install Drivers on the Windows 7 Host Computer was added to Chapter 2, Setup. The section *Using the QuestaSim/ModelSim Simulator* was removed from Chapter 4, Implementing and Simulating the Design, because QuestaSim simulation is not supported in Vivado tool release 2015.1.
02/26/2015	2014.4.1	Initial Xilinx release.

Table of Contents

Introduction

This document describes the features and functions of the PCI Express® Control Plane targeted reference design (TRD). It also describes how to set up, operate, test, and modify the design.

PCI Express Control Plane TRD Overview

The PCI Express Control Plane TRD targets the Kintex® UltraScale™ XCKU040-2FFVA1156E FPGA running on the KCU105 evaluation board. It demonstrates a control plane application using a PCI Express Endpoint block in a x1 Gen1 configuration. Simple base address register (BAR)-mapped read and write transactions are demonstrated using a kernel mode software driver controlled by the Control & Monitoring graphical user interface (GUI).

Figure 1-1 shows the PCI Express Control Plane TRD block diagram.

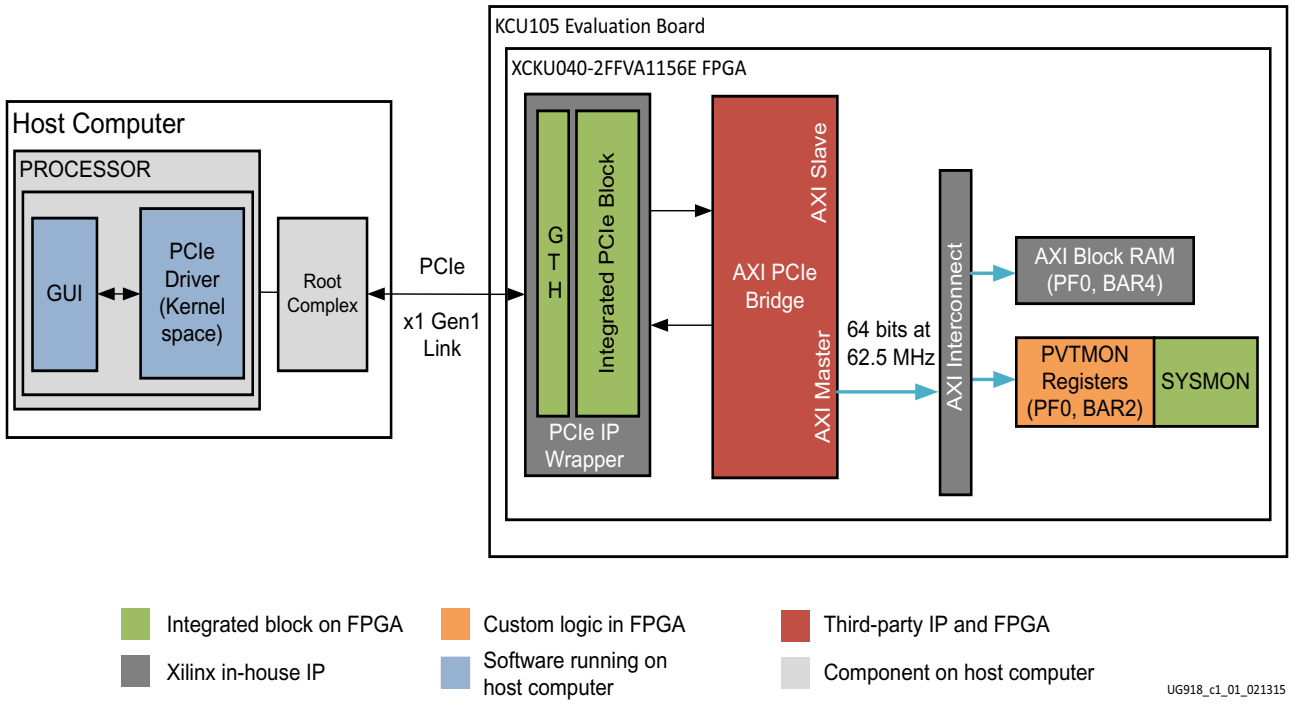


Figure 1-1: **PCI Express Control Plane Targeted Reference Design**

The AXI Bridge from Northwest Logic (NWL) [Ref 1] is used to demonstrate PCIe-to-AXI conversion of transactions. The downstream slaves include a power, voltage, and temperature (PVT) module monitoring parameters from the FPGA system monitor and AXI block RAM IP targeted to BARs.

Components, Features, and Functions

The PCI Express Control Plane TRD includes:

- PCIe x1 Gen1 Endpoint operating at 2.5 GigaTransfers per second (GT/s) per lane/direction
  - Single physical function with support for three 64-bit BARs
- Northwest Logic AXI Bridge IP Core (this is also called *Expresso IP Core*)
  - Ingress address translation capability
  - AXI3 interface
- 64-bit kernel space drivers for Linux and Windows 7, which run on the host computer
- Control & Monitoring GUI which runs on the host computer to pass user inputs to the PCI Express Control Plane TRD and to display outputs

Resource Utilization

Table 1-1 and Table 1-2 list the resources used by the PCI Express Control Plane base and user extension reference designs, respectively, after synthesis has run. Place and route can alter these numbers based on placements and routing paths, so these numbers are to be used as a rough estimate of resource utilization. These numbers might vary based on the version of the PCI Express Control Plane TRD and the tools used to regenerate the design.

Table 1-1: **Resource Utilization for Base Design**

Resource Type	Used	Available	Usage (%)
CLB registers	43,896	484,800	9.05
CLB LUT	27,431	242,400	11.32
Block RAM	22	600	3.66
MMCME3_ADV	1	10	10
Global Clock Buffers	3	240	1.25
BUFG_GT	5	120	4.17
SYSMONE1	1	1	100
IOB	16	520	3.08
GTHE3_CHANNEL	1	20	5
GTHE3_COMMON	0	5	0

Table 1-2: **Resource Utilization for User Extension Design**

Resource Type	Used	Available	Usage (%)
CLB Registers	44,395	484,800	9.16
CLB LUTs	27,817	242,400	11.48
Block RAM	24	600	4
MMCME3_ADV	1	10	10
Global Clock Buffers	3	240	1.25
BUFG_GT	5	120	4.17
SYSMONE1	1	1	100
IOB	16	520	3.08
GTHE3_CHANNEL	1	20	5
GTHE3_COMMON	0	5	0

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 PCI Express® Control Plane 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
- USB cable, standard-A plug to micro-B plug
- Power supply: 100 VAC–240 VAC input, 12 VDC 5.0A output
- ATX power supply
- ATX power supply adapter

Computers

A control computer is required for running the Vivado® Design Suite and configuring the FPGA. It can be a laptop or desktop computer with any operating system supported by Vivado tools, like Redhat Linux or Microsoft® Windows 7.

The reference design test configuration requires a host computer comprised of the chassis, a motherboard with a PCI Express® slot, a monitor, keyboard, and mouse.

- For Linux: A DVD drive is also required.
- For Windows: The 64-bit Windows 7 OS and the Java SE Development Kit 7 must be installed.

Design Tools and Software

- Vivado Design Suite 2015.2
- Fedora 20 LiveDVD, on which the PCI Express Control Plane TRD software and GUI run.

  Note: Fedora 20 is only required if you are using the Linux flow.

Download and installation instructions for 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 the Vivado Design Suite

Install Vivado Design Suite 2015.2 on the control computer. Follow the installation instructions provided in *Vivado Design Suite User Guide Release Notes, Installation, and Licensing* (UG973) [Ref 2].

Download Targeted Reference Design Files

1. Download the PCI Express Control Plane TRD ZIP file `rdf0305-kcu105-trd01-2015-2.zip`.
2. Extract the contents of the file to a working directory.
3. The extracted contents are located at `<working_dir>/kcu105_control_plane`.

The PCI Express Control Plane TRD directory structure is described in Appendix A, Directory Structure.

Set Up and Install Drivers on the Windows 7 Host Computer

Note: If you are using the Linux flow. skip this section and proceed to Set DIP Switches, page 16.

This section includes steps to set up and install KUCon-TRD drivers in a host computer running the Windows 7 64-bit OS.

Disable Driver Signature Enforcement

Note: Windows only allows drivers with valid signatures obtained from trusted certificate authorities to load in a Windows 7 64-bit OS computer. Windows drivers provided for this reference design do not have a valid signature. Therefore, you have to disable *Driver Signature Enforcement* on the host computer as follows:

1. Power up the host system.
2. Press **F8** to go to the Advanced Boot Options menu.
3. Select the **Disable Driver Signature Enforcement** option shown in Figure 2-1, and press **Enter**.

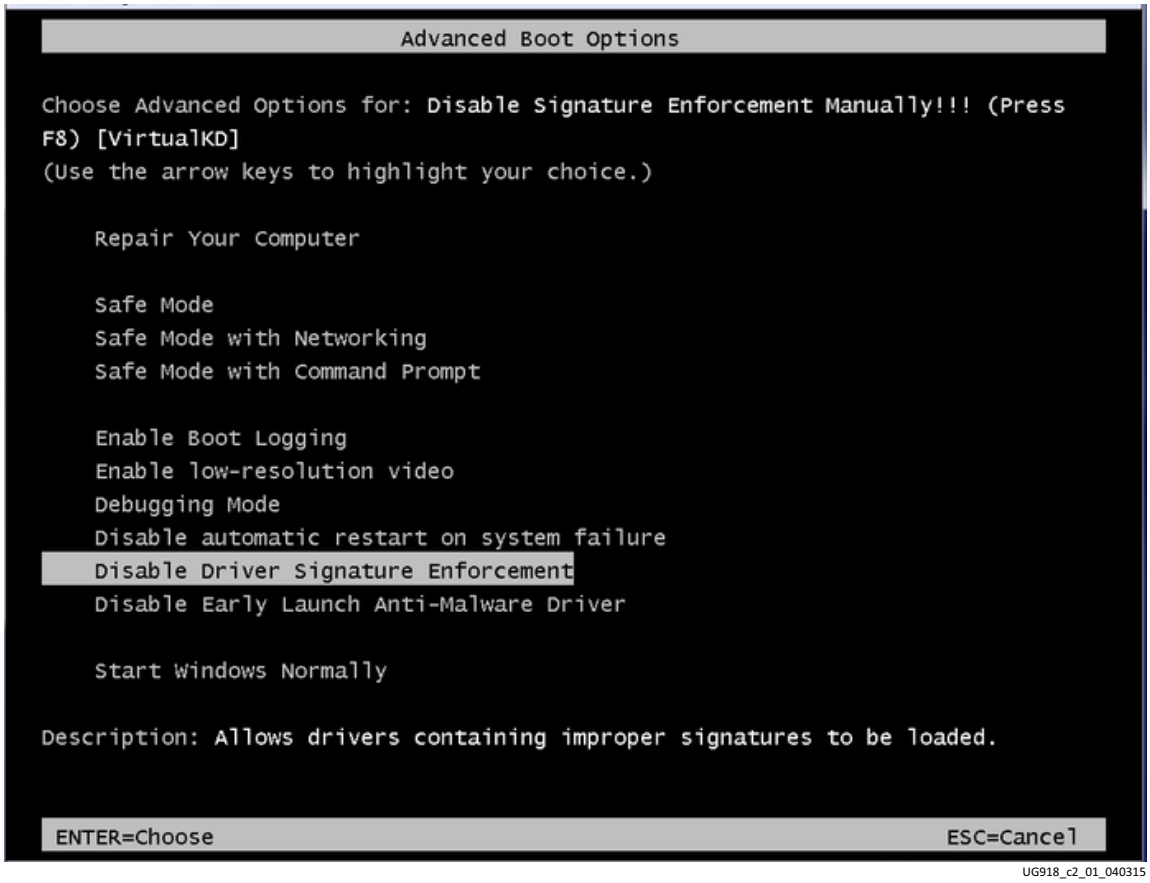


Figure 2-1: **Disable Driver Signature Enforcement**

Install Drivers

1. From Windows Explorer, navigate to the folder in which the reference design is downloaded (`<dir>\kcu105_control_plane\software\windows\`) and run the setup file with Administrator privileges as shown in Figure 2-2.

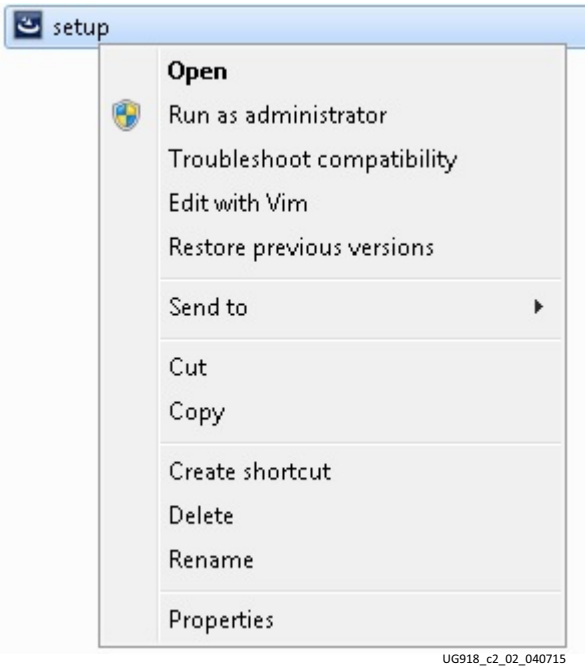


Figure 2-2: **Run the Setup File to Install Drivers**

2. Click **Next** after the InstallShield Wizard opens as shown in Figure 2-3.

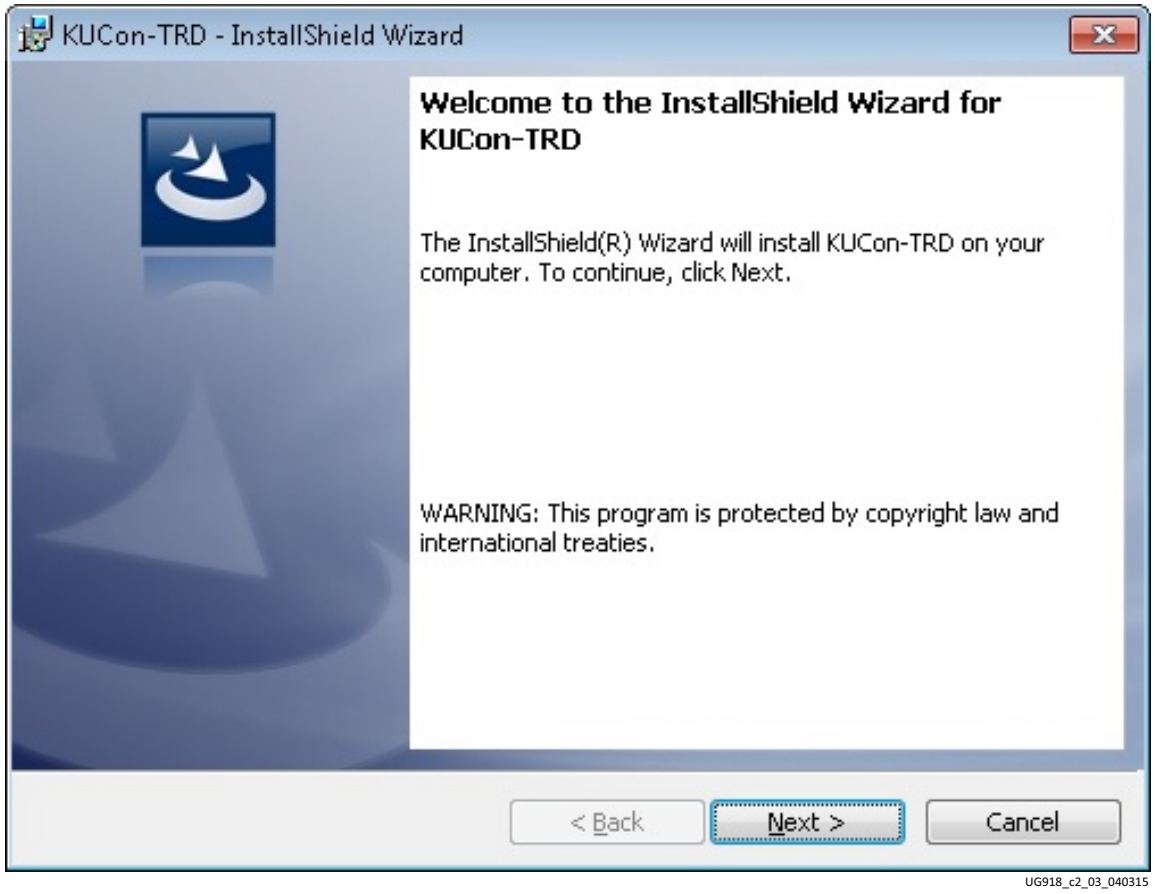


Figure 2-3: **InstallShield Wizard**

3. Click **Next** to install to the default folder; or click **Change** to install to a different folder (see Figure 2-4).

