AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Hardware Tutorial (ISE Design Suite 14.5)

UG914 (v2.0) April 23, 2013

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/04/2012	1.0	Initial Xilinx Release.
11/05/2012	1.1	Changed "system_top - STRUCTURE" to "Design Sources and system_top -STRUCTURE" under Create the Hardware Platform and Customize the Embedded Hardware Platform. Updated Figure 1-12, Figure 1-21, and Figure 1-22. Added step a under Export the BIST Hardware Platform to SDK. Changed "enter the new connection name" to "select from the drop down menu" in step 6 under Connect the Ports.
04/23/2013	2.0	Updated for ISE® Design Suite 14.5. Changed **`board_test_app_console.elf`** to **`board_test_app_Console.elf`** in step 9 under Executing the System, page 9. Under Connect the Ports, page 27, step 6 changed. Replaced Figure 1-5, Figure 1-6, Figure 1-7, Figure 1-10, Figure 1-12, Figure 1-14, Figure 1-15, Figure 1-17, Figure 1-18, Figure 1-19, Figure 1-20, Figure 1-21, and Figure 1-22. Enhanced Appendix A, Additional Resources.

Table of Contents

Revision History 2

Chapter 1: KC705 Embedded Kit MicroBlaze Processor Subsystem Hardware Tutorial

Introduction 5
Hardware and Software Requirements 5
Prerequisites 6
System Overview 6
Included Files and Systems 6
Executing the System 9
Hardware Design Flow 12
Summary 31
Next Steps 31

Appendix A: Additional Resources

Xilinx Resources 33
Solution Centers 33
Further Resources 33
References 34

XILINX®

Chapter 1

KC705 Embedded Kit MicroBlaze Processor Subsystem Hardware Tutorial

Introduction

The Kintex®-7 FPGA embedded kit conveniently delivers the key components of the Xilinx® Embedded Targeted Design Platform (TDP) required for developing embedded software and hardware in a wide range of applications in the broadcast, industrial, medical, and aerospace and defense markets. Hardware designers now have immediate access to a pre-integrated MicroBlaze™ processor subsystem that includes the most commonly used peripheral IP cores, enabling the designers to begin at once developing their custom logic.

This tutorial guides the designer through the three steps required to examine, modify and test the MicroBlaze processor subsystem with the KC705 evaluation board:

1. Loading and executing a design from pre-built bitstream and ELF files
2. Examining and rebuilding a design
3. Adding IP from the Xilinx IP catalog to an embedded system

It is advisable to read UG913, *Getting Started with the Kintex-7 FPGA KC705 Embedded Kit* before going through this tutorial. Additional information related to EDK design flow can be found in UG683, *EDK Concepts, Tools, and Techniques*.

Hardware and Software Requirements

Included with the Kintex-7 FPGA embedded kit:

- Xilinx KC705 evaluation board
- One USB Type-A to Mini-B cable
- One USB Type-A to Micro-B cable
- Ethernet cable
- ISE® Design Suite: Embedded Edition which includes:
  - Integrated Software Environment (ISE)
  - Embedded Development Kit (EDK)
  - Software Development Kit (SDK)

Additional Requirements:

- Host PC with serial communications utility program (e.g., Tera Term)

Prerequisites

Prerequisites required to run the basic tutorial:

- KC705 embedded kit MicroBlaze processor subsystem
- Proper hardware setup and software installation, as described in UG913, *Getting Started with the Kintex-7 FPGA KC705 Embedded Kit*
- Familiarity with the KC705 embedded kit MicroBlaze processor subsystem, documented in DS669, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Data Sheet*
- Familiarity with UG683, *EDK Concepts, Tools, and Techniques*, specifically these sections:
  - Introduction
  - Using Xilinx Platform Studio
  - Working with the Embedded Platform
- General knowledge of FPGAs, digital design concepts, and microprocessors
- Basic familiarity with the Xilinx Design Tools
- Basic VHDL/Verilog knowledge

System Overview

This tutorial is based on the KC705 embedded kit MicroBlaze processor subsystem. Readers are encouraged to refer to DS669, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Data Sheet* during the execution of this tutorial.

Included Files and Systems

The `Tutorial_Sandbox` directory provides the working area for this tutorial. It is part of the KC705 embedded kit. The hardware portion of the Tutorial Sandbox contains two subsystems: Built-In Self Test (BIST) and the partially built video demo system. The video demo system consists of video IP cores that are added to the system to develop a complete, functioning video demo. Figure 1-1 shows the structure of the `Tutorial_Sandbox` directory.