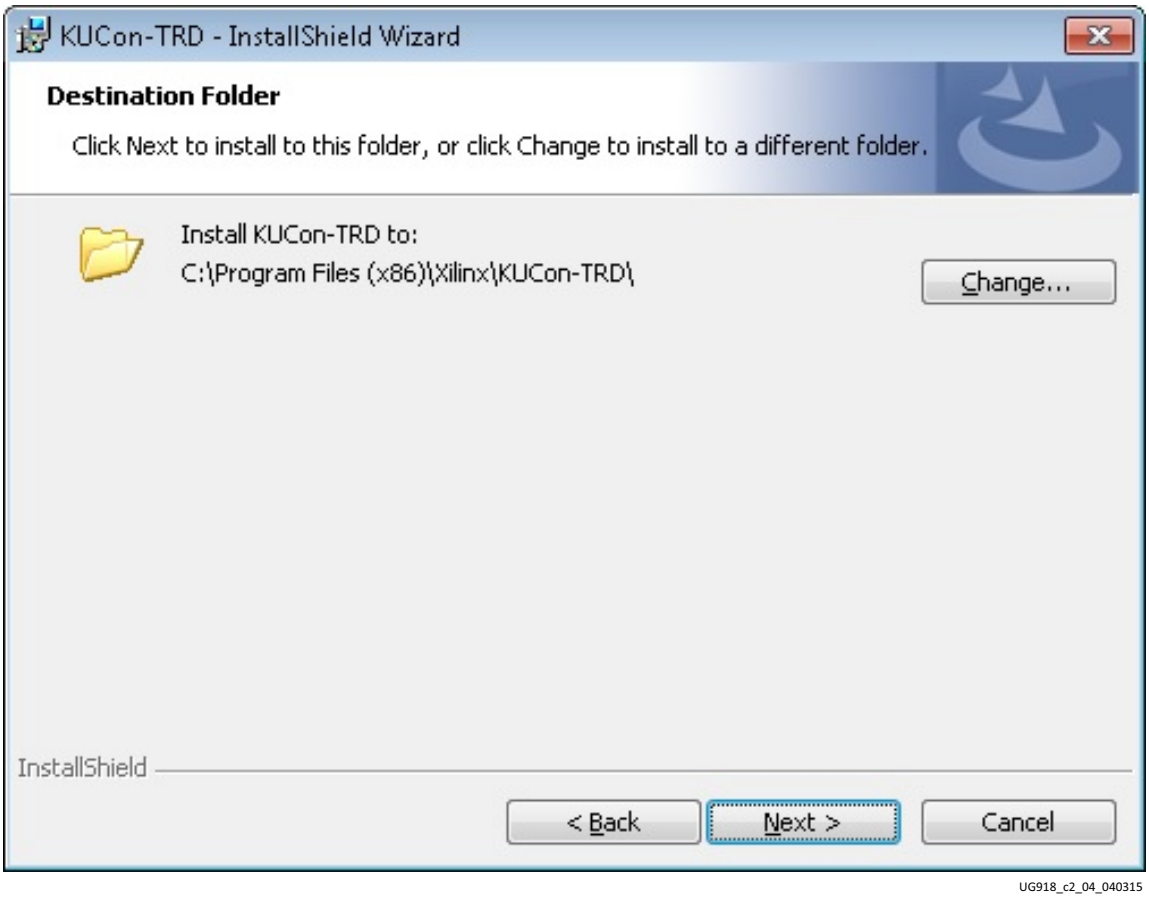


Figure 2-4: **InstallShield - Destination Folder**

4. Click **Install** to begin driver installation (Figure 2-5).

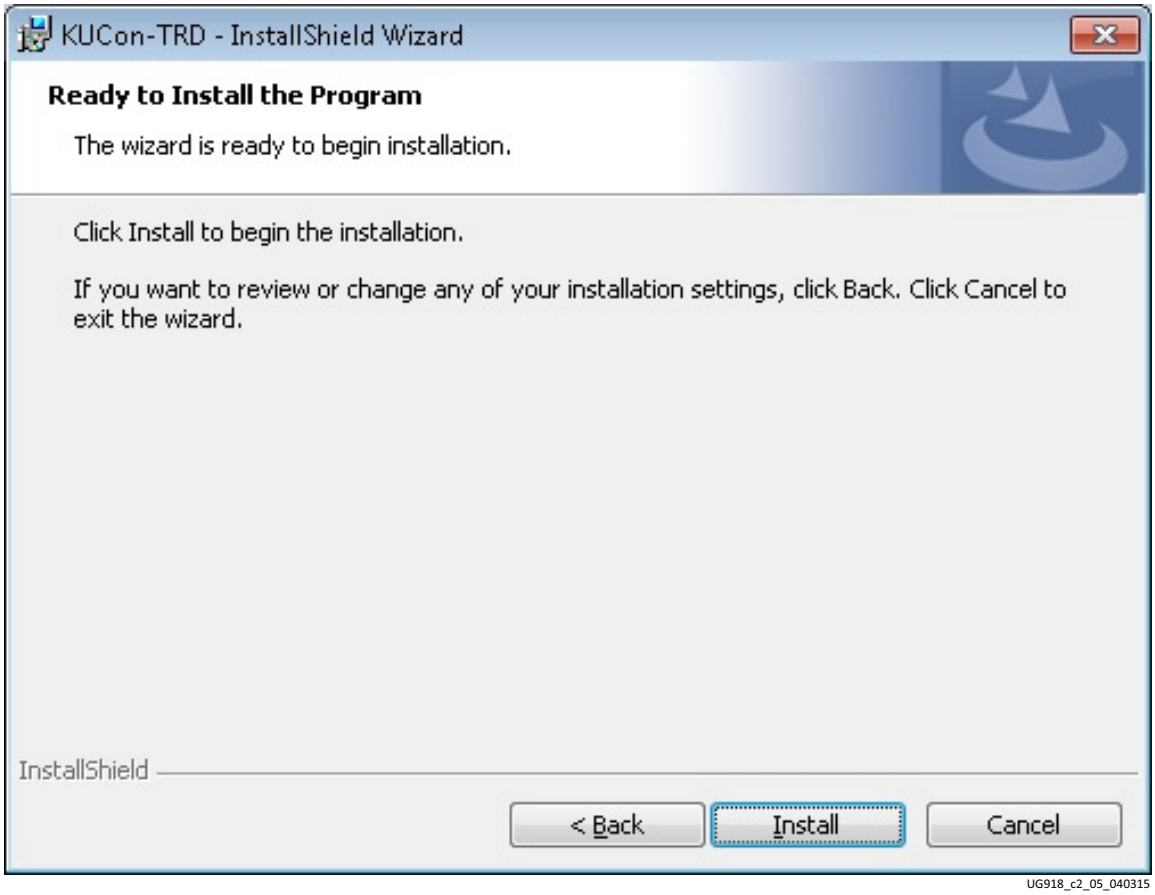


Figure 2-5: **Install Drivers**

A warning screen displays as the drivers are installed, because the drivers are not signed by a trusted certificate authority yet. To install the drivers, ignore the warning message shown in Figure 2-6 and click **Install this driver software anyway**. This warning message pops up two times. Repeat this step.

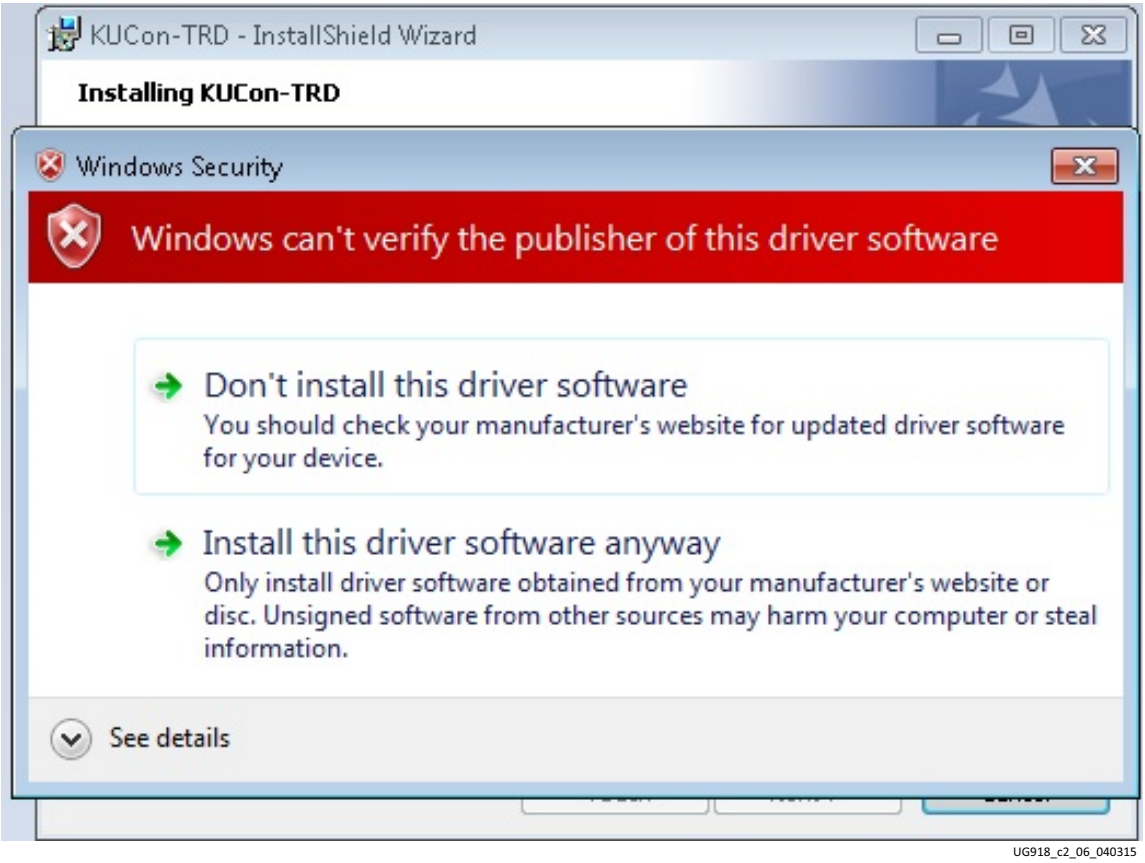


Figure 2-6: **Ignore Windows Security Alert**

5. After installation is complete, click **Finish** to exit the InstallShield Wizard (Figure 2-7).

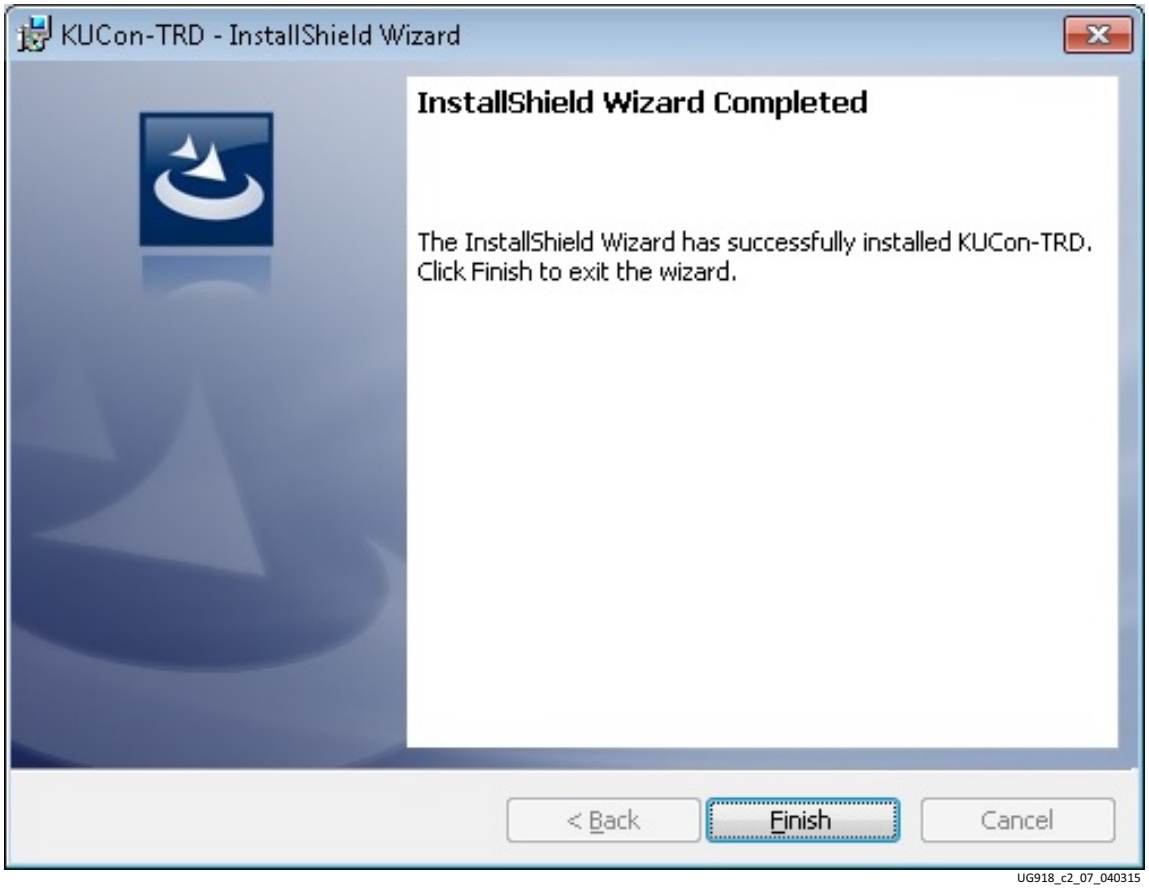


Figure 2-7: **Finish InstallShield Wizard**

Set DIP Switches

Set the DIP switches and jumpers on the KCU105 board to factory default settings as mentioned in *Kintex UltraScale FPGA KCU105 Evaluation Board User Guide* (UG917) [Ref 3].