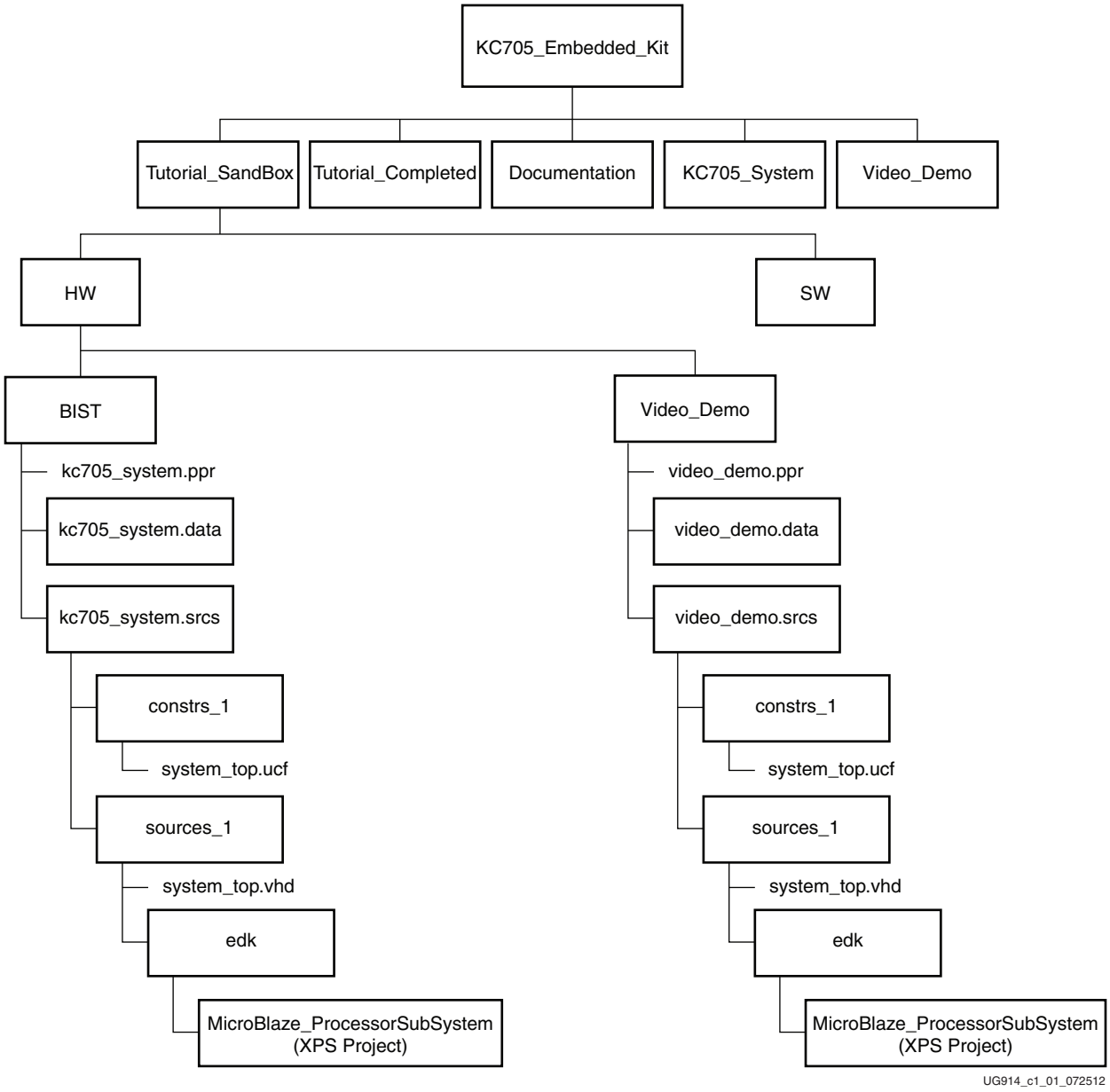


Figure 1-1: **Tutorial Sandbox Directory Structure**

The completed tutorial resides in the `Tutorial_Completed` directory, the structure of which is shown in Figure 1-2.

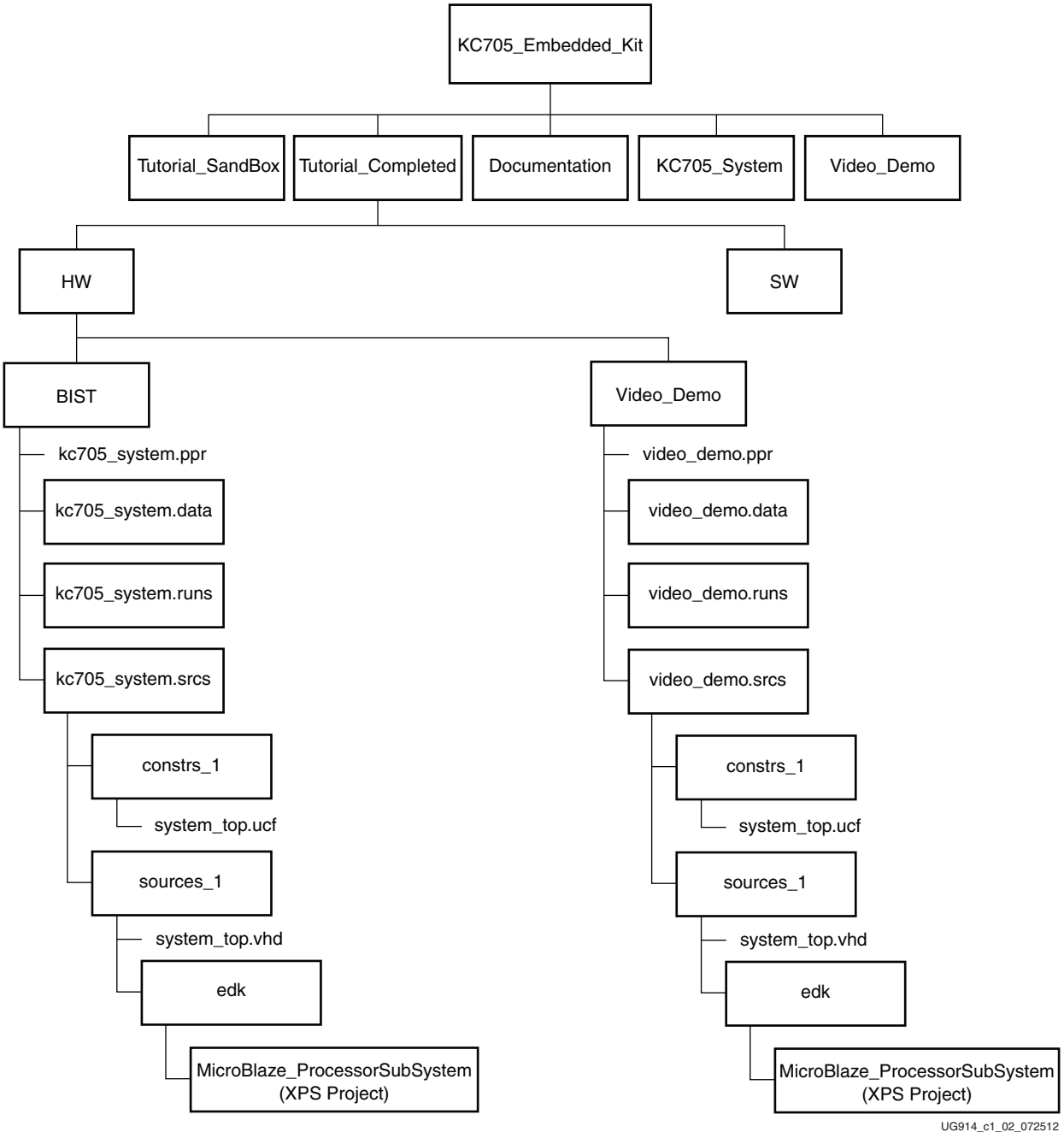


Figure 1-2: **Tutorial Completed Directory Structure**

Executing the System

The `ready_for_download` directory contains the pre-built bitstream and ELF files for the KC705 embedded kit MicroBlaze processor BIST subsystem. The directory structure of the `KC705_System` is shown in DS669, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Data Sheet*.

The pre-built bitstream and ELF files are used to verify that the software is properly installed and that the board is set up correctly. To execute the system using these files:

1. Connect a USB Type-A to Micro-B cable between the KC705 evaluation board USB JTAG connector and the host computer.
2. Verify that a USB Type-A to Mini-B cable is properly connected to the KC705 evaluation board USB UART connection and the host computer.
3. If the board is not already powered on, power on the KC705 board.
4. Open and configure a serial communications terminal utility program with the settings shown in Figure 1-3.

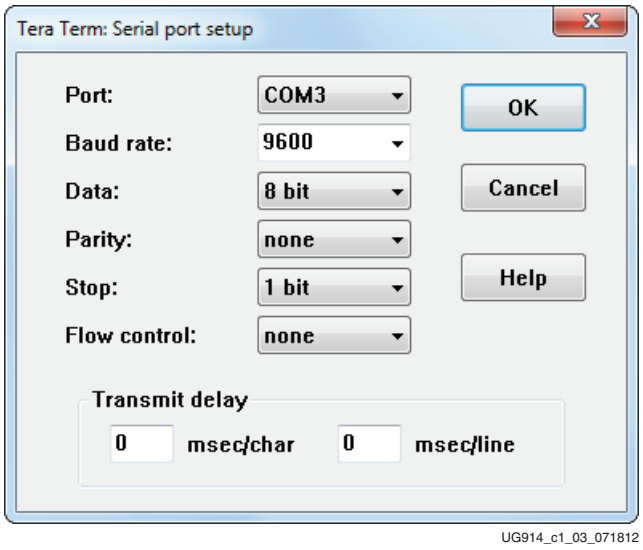


Figure 1-3: **Serial Communications Utility Configuration**

Note: A procedure for setting up a UART connection is provided in Appendix B of UG913, *Getting Started with the Kintex-7 FPGA KC705 Embedded Kit*.

5. Open a command shell with the ISE Design tool and EDK environment settings. Refer to Appendix C of UG913 for ISE tool chain installation and licensing help. If necessary, set the environment variables by running the `settings32.bat` script file located in the `Xilinx` directory of the ISE Design Suite installation area.
6. At the command prompt, enter the appropriate command:

   On a Windows XP system:

   ```
   C:\Xilinx\14.x\ISE_DS\settings32.bat
   ```

   On Windows 7 system:

   ```
   C:\Xilinx\14.x\ISE_DS\settings64.bat
   ```

7. Execute these commands to download the design and connect to the MicroBlaze processor:

```
$ cd KC705_Embedded_Kit/KC705_System/ready_for_download
$ xmd
XMD% fpga -f download.bit
```

Note: This command downloads the hardware bitstream into the FPGA but does not download the software application.

```
XMD% connect mb mdm
```

Note: This command connects to the MicroBlaze processor debug module.

8. XMD allows low-level visibility into the design. There are several useful XMD commands to allow for the peeking and poking of registers and memory locations within the system. For example, writes to and reads from the internal block RAM are accomplished as follows:

```
XMD% mwr 0xC0000000 0xDEADBEEF
```

The value `0xDEADBEEF` is written to location `0xC0000000`.

```
XMD% mrd 0xC0000000
```

The value `0xDEADBEEF` should be returned:

```
C0000000: DEADBEEF
```

As an example of register access, write to the GPIO register which outputs data to the 8-bit GPIO LEDs on the KC705 board:

```
XMD% mwr 0x40600000 0xAA
```

The value written to this register is now reflected in the GPIO LEDs. The next write changes the LED display:

```
XMD% mwr 0x40600000 0x55
```

A read from this register always returns `0x00000000`, because this is an output-only register.

Note: Set all of the DIP switches labeled GPIO DIP SW (SW11) to the ON position. Then read from the GPIO register which inputs data from the DIP switches:

```
XMD% mrd 0x40700000
```

The value `0x0000000F` should be returned:

```
40700000: 0000000F
```

Change the DIP switch settings and verify that the correct value is read from the GPIO register.

Note: Switch 1 is the most significant bit.

For more information about XMD and the commands available within XMD, see UG111, *Embedded System Tools Reference Manual*, Xilinx Microprocessor Debugger (XMD).

9. To download and execute the test software application, enter these commands at the XMD command prompt:

```
XMD% dow board_test_app_Console.elf
XMD% con
```

Figure 1-4 shows the resulting output for the KC705 board test application on the serial communication terminal utility program.

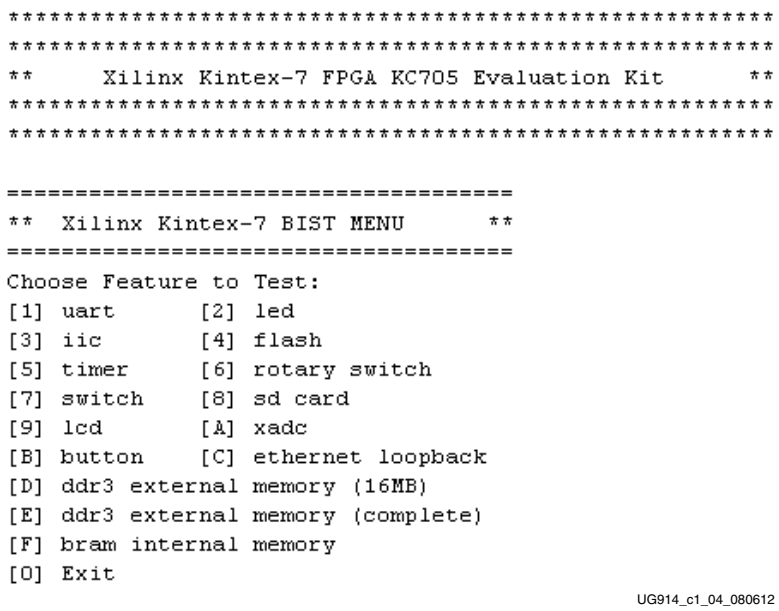


Figure 1-4: **Board Test Application Menu**

10. Execute the chosen tests and then select **0** to exit. Details about the test options can be found in the UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*.
11. When software execution is complete, enter these commands:

```
XMD% stop

XMD% rst

XMD% exit
```

Hardware Design Flow

The design flow for creating the MicroBlaze processor subsystem BIST and modifying the system to include the benchmarking cores requires these steps:

1. Create the Hardware Platform
2. Export the BIST Hardware Platform to SDK
3. Customize the Embedded Hardware Platform
4. Implement and Test the Design

The BIST system is used to demonstrate execution of steps one and two, and the partially built video demo system is used to demonstrate execution of steps three and four.

Create the Hardware Platform

Open the Project

1. Start the PlanAhead™ tool.

   On a Windows system:

   Select **Start > All Programs > Xilinx Design Tools > ISE Design Suite > PlanAhead**

   On a Linux system:

   Enter **`planAhead`** at the command prompt.
2. If necessary, close out the previous project by selecting **File > Close Project**.
3. In Project Commands, select **Open Project...**
4. Browse to `KC705_Embedded_Kit/Tutorial_Sandbox/HW/BIST` and select **kc705_system.ppr**. Click **Open**.
5. In the Project Manager/Sources section, expand Design Sources and system_top -STRUCTURE and double click **system_i - system (system.xmp)**. This invokes XPS for the XPS subsystem.

Figure 1-5 shows the block diagram of the KC705 embedded kit BIST system.