Ready to Bring Up the Design

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

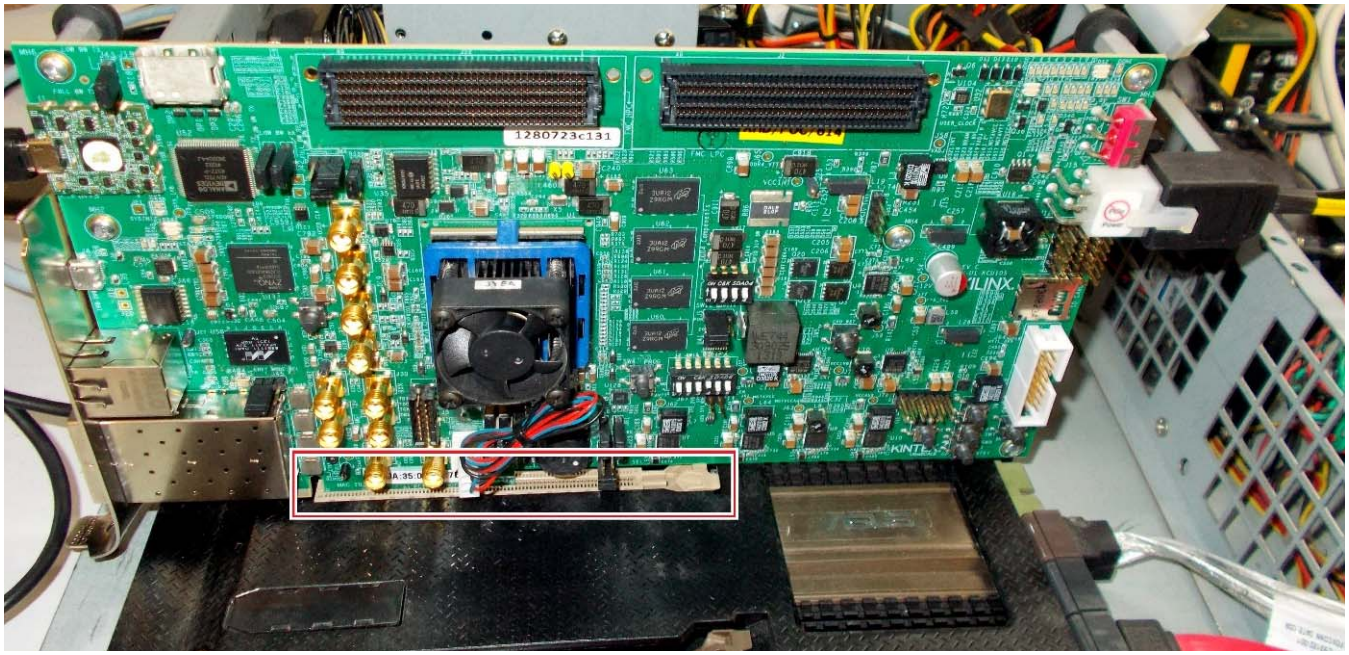
Bringing Up the Design

This chapter describes how to bring up the PCI Express Control Plane TRD.

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

Install the KCU105 Board

For this procedure, see Figure 3-1 for reference.



UG918_c3_01_091314

Figure 3-1: **PCIe Connector Slot**

To install the KCU105 board in the host computer motherboard:

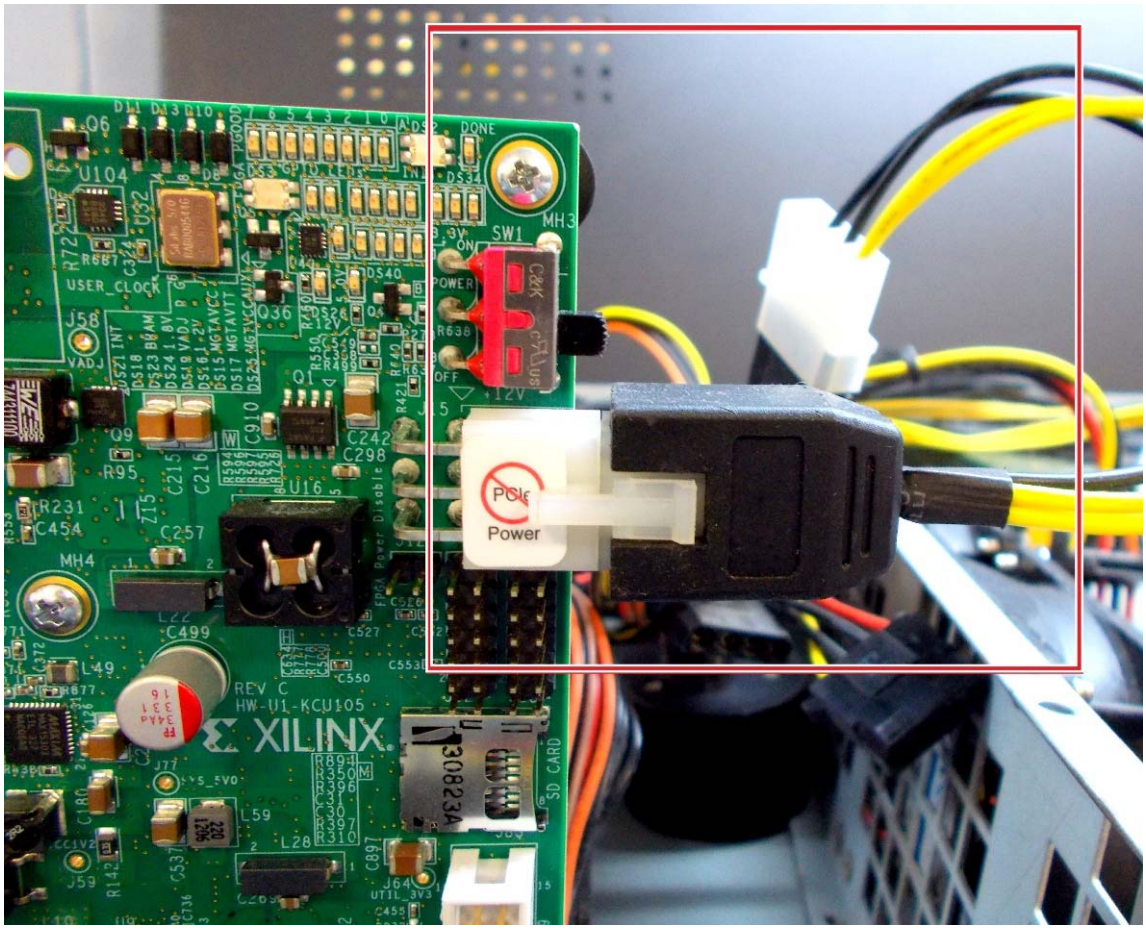
1. Remove all rubber feet and standoffs from the KCU105 board.

2. Power down the host chassis and remove the computer power cord. Ensure that the host computer is powered off.

CAUTION! *Remove the power cord to prevent electrical shock or damage to the KCU105 board or other components.*

3. Open the chassis. Select a vacant PCIe expansion slot and remove the expansion cover at the back of the chassis.
4. Plug the KCU105 board into the PCIe connector at this slot as shown in Figure 3-1.
5. Connect the ATX power supply to the KCU105 board using the ATX power supply adapter cable as shown in Figure 3-2.

Note: A 100 VAC–240 VAC input, 12 VDC 5.0A output external power supply can be substituted for the ATX power supply.



UG918_c3_02_040715

Figure 3-2: **ATX Power Supply Adapter Cable Connecting to KCU105 Board**

6. Slide the KCU105 board power switch SW1 to the ON position. (ON/OFF is marked on the board.)

Set the Host System to Boot from the LiveDVD

Note: This section applies only to the Linux flow. Proceed to Configure the FPGA if you are using Windows.

1. Power on the host system. Stop it during BIOS to select options to boot from a DVD drive. BIOS options are entered by pressing DEL, F12, or F2 keys on most systems.

   Note: If an external power supply is used instead of the ATX power, the FPGA can be configured first. Then power on the host system.

2. Place the Fedora 20 LiveDVD into the DVD drive.

3. Select the option to boot from DVD.

Configure the FPGA

While in BIOS, program the FPGA with the BIT file.

1. Connect the standard-A plug to micro-B plug USB cable to the JTAG port on the KCU105 board and to the control computer laptop as shown in Figure 3-3.

   Note: The host system can remain powered on.

   Note: Figure 3-3 shows a Rev C board. The USB JTAG connector is on the PCIe panel for production boards.

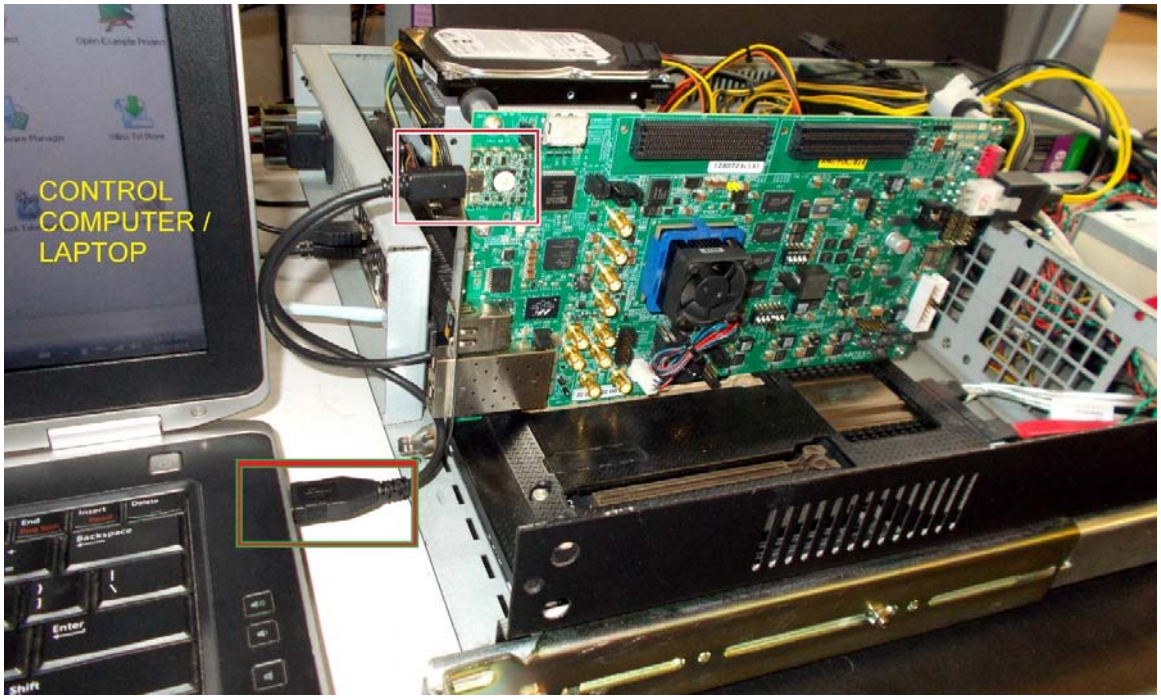


Figure 3-3: **Connect the USB Cable to the KCU105 Board and Control Computer**

2. Launch the Vivado® Integrated Design Environment (IDE) on the control computer:
   a. Select **Start > All Programs > Xilinx Design Tools > Vivado 2015.2 > Vivado 2015.2**.
   b. On the getting started page, click **Open Hardware Manager** (Figure 3-4).

Figure 3-4: **Vivado IDE Getting Started Page, Open Hardware Manager**

3. Open the connection wizard to initiate a connection to the KCU105 board:
   a. Click **Open New Target** (Figure 3-5).

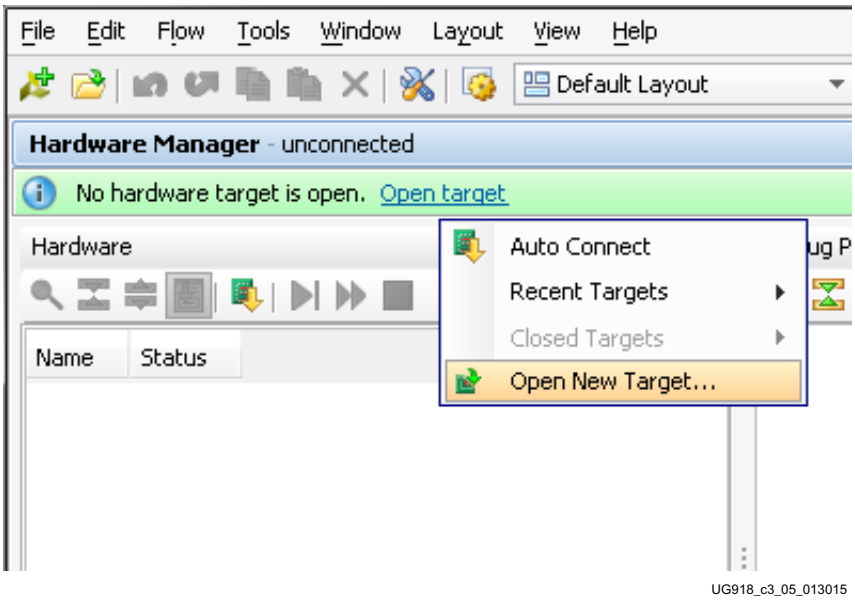


Figure 3-5: **Using the User Assistance Bar to Open a Hardware Target**

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

   a. In the hardware view, right-click **xcku040** and click **Program Device** (Figure 3-6).

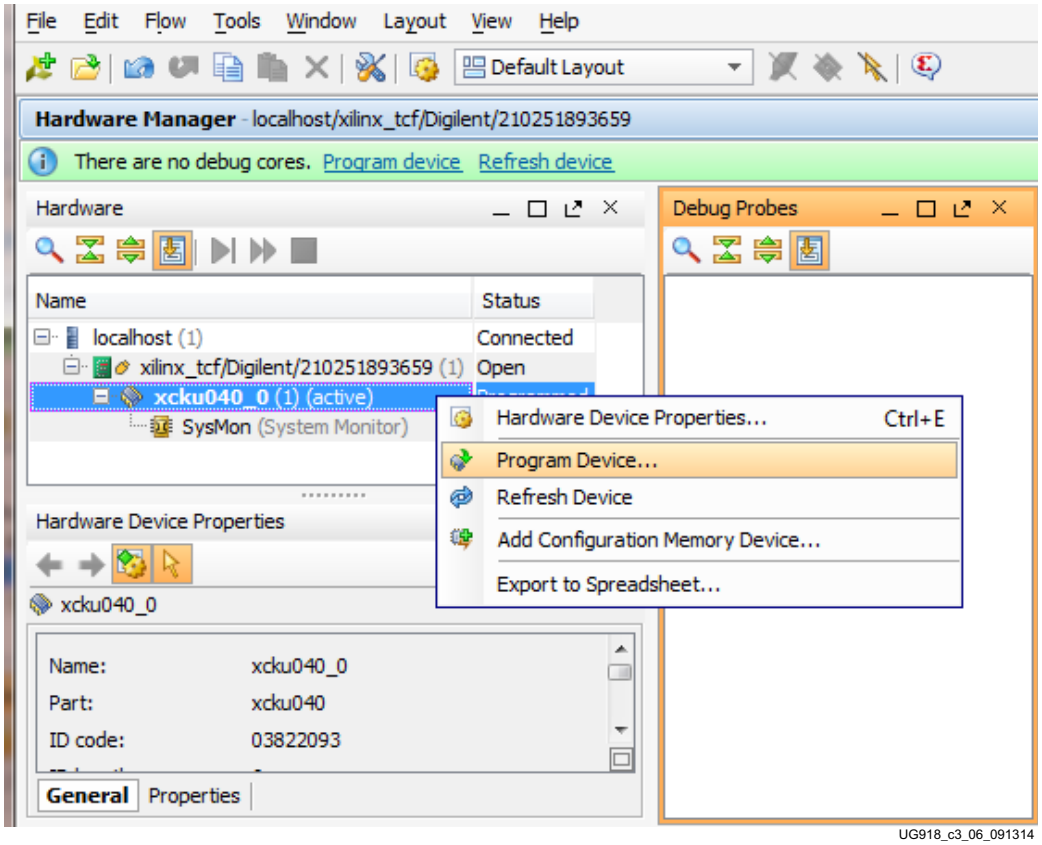


Figure 3-6: **Select Device to Program**

   b. In the **Bitstream file** field, browse to the location of the BIT file `<working_dir>/kcu105_control_plane/ready_to_test/trd01.bit` and click **Program** (see Figure 3-7).

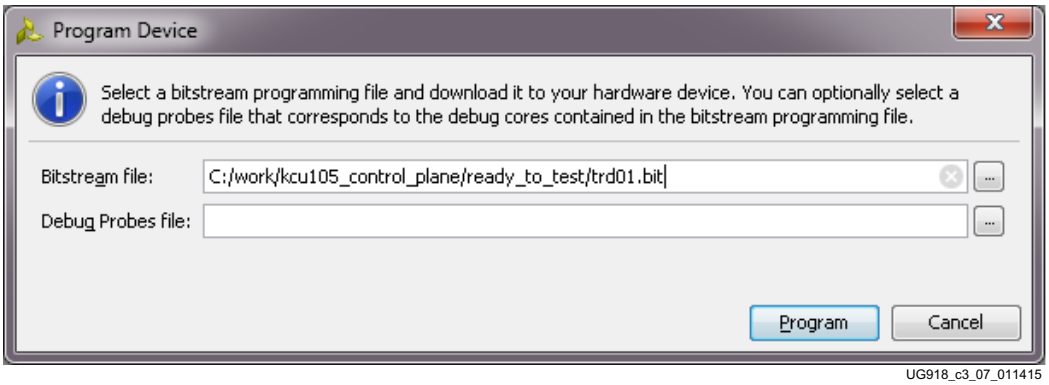


Figure 3-7: **Program Device Window**

5. After the FPGA is programmed, check for LED status on the GPIO LED panel on the top right corner of the board as shown in Figure 3-8.

   LED 0: PCIe user link UP

   LED 1: Heartbeat LED showing the presence of a PCIe user clock.

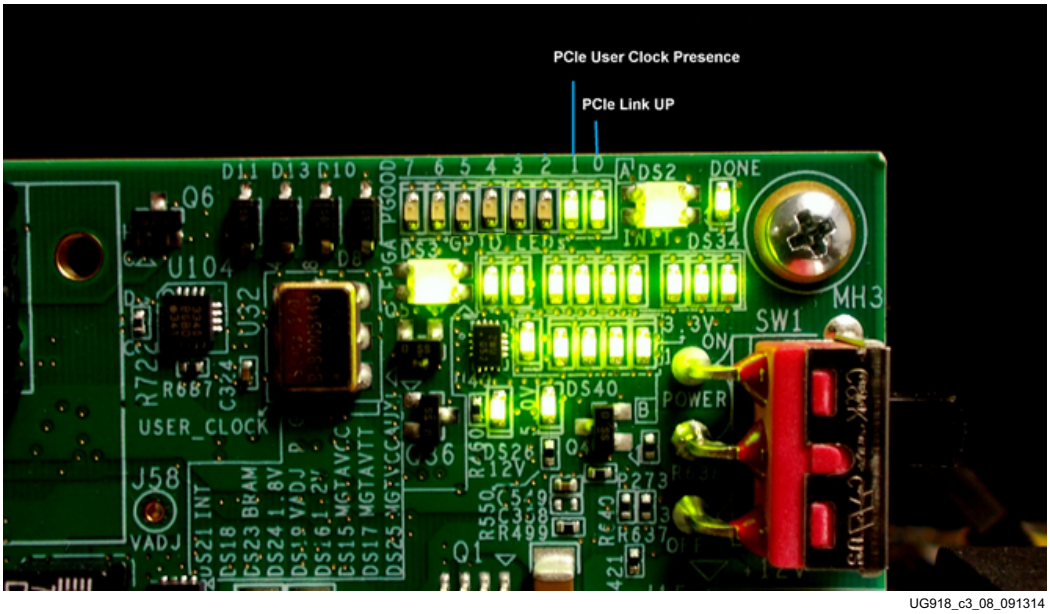


Figure 3-8: **GPIO LED Panel**

6. Exit the BIOS and let the system boot.
7. On most systems, this gives a second reset on the PCIe connector, which should discover the device during enumeration.
   - To know that the PCIe Endpoint is discovered, see Check for PCIe Devices, page 24.
   - If the PCIe Endpoint is not discovered, reboot the system. Do not power off.

Testing the Targeted Reference Design

Testing on a Linux System

When using Linux drivers, use the Fedora 20 LiveDVD provided. Use the following steps to continue with testing on a Linux system.

Figure 3-9 shows different boot stages of Fedora 20. After you reach the third screen shown in Figure 3-9, click the **Try Fedora** option, then click **Close**. It is recommended that you run the Fedora operating system from the DVD.

CAUTION! If you want to install Fedora 20 on the hard drive connected to the host system, click the **Install to Hard Drive** option. **BE CAREFUL! This option erases any files on the hard disk!**

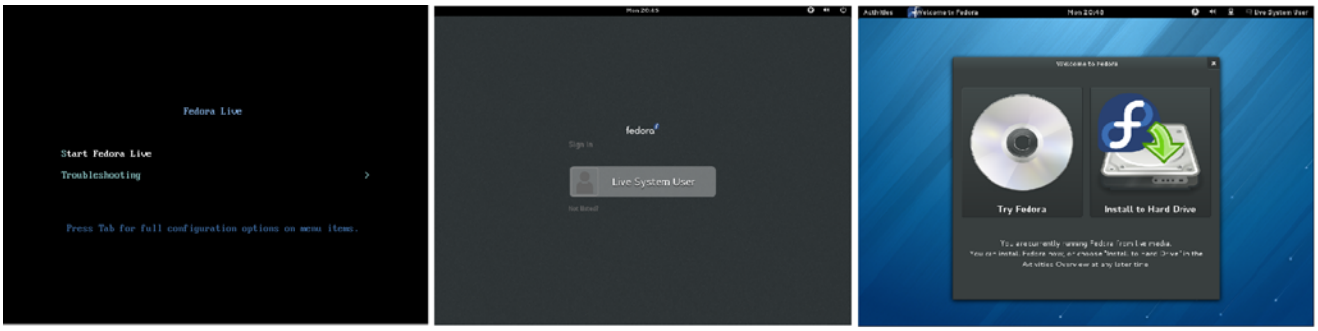


Figure 3-9: **The Fedora 20 Boot Sequence**

Check for PCIe Devices

1. After the Fedora 20 OS boots, open a terminal and use **lspci** to see a list of PCIe devices detected by the host computer.

   $ lspci | grep -i xilinx

   ```
   03:00.0 Memory controller: Xilinx Corporation Device 8011
   ```

Note: If the host computer does not detect the Xilinx PCIe Endpoint, `lspci` does not show a Xilinx device.

Running the Design on the Host Computer

Running the Design on a Linux Host Computer

1. Copy the design ZIP file `rdf0305-kcu105-trd01-2015-2.zip` under KCU105 Evaluation Kit documentation to any directory of your choice (for example `/home/liveuser/`).
2. Open a terminal and type:

   $ cd /home/liveuser/kcu105_control_plane

3. Login as super user by typing:

   $ su

4. Execute the following commands:

   $ chmod +x quickstart.sh

   $ ./quickstart.sh

5. The screen in Figure 3-10 shows the installer page which has detected a PCIe device with ID 8011 in lspci—by default the control plane design is selected. Click **Install** to install the drivers.

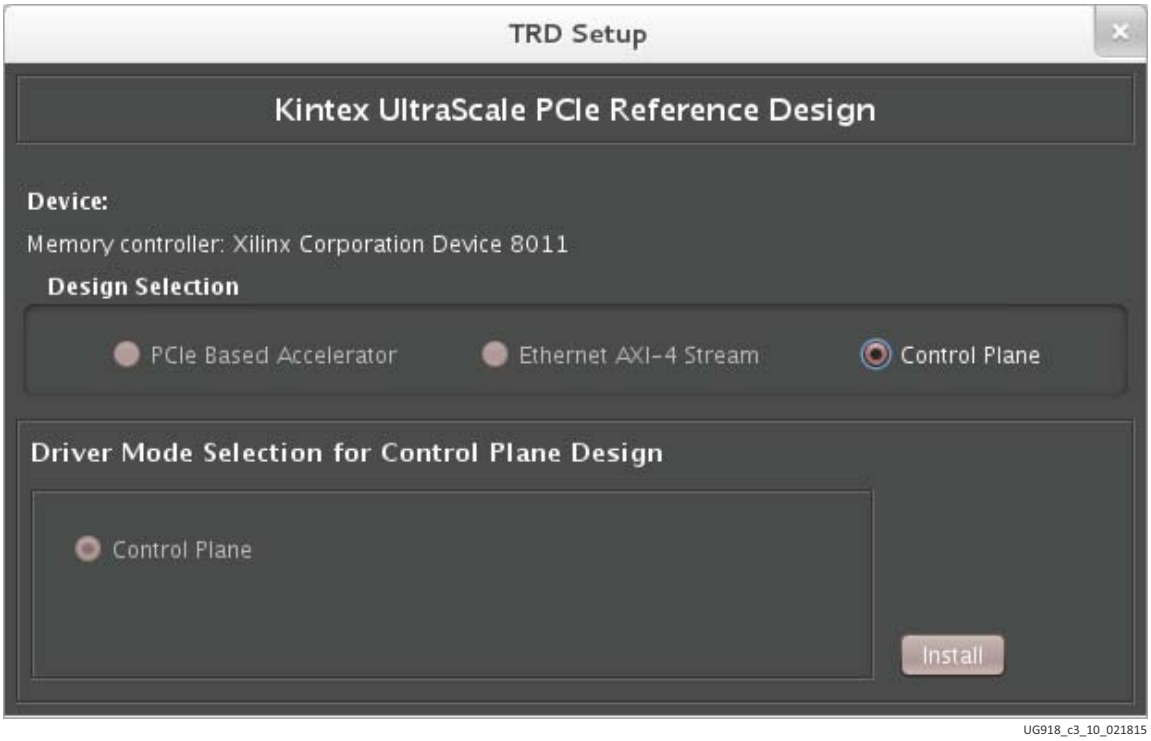


Figure 3-10: **Installer Page Detecting a PCIe Device**

Running the Design on a Windows 7 Host Computer

1. Repeat the steps in section Disable Driver Signature Enforcement, page 10.
2. Open Device Manager (**Start > devmgmt.msc > Enter**) and look for the Xilinx PCI Express Device as shown in Figure 3-11.