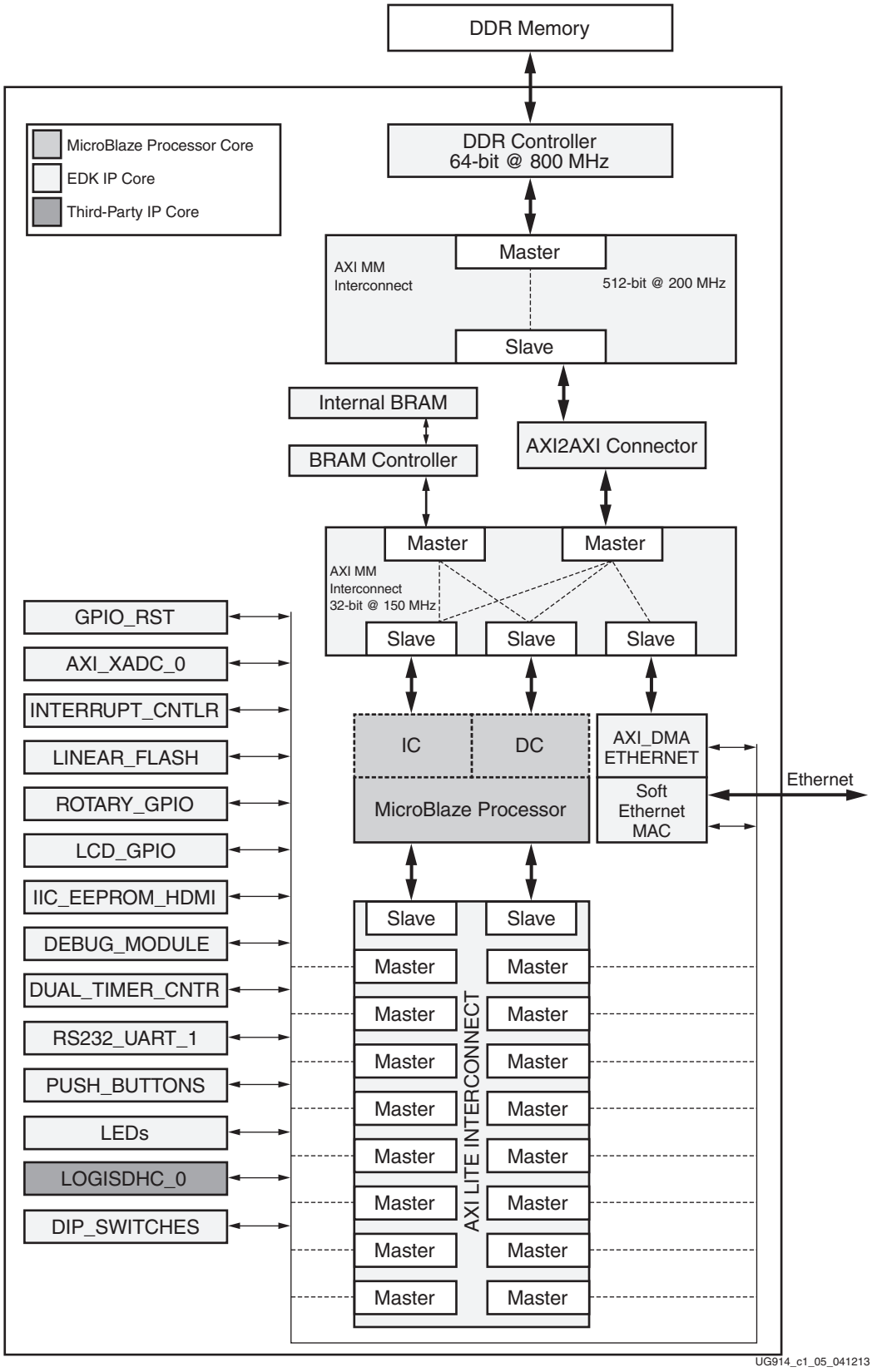


Figure 1-5: **BIST System**

Examine the System

The System Assembly View (SAV) of the design provides visibility into the MicroBlaze processor subsystem bus interfaces, port connections, and address map.

Note: Refer to UG683, *EDK Concepts, Tools, and Techniques* for additional details or instructions pertaining to any of the steps outlined in this section of the tutorial.

Examine Bus Interfaces

1. To examine the bus structure of the MicroBlaze processor subsystem, select the System Assembly View tab below the workspace and then select the Bus Interfaces tab at the top edge of the workspace as shown in Figure 1-6.

   For MicroBlaze processor masters (Instruction Cache (IC), Data Cache (DC), and Data Port (DP)), both AXI_MM and AXI_Lite interconnects are used. Two Local Memory Buses (LMB) are used by the MicroBlaze processor to interface to the local BRAM memory for the processor. These interfaces can be seen by clicking the **+** to the left of the `microblaze_0` component. (See: Figure 1-6).

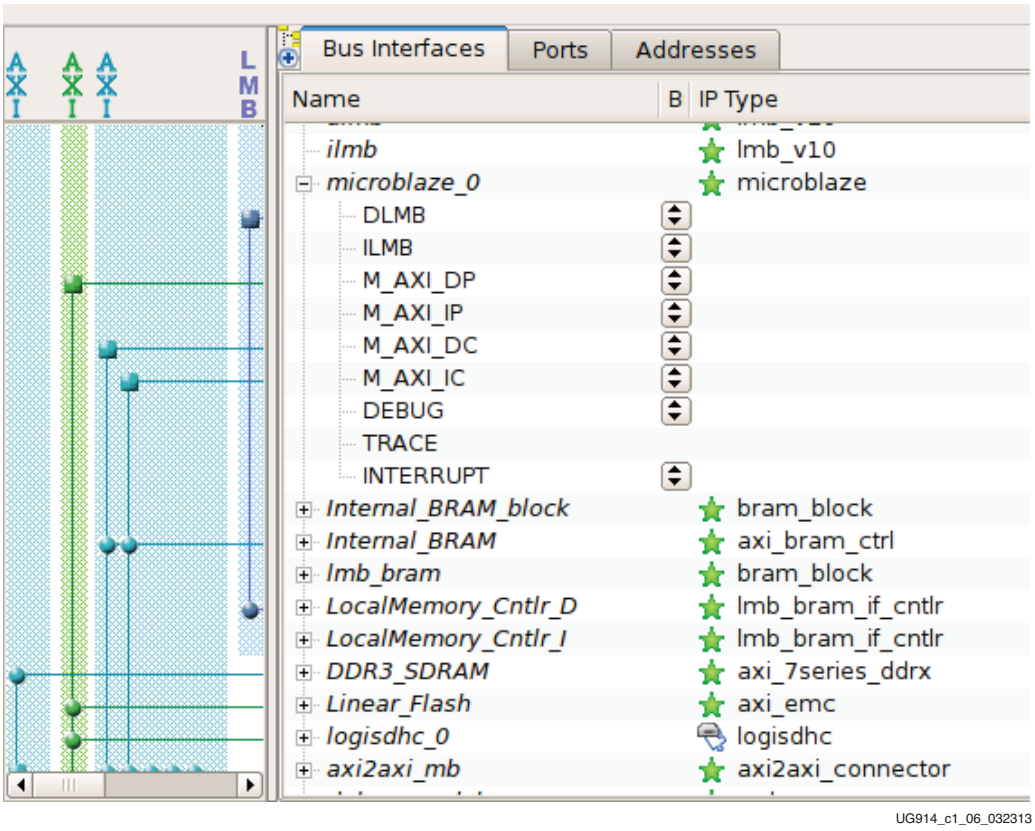


Figure 1-6: **Bus Interfaces View of the MicroBlaze Processor Subsystem**

Examine Ports

1. Select the Ports tab along the top edge of the workspace. By clicking the **+** to the left of any of the components, the connections to that component are visible. (See: Figure 1-7)