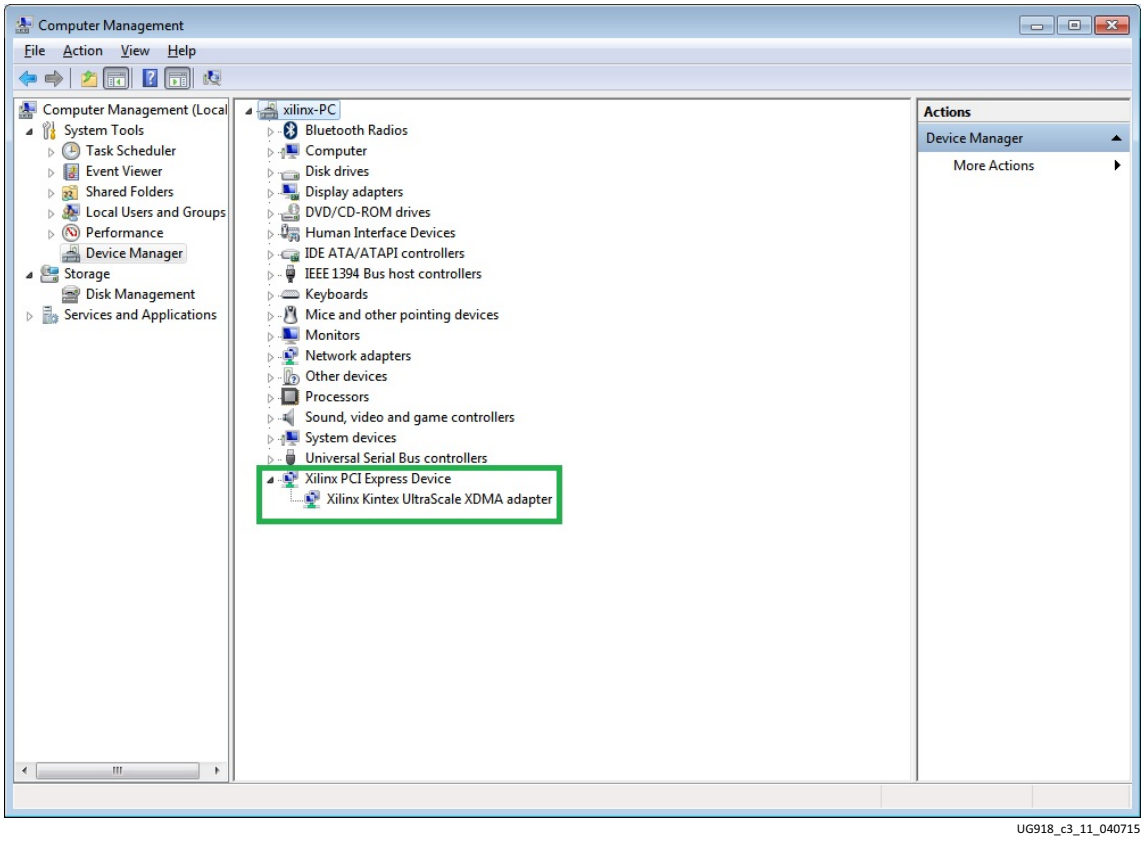


Figure 3-11: **Xilinx PCI Express Device in Device Manager**

3. Open a command prompt with administrator privileges as shown in Figure 3-12.

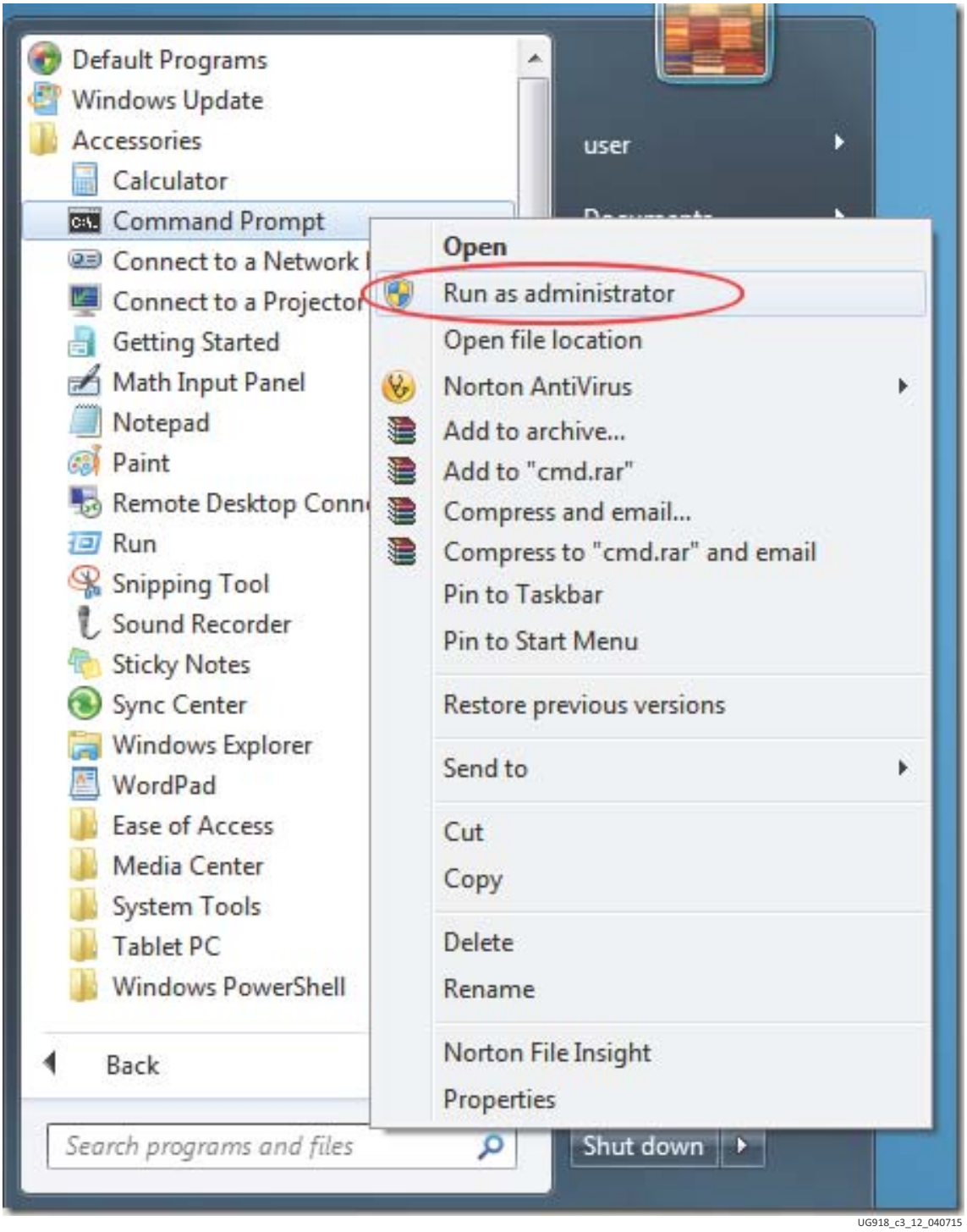


Figure 3-12: **Command Prompt with Administrator Privileges**

4. Navigate to the folder where the reference design is copied:

   cd <dir>\kcu105_control_plane

5. Run the batch script `quickstart_win7.bat`:

   quickstart_win7.bat

6. The screen in Figure 3-13 shows the TRD Setup screen of the GUI. Click **Proceed** to test the reference design.

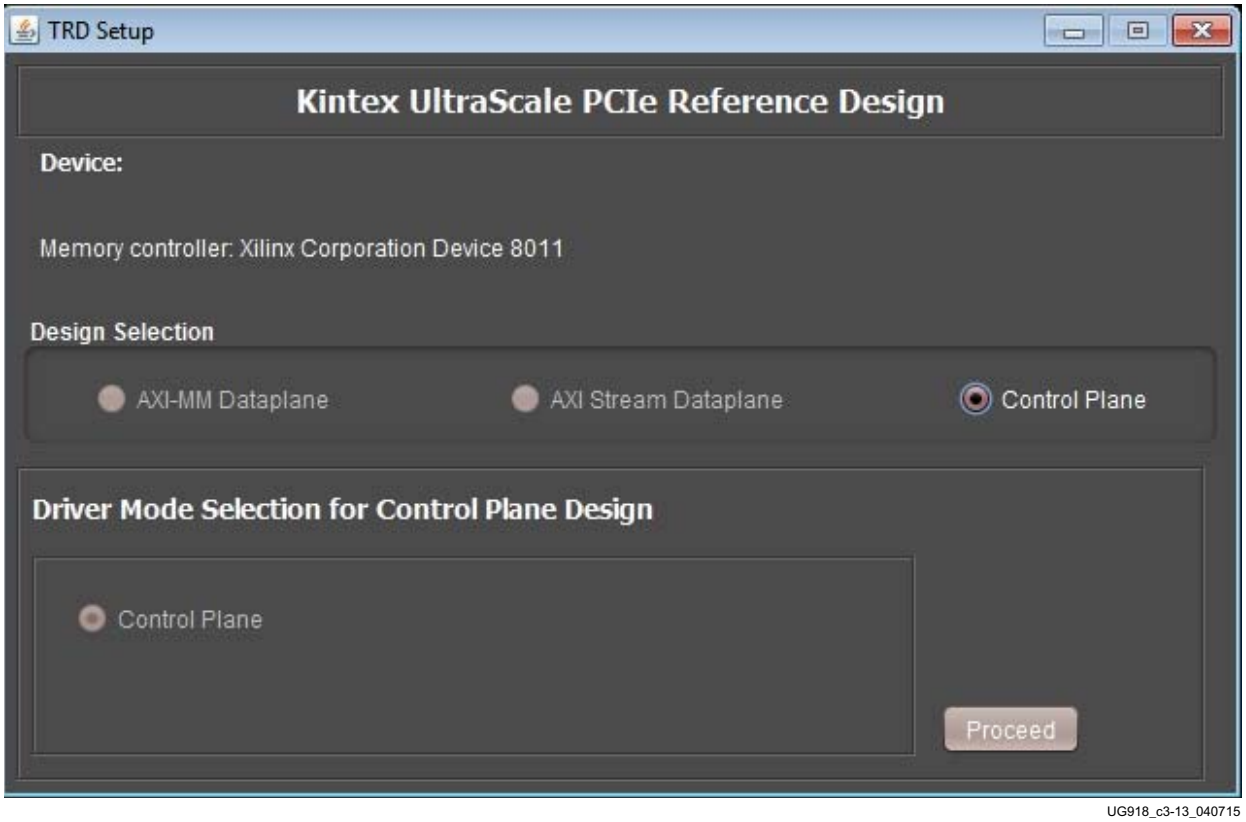


Figure 3-13: **GUI - TRD Setup Screen**

Testing the Reference Design

The GUI provides power monitoring and die temperature (PVTMON is mapped to BAR2), PCIe link status, virtual address assigned to BARs, and user options to access BAR-mapped regions.

The following can be done through the main GUI:

- You can read an address offset from BAR4.
- You can write to an address offset from BAR4 by providing the offset value and data value to be written.

- You can obtain a dump of the data from a specific address offset from BAR4.

By clicking the **Block Diagram** button in the top right corner (Figure 3-14), a block diagram of the design appears.

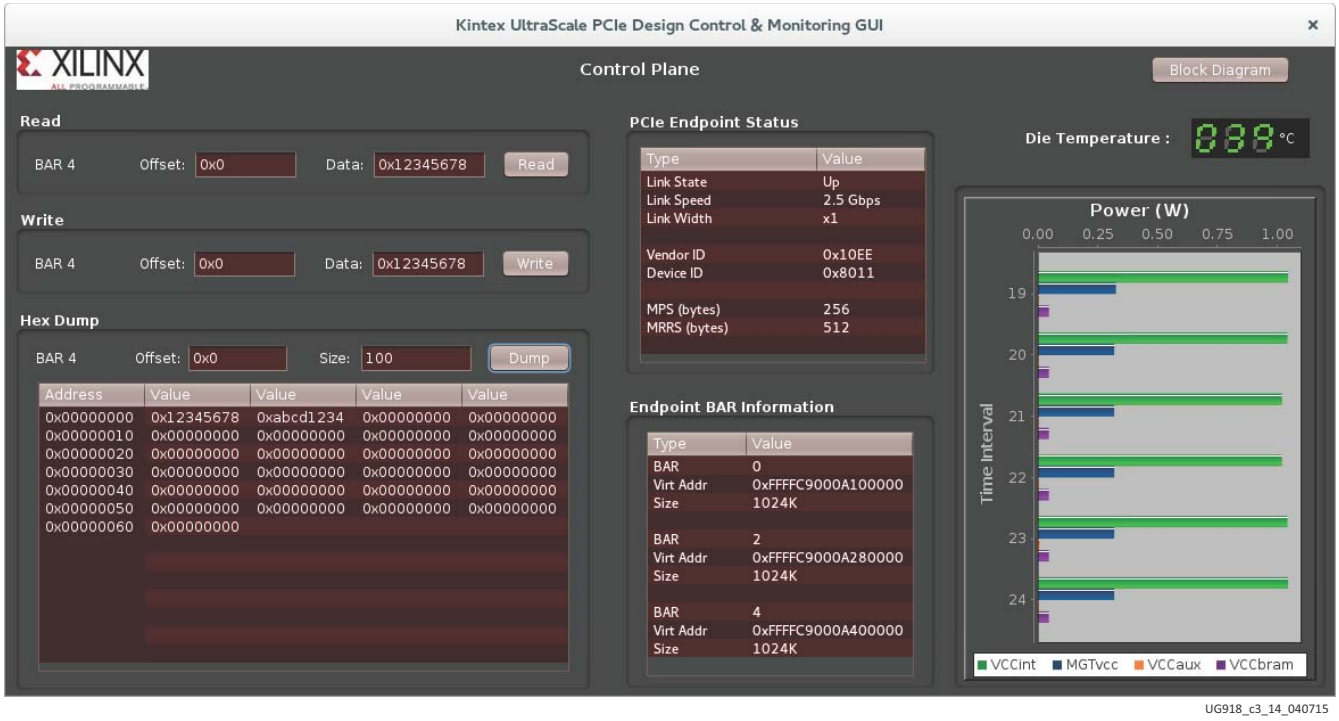


Figure 3-14: **Control & Monitoring GUI**

Closing the GUI

Click the **X** mark on the top right corner of the GUI to close the main screen.

- On a Linux host computer, this step uninstalls the drivers and returns the GUI to the TRD Setup screen.
- On a Windows host computer, this step returns the GUI to the TRD Setup screen. Close the TRD Setup screen and power off the host machine and then the KCU105 board.

To uninstall the drivers on a Windows host computer, use the following steps.

Remove Drivers from a Windows Host Computer

To uninstall drivers:

1. Power on the host machine.
2. After it boots, from Windows Explorer, navigate to the folder in which the reference design is downloaded (`<dir>\kcu105_control_plane\software\windows\`).

3. Run the setup file with Administrator privileges as shown in Figure 2-2.
4. Click **Next** after the InstallShield Wizard opens as shown in Figure 2-3.
5. Select **Remove** and click **Next** to proceed (Figure 3-15).

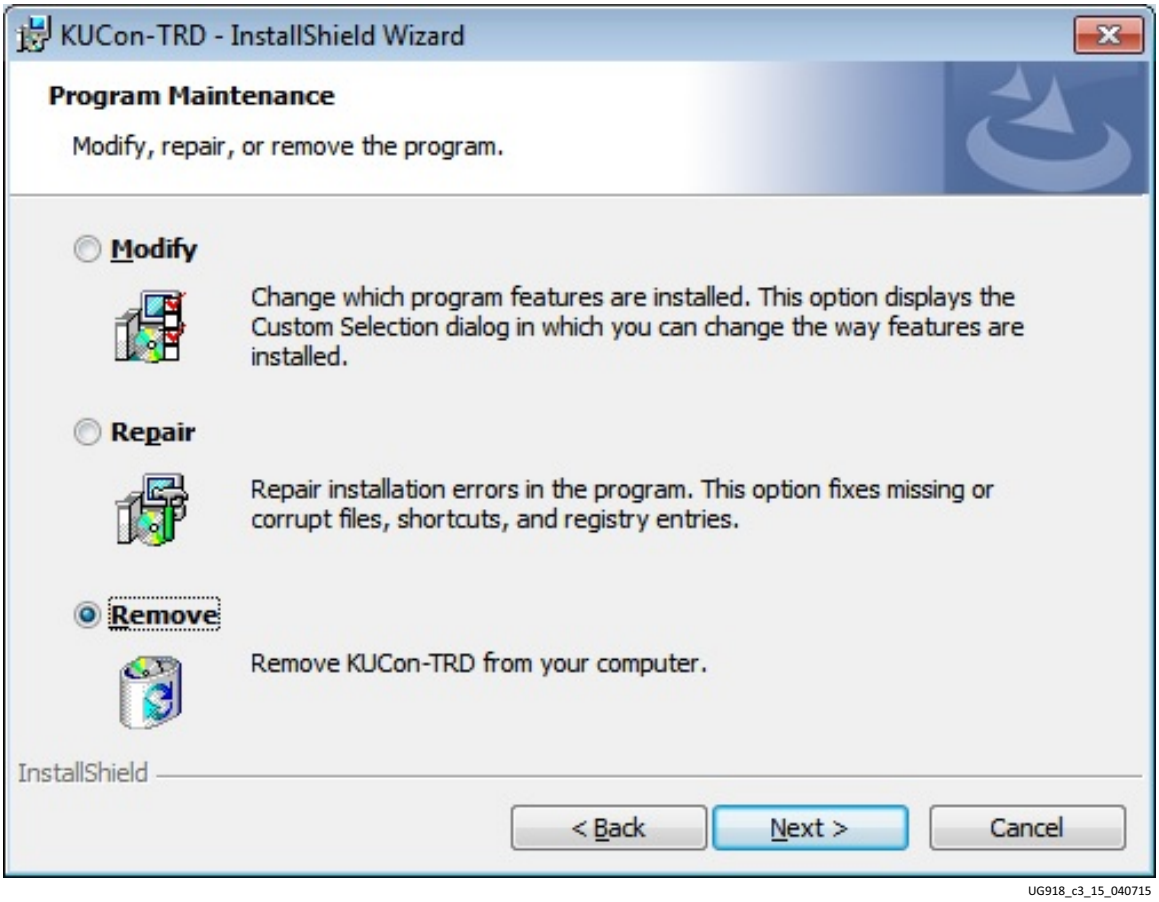


Figure 3-15: **InstallShield Wizard - Remove Drivers Selection**

6. Click **Remove** to remove drivers from the host system as shown in Figure 3-16.

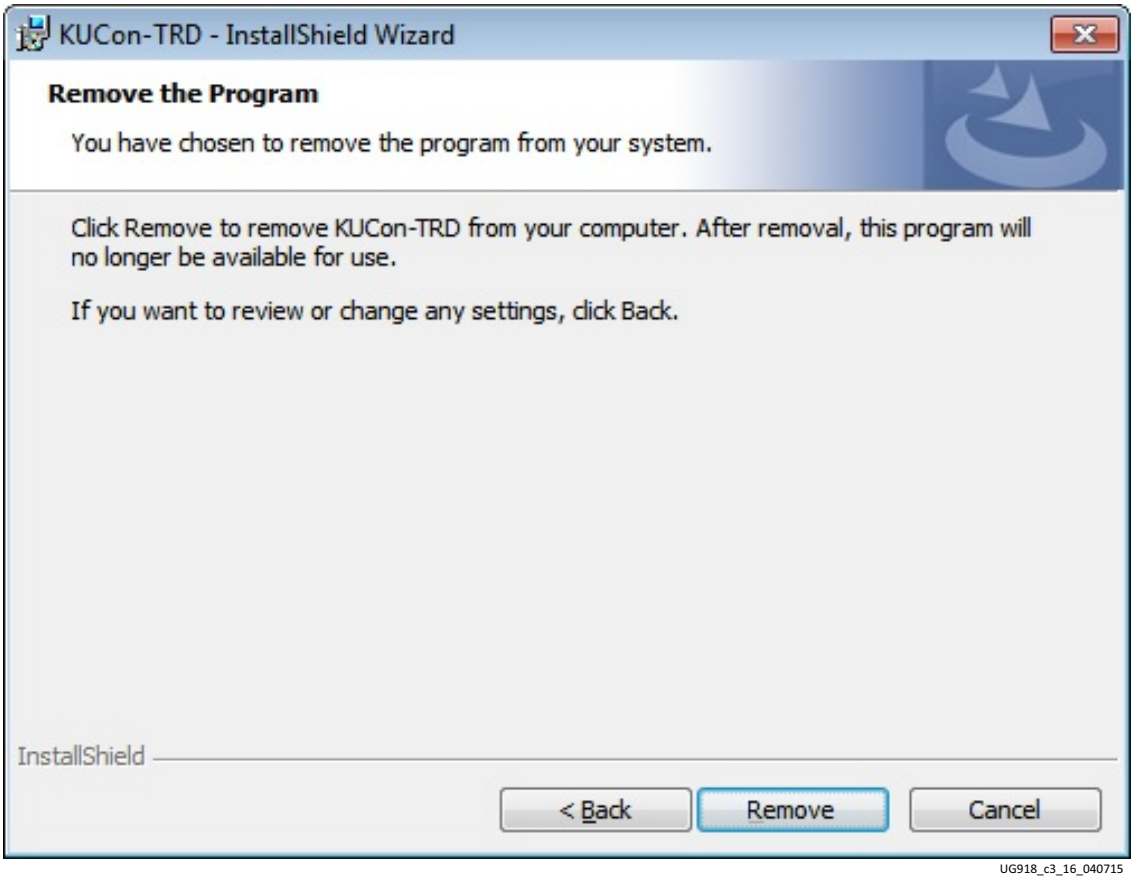


Figure 3-16: **InstallShield Wizard - Remove Drivers**

7. Click **Finish** to exit the wizard (Figure 3-17).

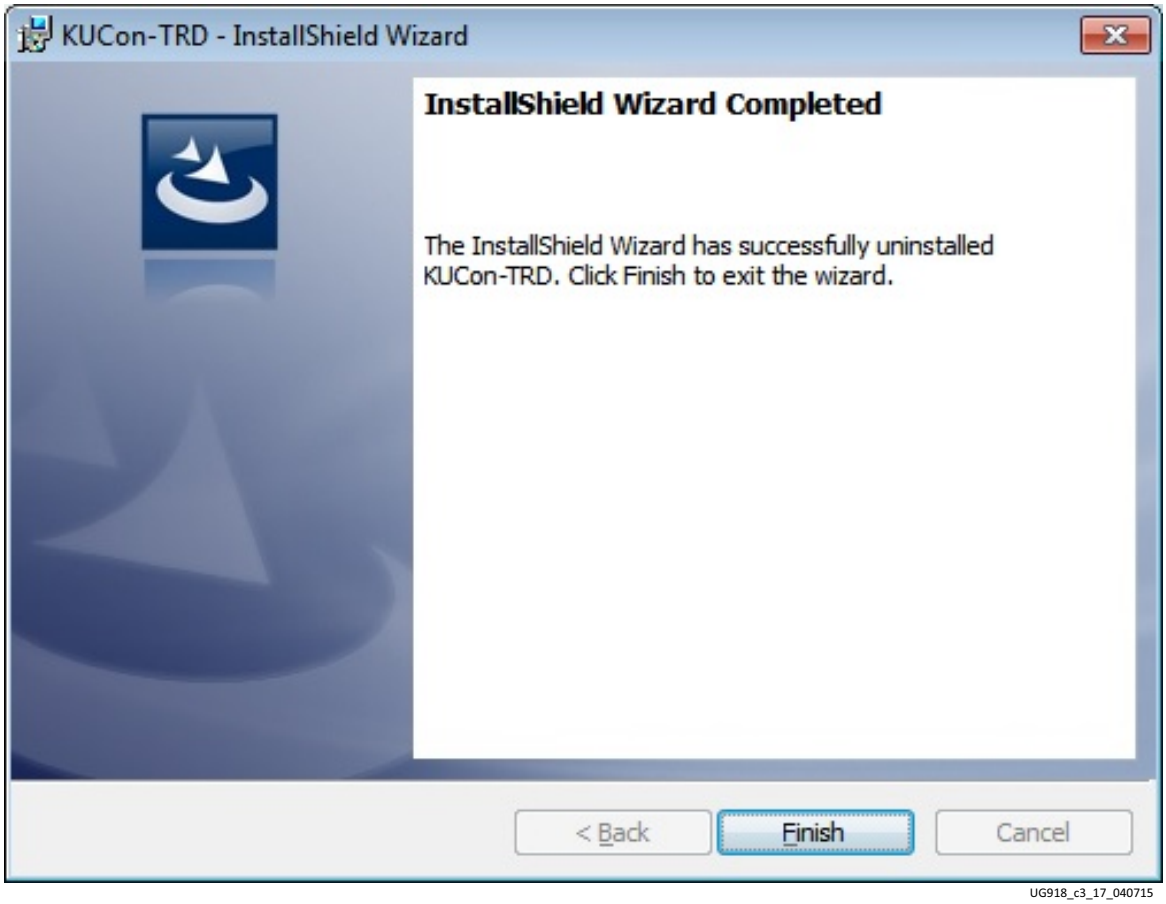


Figure 3-17: **Successful Removal of the Drivers**

Implementing and Simulating the Design

This chapter describes how to implement and simulate the PCI Express Control Plane TRD.

All the steps mentioned in this chapter to run simulation and implementation should be run on the control PC that has Vivado® tools installed.

Note: In Windows, if the path length is more than 260 characters, then design implementation or simulation using the Vivado Design Suite might fail. This is due to a Windows OS limitation. Refer to the KCU105 Evaluation Kit Master Answer Record (AR 63175) for more details.

The time required to build the design and run simulation and implementation can vary from system to system, depending on the system hardware configuration.

Implementing the Base Design

1. Copy the reference design ZIP file to the desired directory on the control PC. Unzip the ZIP file.
2. Open a terminal window on a Linux system with the Vivado environment set up, or open a Vivado tools Tcl shell on a Windows system.
3. Navigate to the `kcu105_control_plane/hardware/vivado/scripts/base` folder.
4. To run the implementation flow, type:

   $ vivado -source trd01_base.tcl

This opens the Vivado Integrated Design Environment (IDE), loads the block diagram, and adds the required top file and Xilinx design constraints (XDC) file to the project. See Figure 4-1.

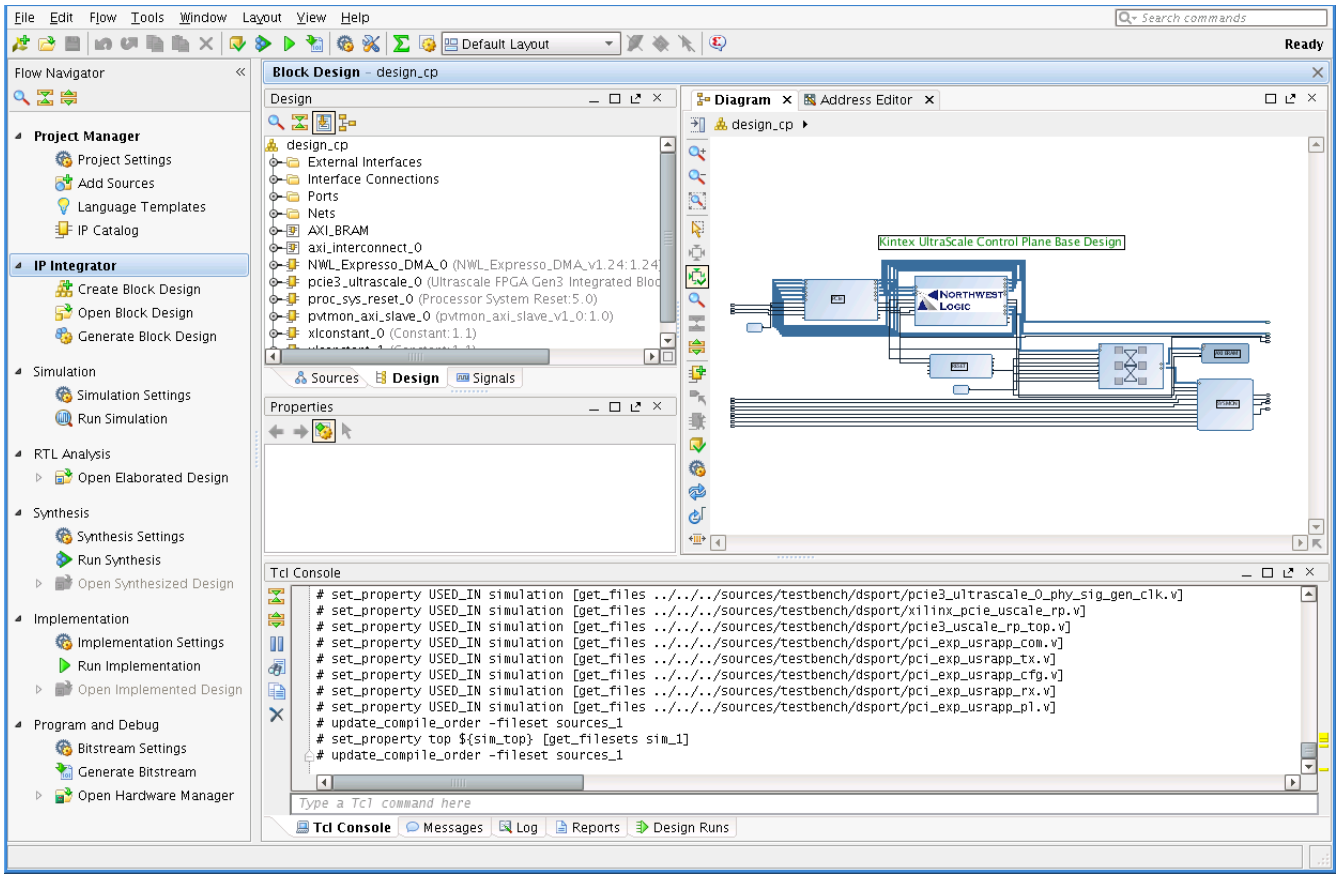


Figure 4-1: **Base Design—Project View**

5. In the Flow Navigator panel, click **Generate Bitstream** (option), which runs synthesis, implementation, and generates a BIT file. See Figure 4-2.

   The generated bitstream can be found under the `kcu105_control_plane/hardware/vivado/runs_base/trd01.runs/impl_1/` directory.

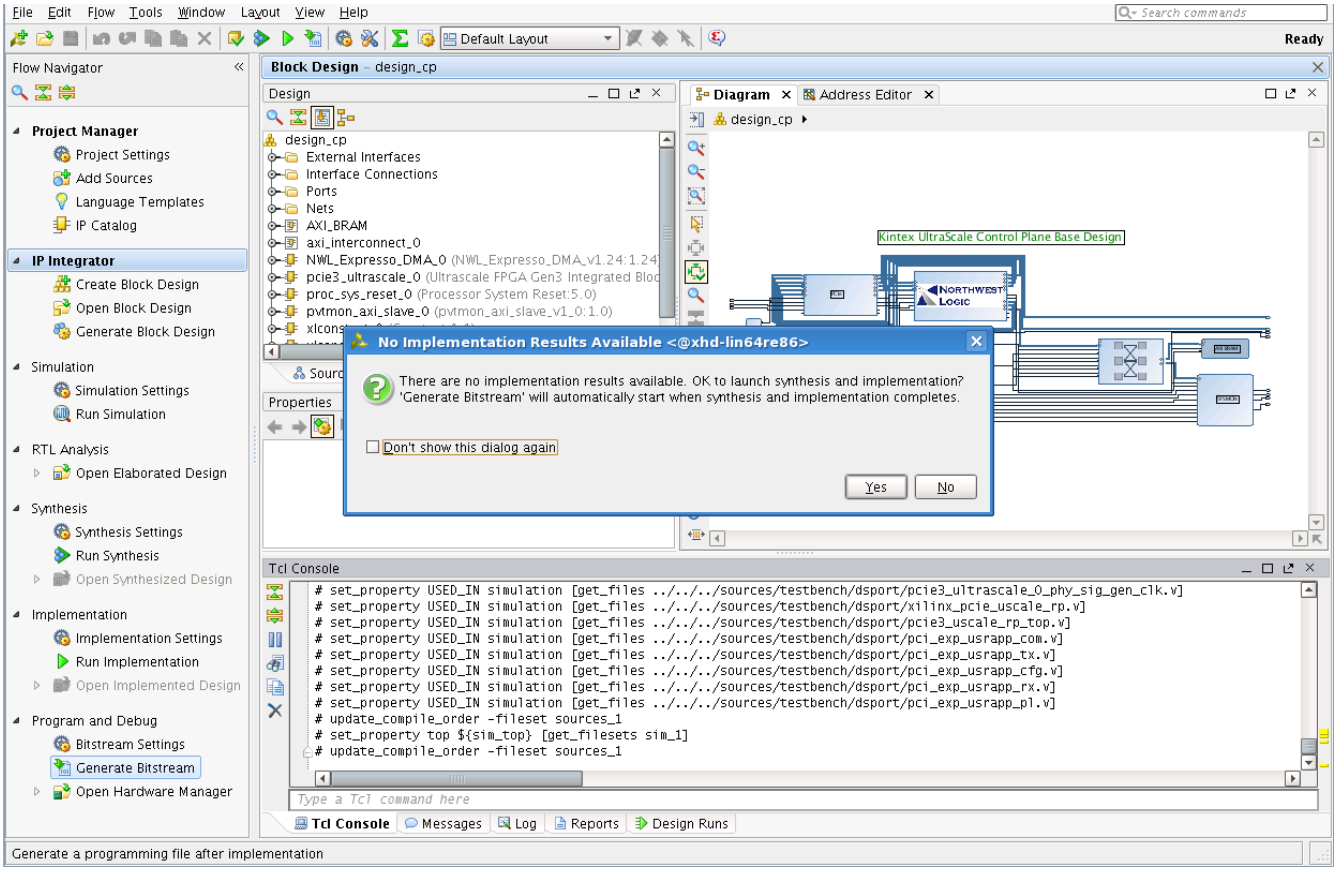


Figure 4-2: **Base Design—Generate Bitstream**

Implementing the User Extension Design

1. Open a terminal window on a Linux system with the Vivado environment set up, or open a Vivado tools Tcl shell on a Windows system.
2. Navigate to the `kcu105_control_plane/hardware/vivado/scripts/user_extn` folder.
3. To run the implementation flow, type:

   $ vivado -source trd01_user_extn.tcl

   This opens the Vivado IDE, loads the block diagram, and adds the required top file and XDC file to the project. See Figure 4-3.

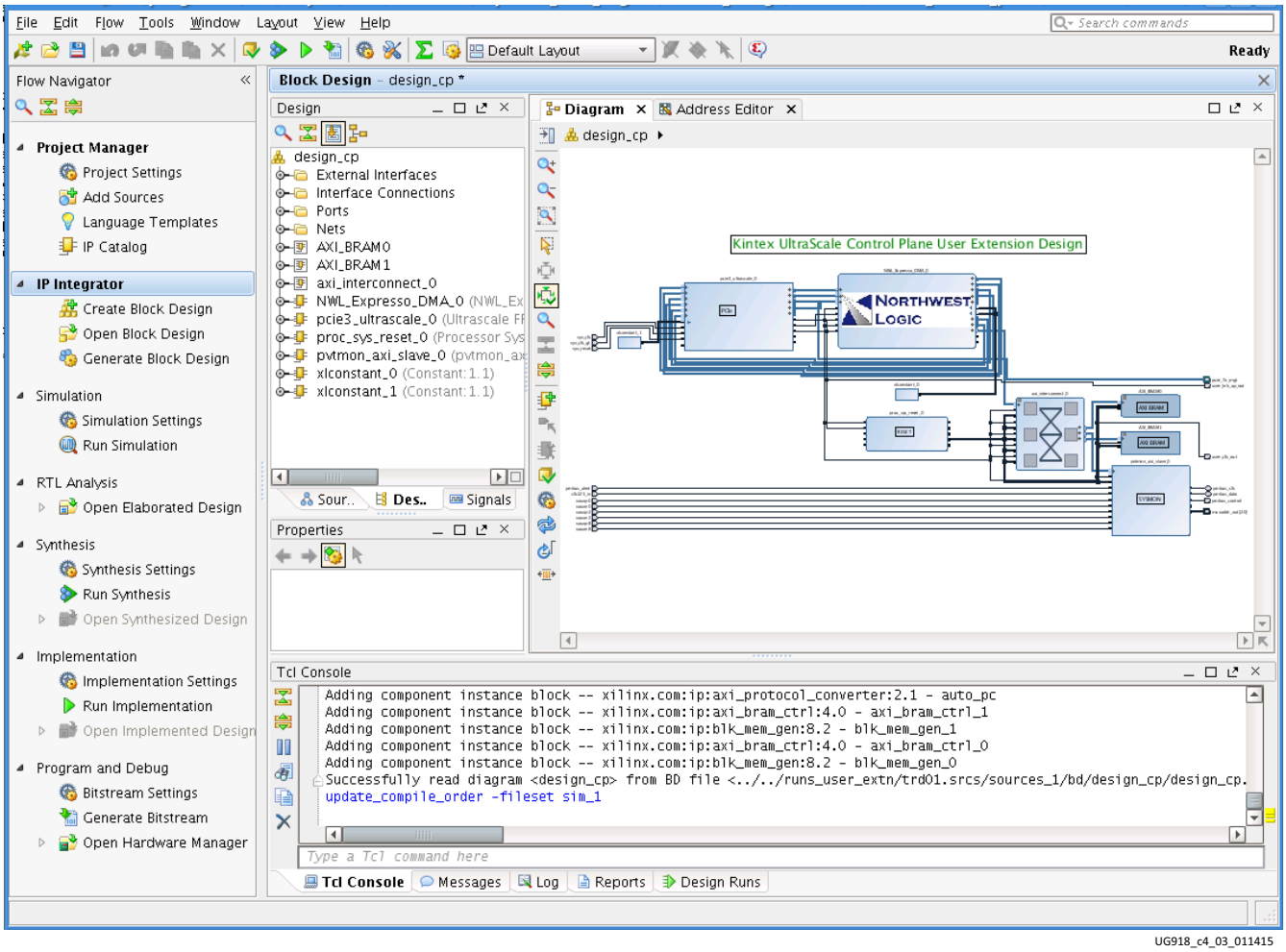


Figure 4-3: **User Extension Design—Project View**

4. In the Flow Navigator panel, click **Generate Bitstream** (option), which runs synthesis, implementation, and generates a BIT file. See Figure 4-4.

   The generated bitstream can be found under the `kcu105_control_plane/hardware/vivado/runs_user_extn/trd01.runs/impl_1/` directory.