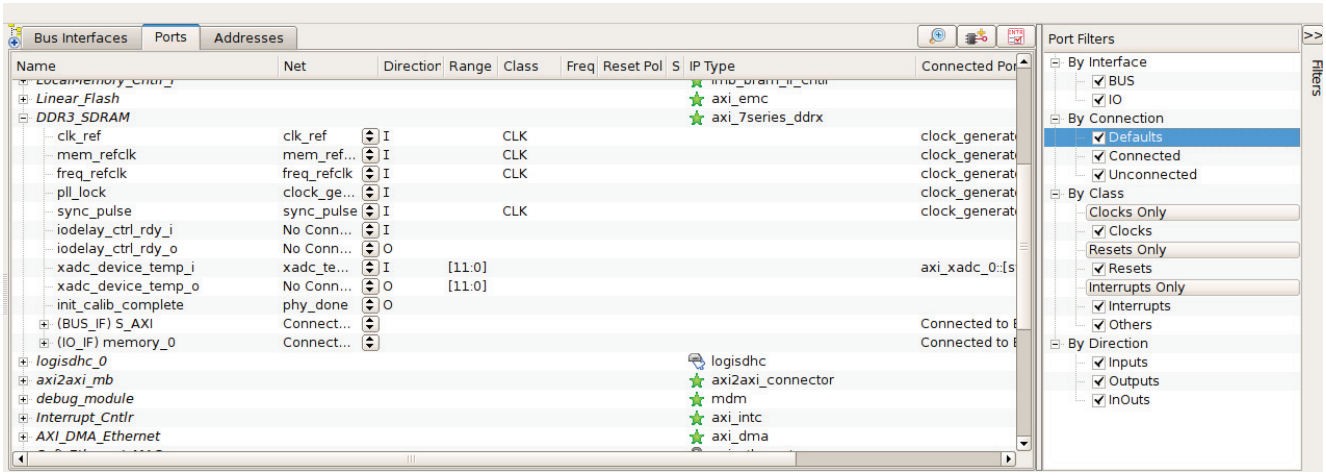


Figure 1-7: **Ports View of the MicroBlaze Processor Subsystem**

Note: The visibility of the port connections for the system is controlled by the Port Filters as shown in Figure 1-8. Many of the peripherals have only default port connections to the AXI interconnect and these connections are not shown unless Defaults filter item is checked.

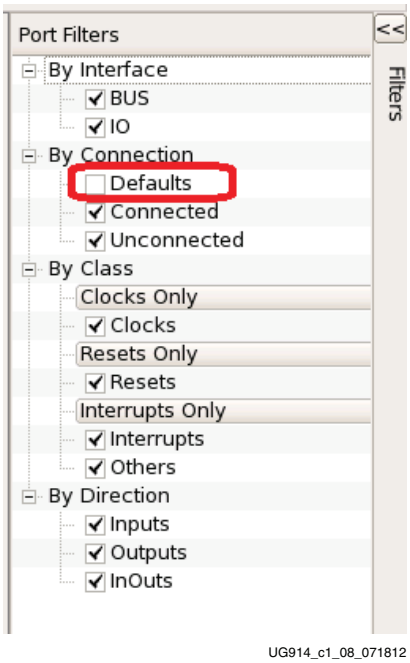


Figure 1-8: **Port Filters**

Examine Addresses

1. Select the Addresses tab along the top edge of the workspace. Click the **+** to the left of the `microblaze_0` address map to expand the map as shown in Figure 1-9. This workspace is used to modify the addresses of peripherals in the system.

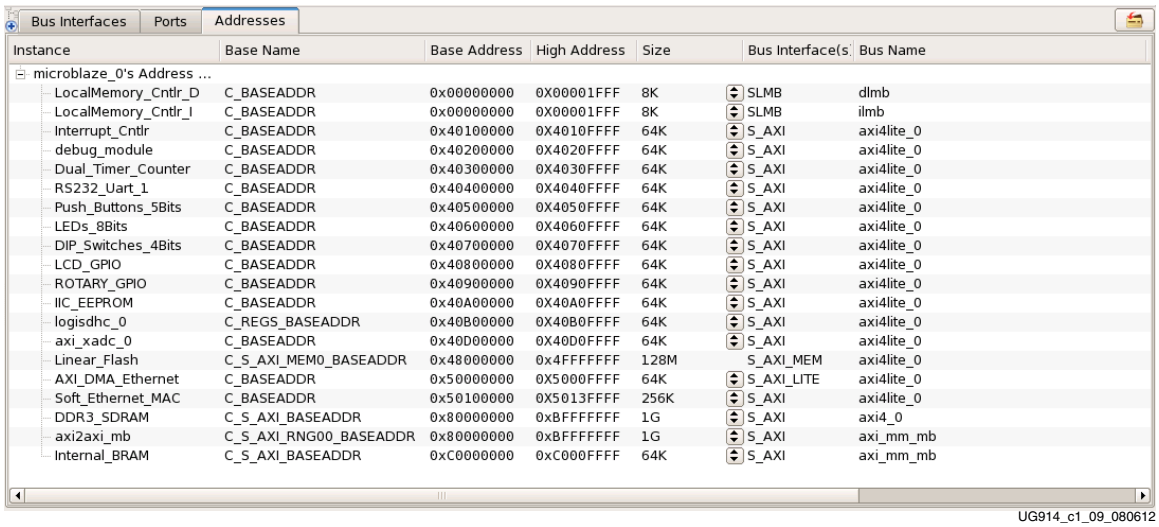


Figure 1-9: **Addresses View of the MicroBlaze Processor Subsystem**

Configure IP Cores

The IP cores within the system are configured to the required functionality for this embedded system. The configurations of the IP cores are described in the System Configuration section of DS669, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Data Sheet*.

1. With the Bus Interfaces tab selected in the System Assembly view, right-click the IP core of interest, as shown in Figure 1-10.

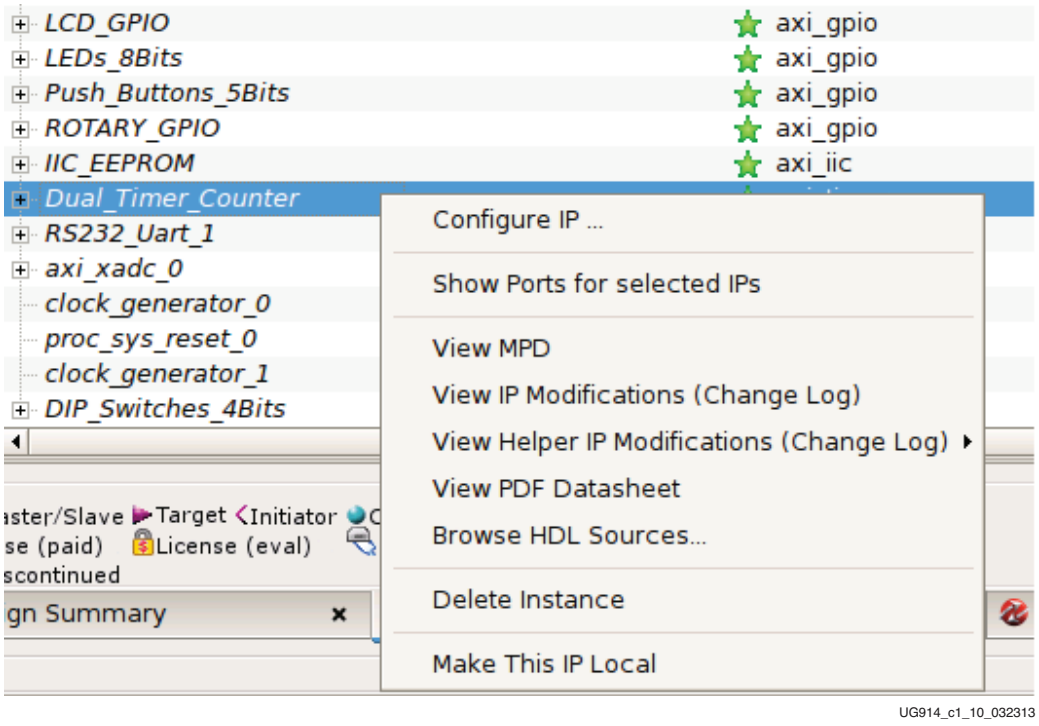


Figure 1-10: **Menu Option to Configure IP**

2. Select **Configure IP...** and the configuration wizard for that core is displayed. Select the tab to show the particular feature or parameter of interest. Figure 1-11 shows the configuration wizard for the `Dual_Timer_Counter` core.

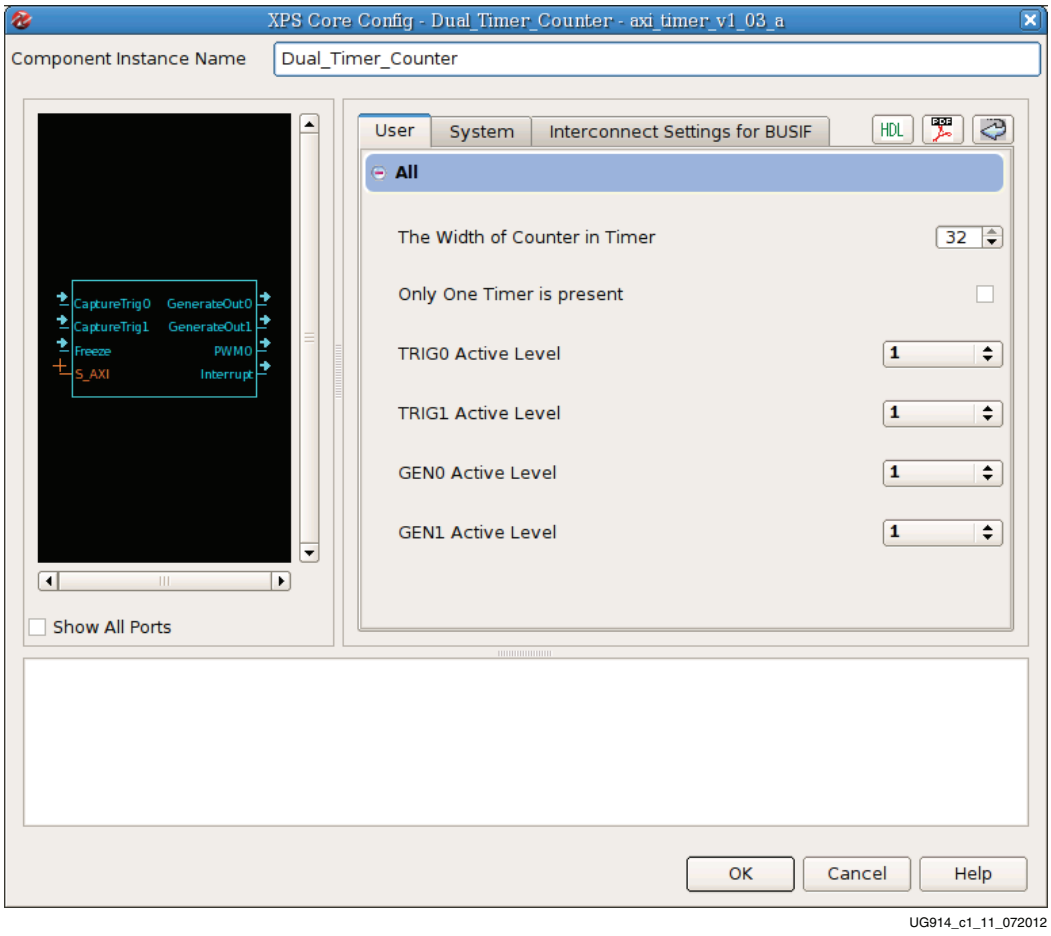


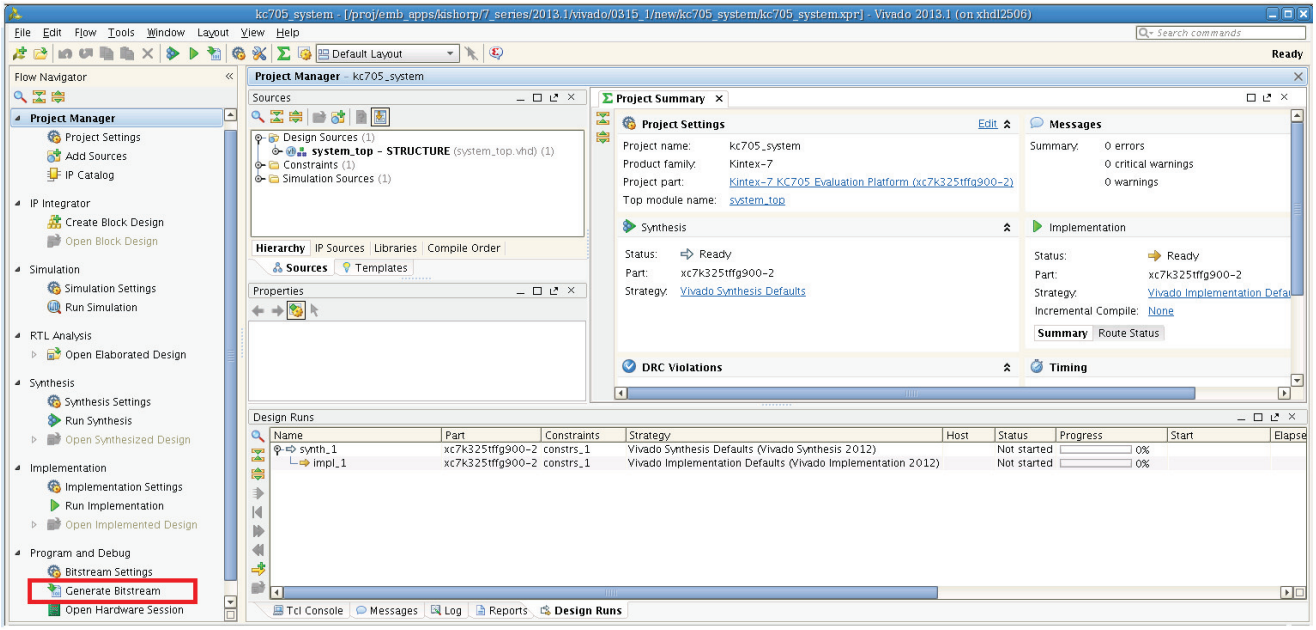
Figure 1-11: **Dual Timer Counter - AXI_Timer Configuration Wizard**

3. Click **Cancel** to return to the System Assembly View since no changes are needed to the cores in the system at this time.

Generate the Hardware Platform

1. Close the XPS project by selecting **File > Exit**.
2. In the Project Manager/Sources pane, click **system_top - STRUCTURE**.
3. In the Flow Navigator pane, double click **Generate Bitstream** under the Program and Debug section, as shown in Figure 1-12.