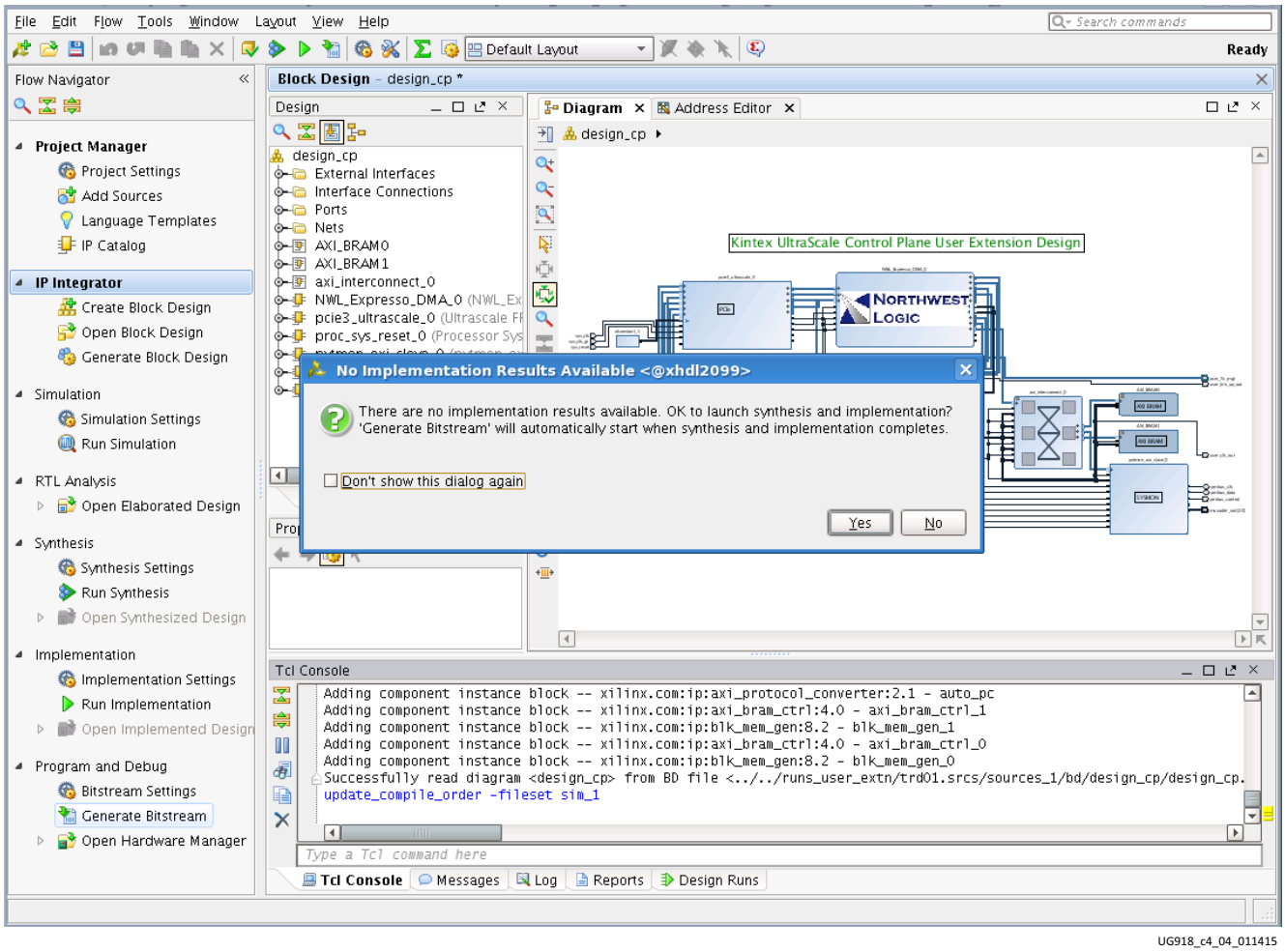


Figure 4-4: **User Extension Design—Generate Bitstream**

Simulating the Base Design

The PCI Express Control Plane TRD can be simulated using the Vivado simulator.

Note: The test bench and the Endpoint PCIe IP block are configured to use PHY Interface for PCI Express (PIPE) mode simulation.

The test bench initializes the bridge, does one double word (DW) write to BAR-mapped address space, reads back from the same address, and compares the data with expected pattern.

Simulation setup is provided only for the base design and not for the pre-built user extension design.

Using the Vivado Simulator

1. Open a terminal window on a Linux system with the Vivado environment set up, or open a Vivado tools Tcl shell on a Windows system.
2. Navigate to the `kcu105_control_plane/hardware/vivado/scripts/base` folder.
3. To run simulation, enter:

   $ vivado -source trd01_base.tcl

This step creates the project and opens the Vivado IDE with target simulator settings set to Vivado Simulator (Figure 4-5).

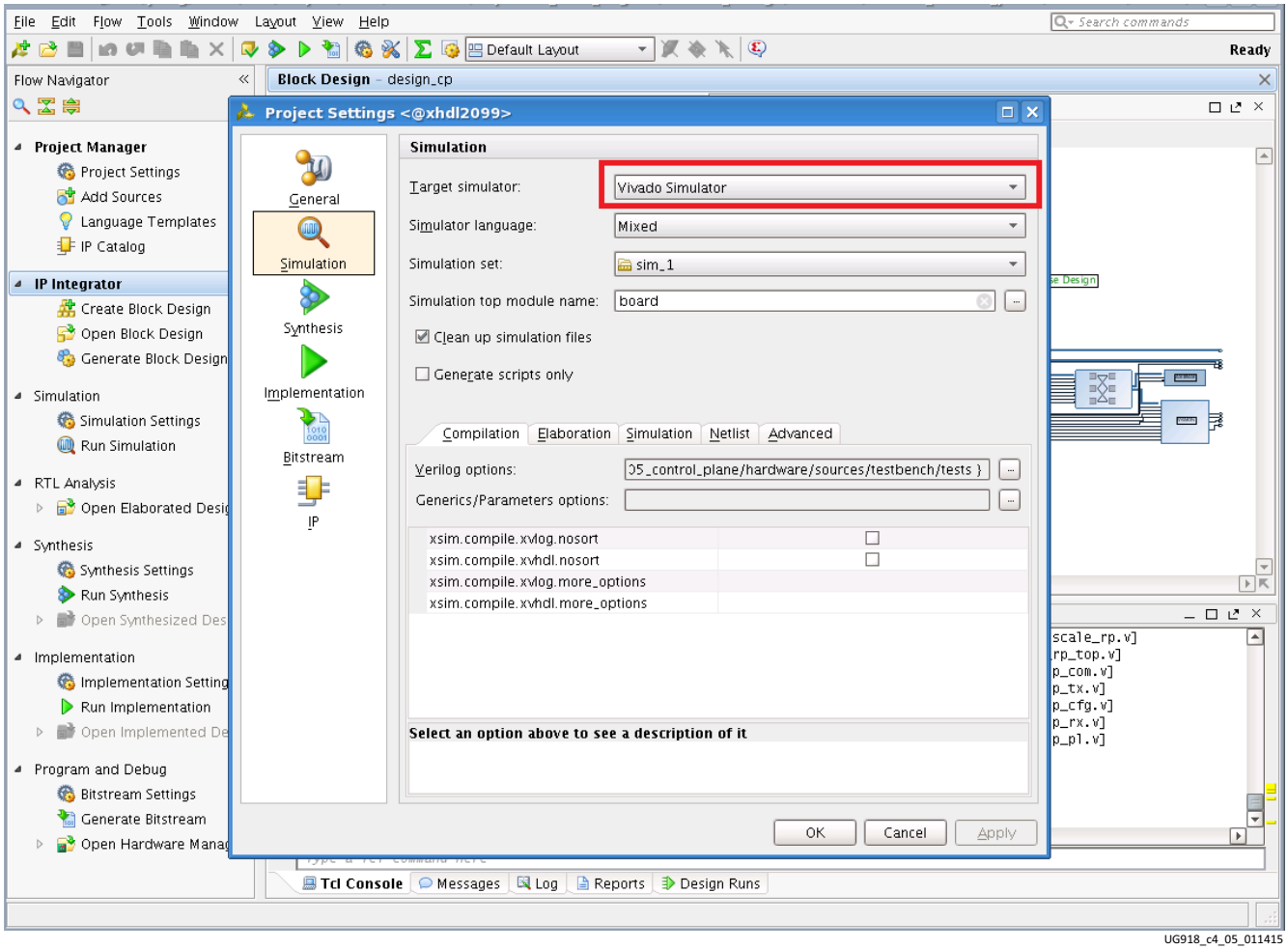


Figure 4-5: **Base Design Project Settings for Simulation**

4. In the Flow Navigator panel, click **Run Simulation** and select **Run Behavioral Simulation** (Figure 4-6).

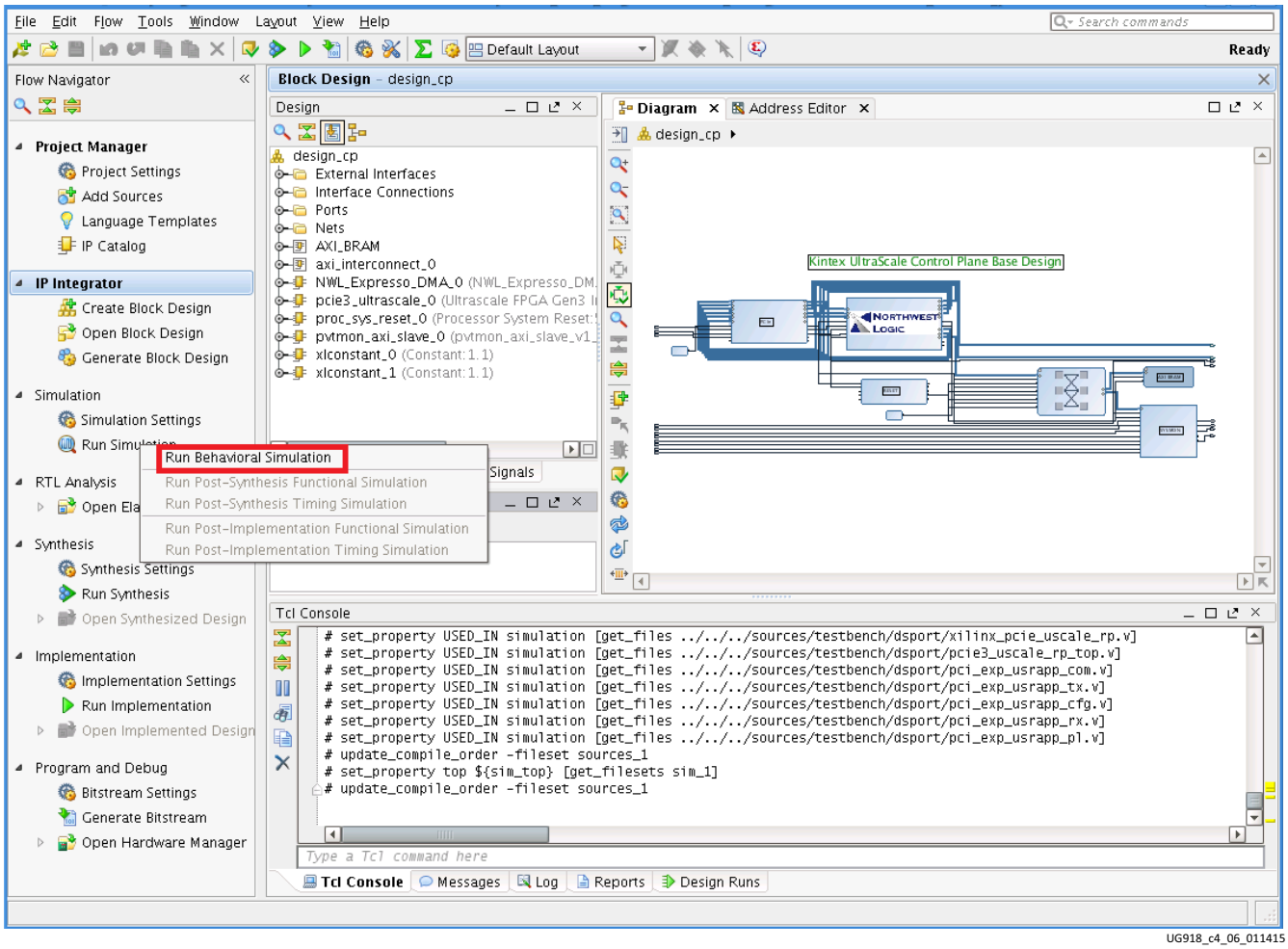


Figure 4-6: **Base Design Behavioral Simulation using the Vivado Simulator**

TRD Details and Modifications

This chapter describes PCI Express Control Plane TRD details. The design details are discussed under hardware and software subsections.

Hardware

The block diagram in Figure 5-1 demonstrates the different hardware design components. Subsequent sections discuss each of the components in detail.

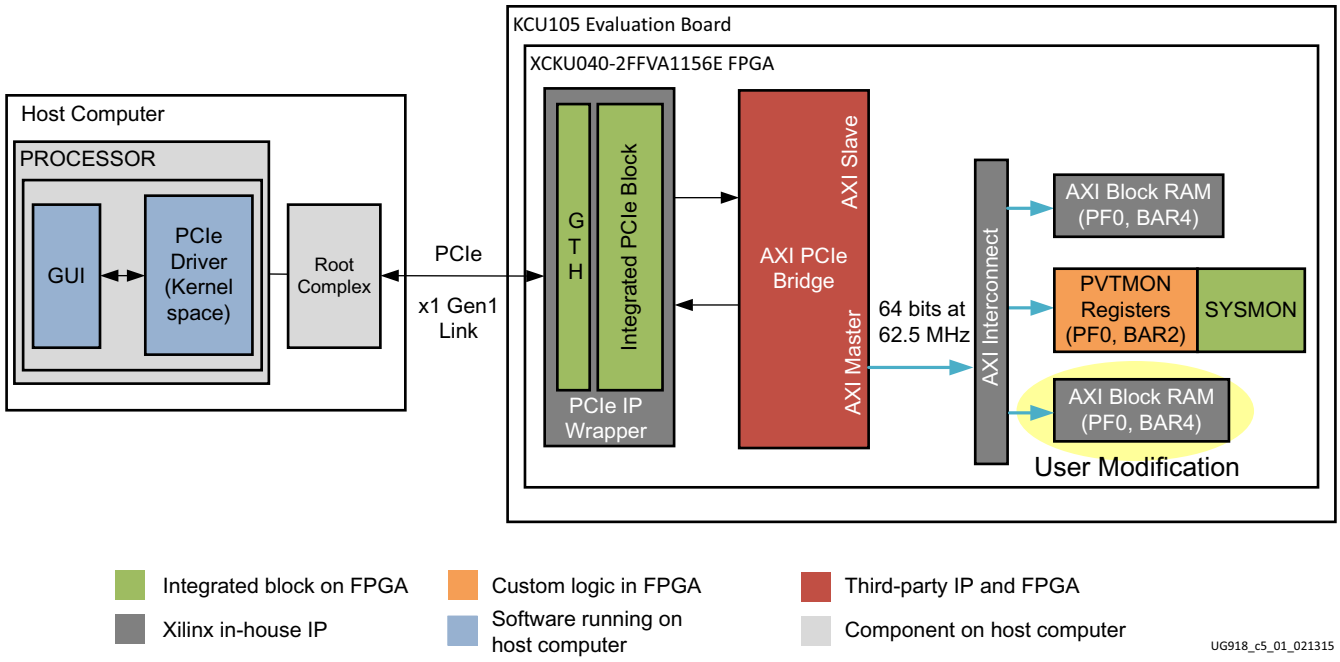


Figure 5-1: **Functional Block Diagram**

Kintex UltraScale PCIe IP

The PCI Express IP for Kintex® UltraScale™ is used in the following configuration:

- Support for x1 Gen1 line rate (2.5 GT/s/lane/direction)
- Support for three 64-bit BARs

See *LogiCORE IP UltraScale FPGAs Gen3 Integrated Block for PCI Express Product Guide* (PG156) [Ref 4] for more information.

Northwest Logic Expresso IP Core

The Expresso IP is from Northwest Logic (NWL). This IP is a protocol conversion unit between PCIe and AXI3. The IP has the following features:

- Converts protocol from PCIe transactions to AXI3 and vice versa
- Support for four ingress translation regions to convert PCIe BAR-mapped transactions to AXI3 domain transactions

Note: The Northwest Logic Expresso IP Core provided with the design is an evaluation version of the IP. It times out in hardware after 12 hours. To obtain a full license of the IP, contact Northwest Logic [Ref 5].

See the Northwest Logic Expresso DMA Bridge Core website to obtain a user guide [Ref 1].

Initialization Steps

This section describes the initialization steps for AXI-PCIe Bridge IP. The AXI-PCIe bridge consumes transactions hitting BAR0 in the Endpoint.

- The bridge registers are accessible from BAR0 + 0x8000.
- During ingress translation initialization:
  - Two ingress translations are enabled (0x800 and 0x820).
  - The address translation maps listed in Table 5-1 are set up.

For example, assume that the PCIe BAR2 physical address is 0x2E000000. A memory read request targeted to address 0x2E000000 is translated to 0x44A00000.

Table 5-1: **Address Translation Maps**

Ingress Source Base	Ingress Destination Base	Comments
BAR2	0x44A00000	PCIe BAR2 mapped to power monitor slave
BAR4	0xC0000000	PCIe (BAR4) mapped to AXI block RAM

- During Bridge register initialization
  - Bridge base low (0x210) is programmed to (BAR0 + 0x8000).
  - Bridge Control register (0x208) is programmed to set the bridge size and enable translation.
- After bridge translation has been enabled, ingress registers can be accessed with bridge base + 0x800.

AXI Block RAM Controller

The AXI block RAM controller provides block RAM with an AXI4 memory-mapped interface. This behaves as a register file in this design to which BAR-mapped transactions are targeted.

See *LogiCORE IP AXI Block RAM (BRAM) Controller Product Guide* (PG078) [Ref 6] for more details.

AXI Interconnect

AXI Interconnect is used to connect the various IPs together in a memory-mapped system. The interconnect is responsible for:

- Converting AXI3 transactions from AXI-PCIe bridge into AXI4 transactions for various slaves
- Decoding address to target appropriate slave

See *LogiCORE IP AXI Interconnect Product Guide* (PG059) [Ref 7] for more details.

Power and Temperature Monitoring

The design uses the SYSMON block to provide system power and die temperature monitoring capabilities (Figure 5-2).

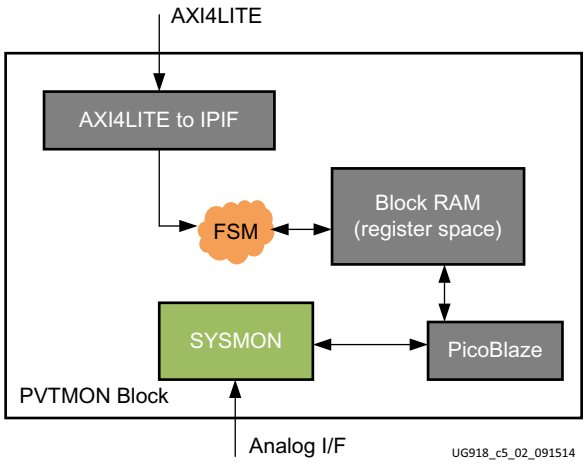


Figure 5-2: **Power and Temperature Monitoring Block**

The System Monitor block (17 channel, 200 kSPS) provides analog-to-digital conversion and monitoring capabilities. It enables reading of voltage and current on different power supply rails (supported on the KCU105 board) which are then used to calculate power.

A lightweight PicoBlaze™ controller is used to set up the SYSMON registers in continuous sequence mode and read various rail data periodically. The output from the PicoBlaze controller is made available in block RAM (register space), and this can be accessed over PCIe through the BAR-mapped region.

The AXI4-Lite IPIF core is used in the design and the interface logic between the block RAM and the AXI4-Lite IPIF reads the power and temperature monitor registers from block RAM. Providing an AXI4-Lite slave interface brings the additional flexibility of using the module in other designs.

See *UltraScale Architecture System Monitor User Guide* (UG580) [Ref 8] for more details.

Software

Software Architecture

The software component of the TRD framework is comprised of a kernel-space driver module as shown in Figure 5-3. It also contains a GUI that controls the design operation. The software comprises building blocks designed with scalability in mind. Additional user-space applications can be designed with the existing blocks.

This section explains the software architecture of the PCI Express Control Plane TRD. The design consists of user and kernel components.

User-Space Components

A user-space component consists of a GUI block.

Graphical User Interface

The user-space GUI is a Java-based GUI that provides the following features:

- GUI management of the driver and device
  - In Linux, the GUI installs the selected design mode drivers and can configure and control device and test parameters.
  - In Windows, the GUI can configure and control device and test parameters.
- GUI front end graphical display of statistics collected from the underlying driver through the driver interface

This design demonstrates the use of PCIe in control plane applications. A simple kernel driver on a host computer demonstrates BAR-mapped single double word (DW) register transfers.

Apart from generic GUI functionality described earlier, the GUI allows you to read from or write to BAR-mapped registers in hardware and display them in the GUI window.

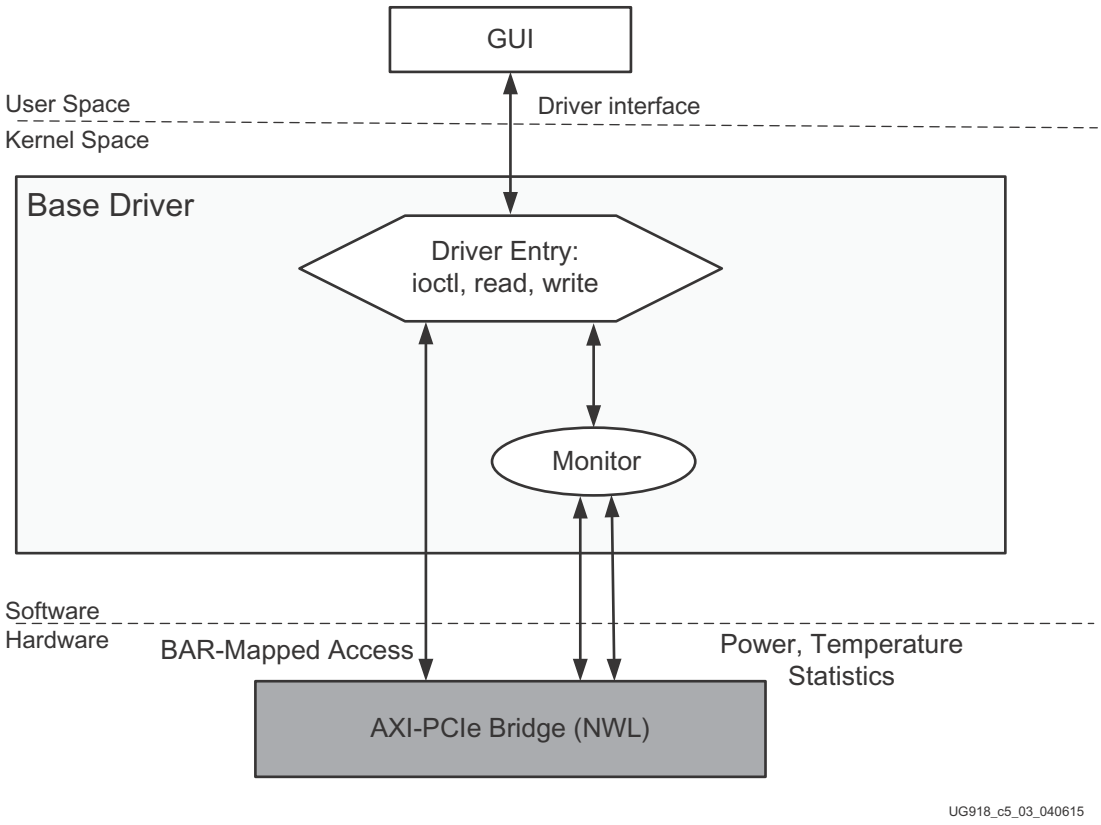


Figure 5-3: **PCI Express Control Plane TRD Software Design**

Software Data Flow

This section summarizes the software data flow of the PCI Express Control Plane TRD.

Transmit Path

1. The GUI opens the driver interface for read/write functionality.
2. The GUI issues WRITE system calls for writing into BAR-mapped registers based on your input.
3. The driver writes appropriate values into BAR-mapped registers.

Receive Path

1. The GUI issues a READ system call to read BAR-mapped registers based on your input.
2. The Character driver reads appropriate BAR-mapped registers and conveys the readings to the GUI.

3. The GUI displays the read BAR-mapped register information.

TRD Modifications

This section describes TRD modifications.

Pre-built Modification: Adding Another Block RAM Controller

This section describes the pre-built modification shipped with the TRD. It describes how to add another AXI block RAM controller to the design and set up ingress translations through BAR4 to access this memory space.

Rebuilding Hardware

A pre-built design script is provided for the user modification design which can be run to generate bitstream with an additional AXI block RAM controller added. The additional block RAM is mapped to AXI address `0xD000_0000`. The steps needed to build the user modification design are described in Chapter 4, Implementing and Simulating the Design.

Software Modification

IMPORTANT: *This software modification section applies only to Linux; not Windows.*

In the software driver, access to newly added block RAM can be added as follows.

In the file `software/linux_driver_app/driver/ctrlplane/xpcie.c`, under the InitBridge function, the line can be changed as shown:

```
//- Program DST address to be AXI domain address for BRAM Controller

  XIo_Out32((bar0_addr +REG_BRDG_BASE + REG_INGR_AXI_BASE + SECOND_TRANS
+OFFSET_INGR_AXI_DST_LO ), 0xC0000000);

//- Program DST address to be AXI domain address for BRAM Controller

  XIo_Out32((bar0_addr +REG_BRDG_BASE + REG_INGR_AXI_BASE + SECOND_TRANS
+OFFSET_INGR_AXI_DST_LO ), 0xD0000000);
```

This maps the newly added block RAM controller to BAR4. With this minor change, the same GUI can be used for read/write access.

To enable read access to both block RAM controllers, an additional ingress translation aperture can be mapped and the ReadUserReg function in the driver can be used to access those registers. The display from the software driver can be seen in the system dmesg log.

The ability to read multiple user registers or block RAM controllers is not supported by the GUI, and the aperture size is currently limited to 4K.

Directory Structure

The directory structure in the PCI Express Control Plane TRD ZIP file is shown in Figure A-1 and described in Table A-1.

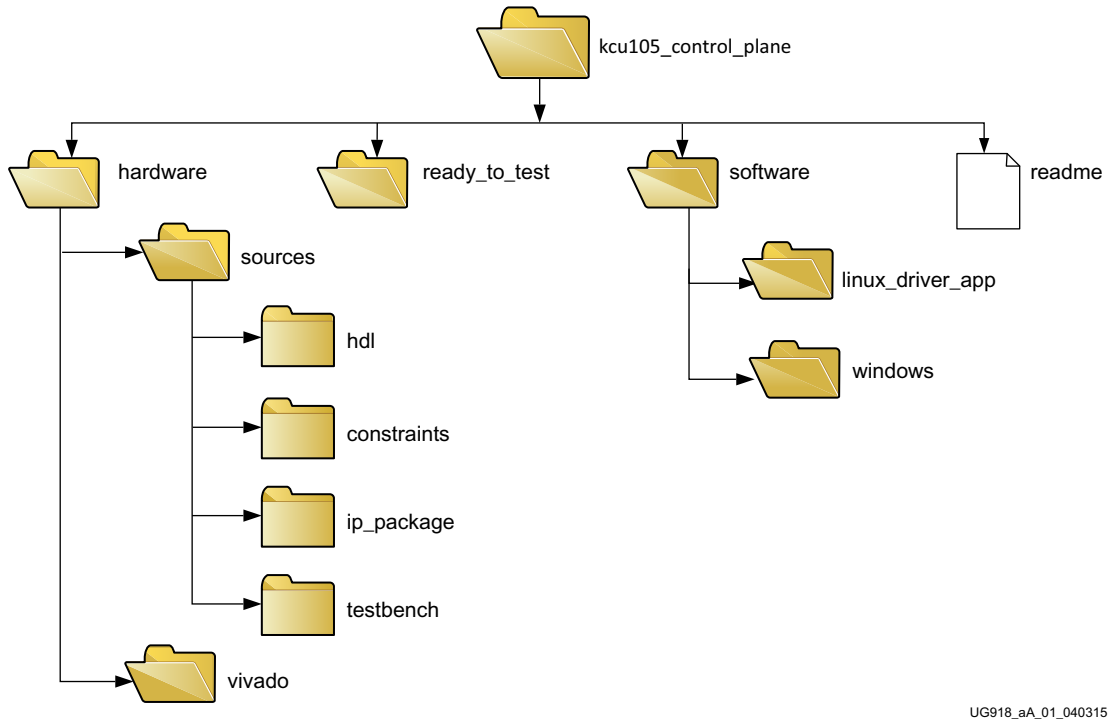


Figure A-1: **Directory Structure**

For a detailed description of each folder, see the Readme file that accompanies the TRD ZIP file.

Table A-1: **Directory Structure Details**

Folder	Description
readme	A TXT file that includes revision history information, steps to implement and simulate the design, required Vivado® tool software version, and known limitations of the design (if any).
hardware	Contains hardware design deliverables
sources	
hdl	Contains HDL files
constraints	Contains constraint files
ip_package	Contains custom IP packages
testbench	Contains test bench files
vivado	Contains scripts to create a Vivado Design Suite project and outputs of Vivado runs
ready to test	Contains the BIT file to program the KCU105 PCI Express® Control Plane application
software linux_driver_app windows	Contains software design deliverables for Linux and Windows

Recommended Practices and Troubleshooting in Windows

Recommended Practices

1. Make a backup of the system image and files using the Backup and Restore utility of the Windows 7 operating system before installing reference design drivers. (As a precautionary measure, a fresh installation of the Windows 7 OS is recommended for testing the reference design.)

Troubleshooting

Problem: The TRD Setup screen of the GUI does not detect the board.

Corrective Actions:

1. If the GUI does not detect the board, open **Device Manager** and see if the drivers are loaded under **Xilinx PCI Express Device**.
2. If the drivers are not loaded, check the PCIe Link Up LED on the board (see Figure 3-8).
3. If the drivers are loaded but the GUI is not detecting the board, remove non-present devices from Device Manager using the following steps.
   a. Open a command prompt with Administrator privileges.
   b. At the command prompt, enter the following bold text:

      set devmgr_show_nonpresent_devices=1

      start devmgmt.msc

   c. Click the **View** menu and select **Show hidden devices** on the Device Manager window.
   d. Non-present devices are indicated by a lighter shade of text.
   e. Look for all the Non Present/Hidden devices. Right-click each one, and select **Uninstall**. Remove the driver if prompted for it.

4. Invoke the GUI of the reference design and check if it detects the board.

Additional Resources and Legal Notices

Xilinx Resources

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

For continual updates, add the Answer Record to your myAlerts.

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 website

KCU105 Evaluation Kit documentation

KCU105 Evaluation Kit Master Answer Record (AR 63175)

These documents and sites provide supplemental material useful with this guide:

1. Northwest Logic Expresso DMA Bridge Core
2. *Vivado Design Suite User Guide Release Notes, Installation, and Licensing* (UG973)
3. *Kintex UltraScale FPGA KCU105 Evaluation Board User Guide* (UG917)
4. *LogiCORE IP UltraScale FPGAs Gen3 Integrated Block for PCI Express Product Guide* (PG156)
5. Northwest Logic PCI Express Solution

6. *LogiCORE IP AXI Block RAM (BRAM) Controller Product Guide* (PG078)
7. *LogiCORE IP AXI Interconnect Product Guide* (PG059)
8. *UltraScale Architecture System Monitor User Guide* (UG580)

Please Read: Important Legal Notices