   Note: Because the design is not yet implemented, a message window will open to indicate that no implementation results are available. Click **Yes** to close the message window.



UG914_c1_12_032313

Figure 1-12: **Generating the Hardware Bitstream**

Export the BIST Hardware Platform to SDK

Because the BIST hardware platform is used with the Software Development Kit (SDK) as described in UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*, information about the hardware platform must be provided to the SDK to allow development of software platforms and applications. Use the following steps to export this information to the SDK:

1. When bitstream generation is complete:
   a. In the Flow Navigator pane, double click **Open Implemented Design** under the Implementation section.
   b. Under Sources, click **system_i - system (system.xmp)**.
   c. Under the File tab, select **Export** and click **Export Hardware** to open the dialog shown in Figure 1-13.

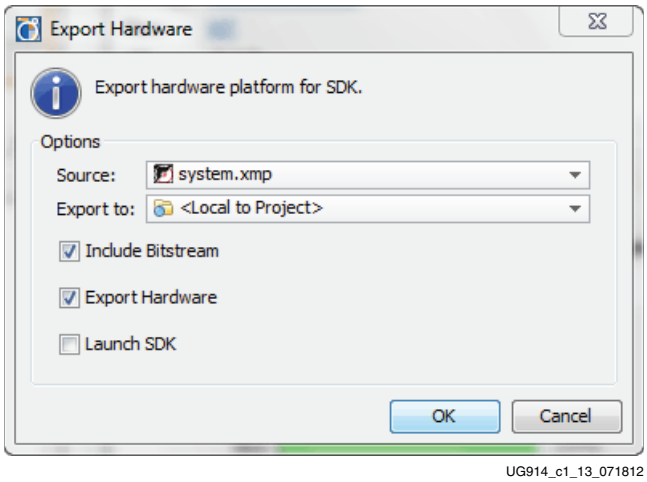


Figure 1-13: **Exporting BIST Hardware Platform to SDK**

d. The exported hardware description files are stored in this location:

`KC705_Embedded_Kit/Tutorial_Sandbox/HW/BIST/kc705_system.sdk/SDK/SDK_Export`

2. Copy the SDK_Export directory into the SW area of the Tutorial sandbox for use with the procedure described in UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*. The existing directory can be overwritten.

   Copy:

   `KC705_Embedded_Kit/Tutorial_Sandbox/HW/BIST/kc705_system.sdk/SDK/SDK_Export`

   to:

   `KC705_Embedded_Kit/Tutorial_Sandbox/SW/board_test_app`

   The SDK is run using the procedure described in UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*.

Download and Verify BIST Design

1. The steps for testing the newly generated bit file are the same as those used to generate the System section where pre built bit and ELF files are tested. The only difference is that the new bit file should be loaded from `KC705_Embedded_Kit/Tutorial_Sandbox/HW/BIST/kc705_system.runs/impl_1/download.bit`
2. The UART output will be displayed and should look similar to Figure 1-4, page 11.
3. Execute the chosen tests and then select **0** to exit. Refer to UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial* for details about these test options.

At this point in the tutorial, the KC705 embedded kit MicroBlaze processor subsystem is successfully built, verified, and exported for use with the SDK. The next section in this tutorial explains how to customize an embedded hardware platform. Alternately, if no modifications to the hardware subsystem are required, proceed with the tutorial presented in UG915, *KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*.

Customize the Embedded Hardware Platform

In this section of the tutorial, the partially built video demo system from the Tutorial Sandbox directory is modified to include an additional AXI_VDMA for developing a complete video system. After building the hardware, the video demonstration application uses a Web server to display the video throughput data.

Every data point consists of a data transfer of a gigabyte by means of multiple AXI Video DMAs. Each Video DMA transfers the video data to and from the DDR memory through the AXI interconnect and the DDR controller. The Perf_AXI core monitors the transactions on the AXI interface to provide data throughput calculations.

Note: For More details on the video demo, refer to DS669, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Data Sheet.*

The steps to customize the MicroBlaze processor subsystem include:

1. Add IP from the Xilinx IP Catalog.

   Note: The VDMA which will be added to the partially built video demo system is highlighted in Figure 1-14.

2. Connect the Bus Interfaces.
3. Connect the Ports.

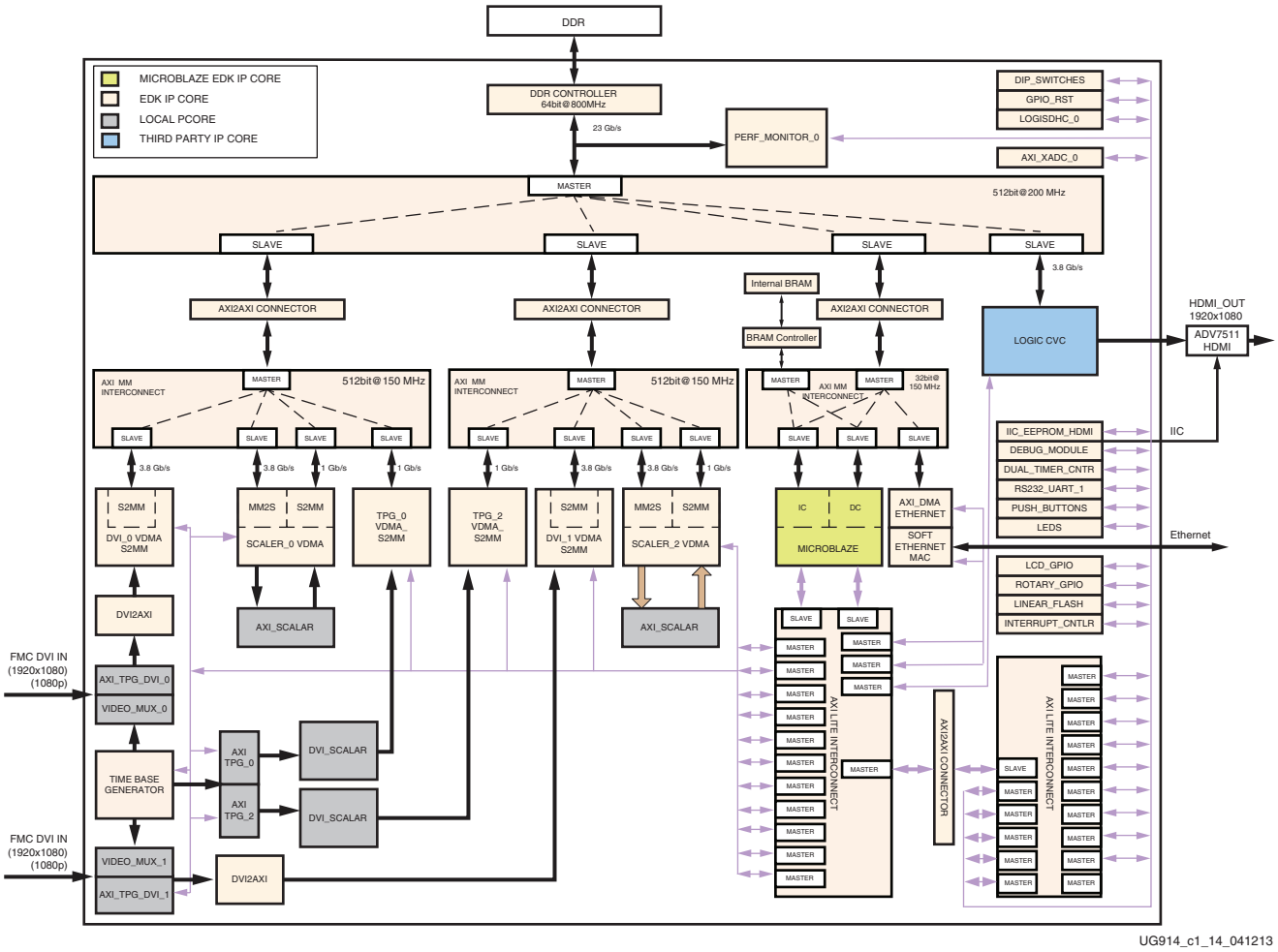


Figure 1-14: **Video Demo MicroBlaze Processor Subsystem**

Add IP from the Xilinx IP Catalog

To Add the AXI VDMA Core:

1. The PlanAhead tool should already be running (from the previous section of this tutorial).
2. Close the BIST project by selecting **File > Close Project**.
3. In Project Commands, select **Open Project...**
4. Browse to `KC705_Embedded_Kit/Tutorial_Sandbox/HW/Video_Demo` and select **video_demo.ppr**. Click **OK**.
5. Under Sources, expand Design Sources and system_top -STRUCTURE and double click **system_i - system (system.xmp)**. This invokes XPS for the XPS subsystem.
6. In the EDK_Install section, Expand the DMA and Timer list by clicking the respective **+**. The AXI Video DMA IP core is now displayed.

   Note: The description field can be made wider by placing the cursor near the divider between the fields until it changes into two vertical bars with two opposing horizontal arrows. Click and drag to adjust the width.
7. Right-click the AXI Video DMA and select Add IP, as shown in Figure 1-15.

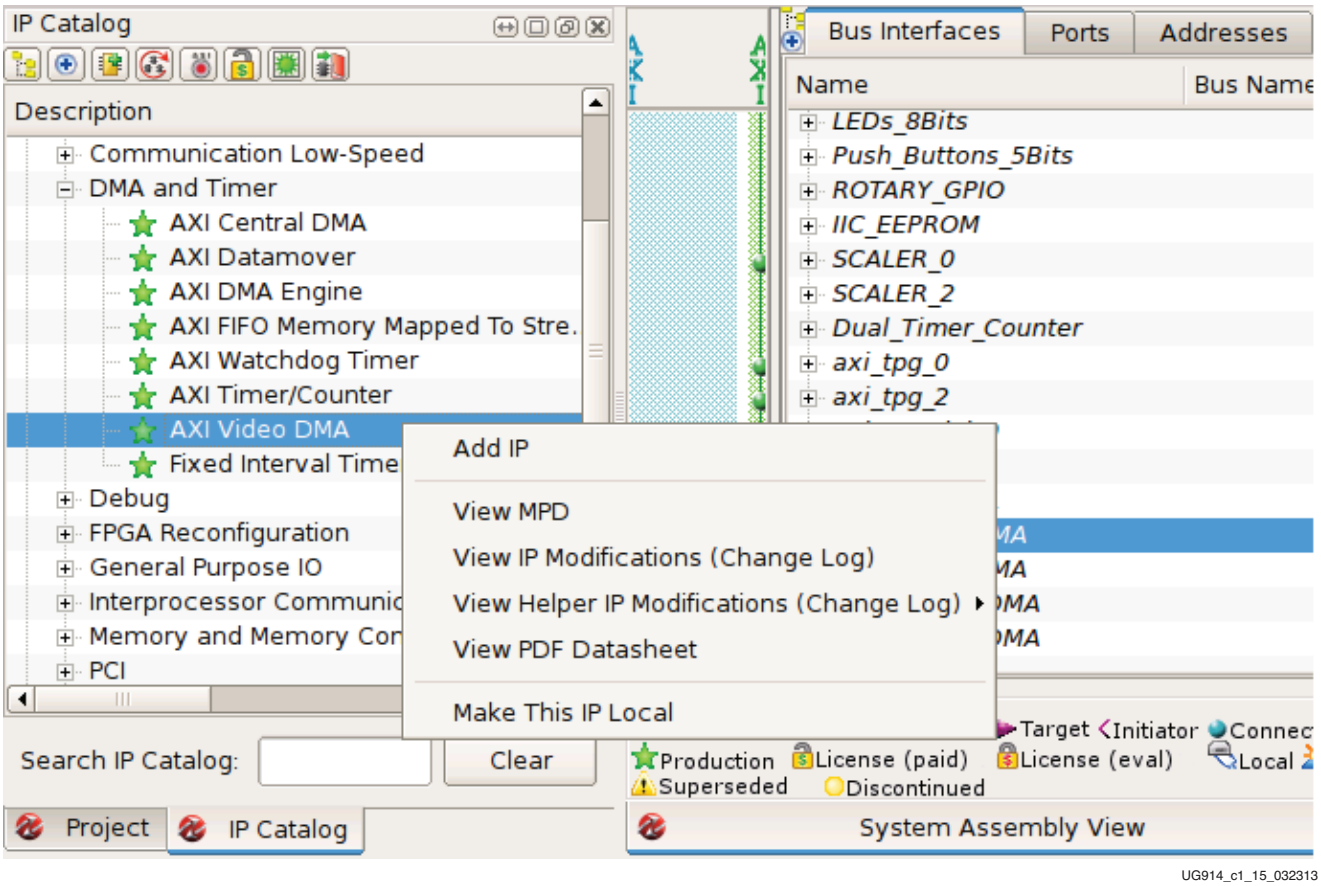


Figure 1-15: **Adding the AXI_VDMA Core**

Note: Warning messages about the axi_vdma or other cores added into the system not being accessible from any processor in the system can be safely ignored. The procedure to connect the bus interfaces is discussed later in this tutorial.

8. The XPS Core Configuration dialog is automatically invoked (See: Figure 1-16, page 24). This dialog sets AXI interconnect parameters for the core such as register slicing.
9. In the Configuration dialog, make the following settings:
   - Component instance name: **DVI_IN_1_VDMA**
   - In the USER tab:
     - Select Enable Asynchronous Clocks within the VDMA options
     - Clear Enable Channel within MM2S Channel options
   - Do following changes within the S2MM Channel options:
     - Set Memory Map Data Width as **64** from pull down menu
     - Select Allow Unaligned Transfers (DRE)
     - Set Maximum Burst Size to **256** from pull down menu
     - Set Line Buffer Depth to **4096**
   - In the SYSTEM tab:
     - Base Address: **0x50900000**
     - High Address: **0X5090FFFF**
   - In the Interconnect Settings for BUSIF tab, M_AXI_S2MM section
     - User Register Slice on all channels (AW,AR,W,R and B): **AUTOMATIC**
     - Write Data FIFO Depth: **512 (BRAM)**
   - In Interconnect Settings for BUSIF tab, S_AXI_LITE section
     - User Register Slice on all channels (AW,AR,W,R and B): **AUTOMATIC**
   - Click **Ok**

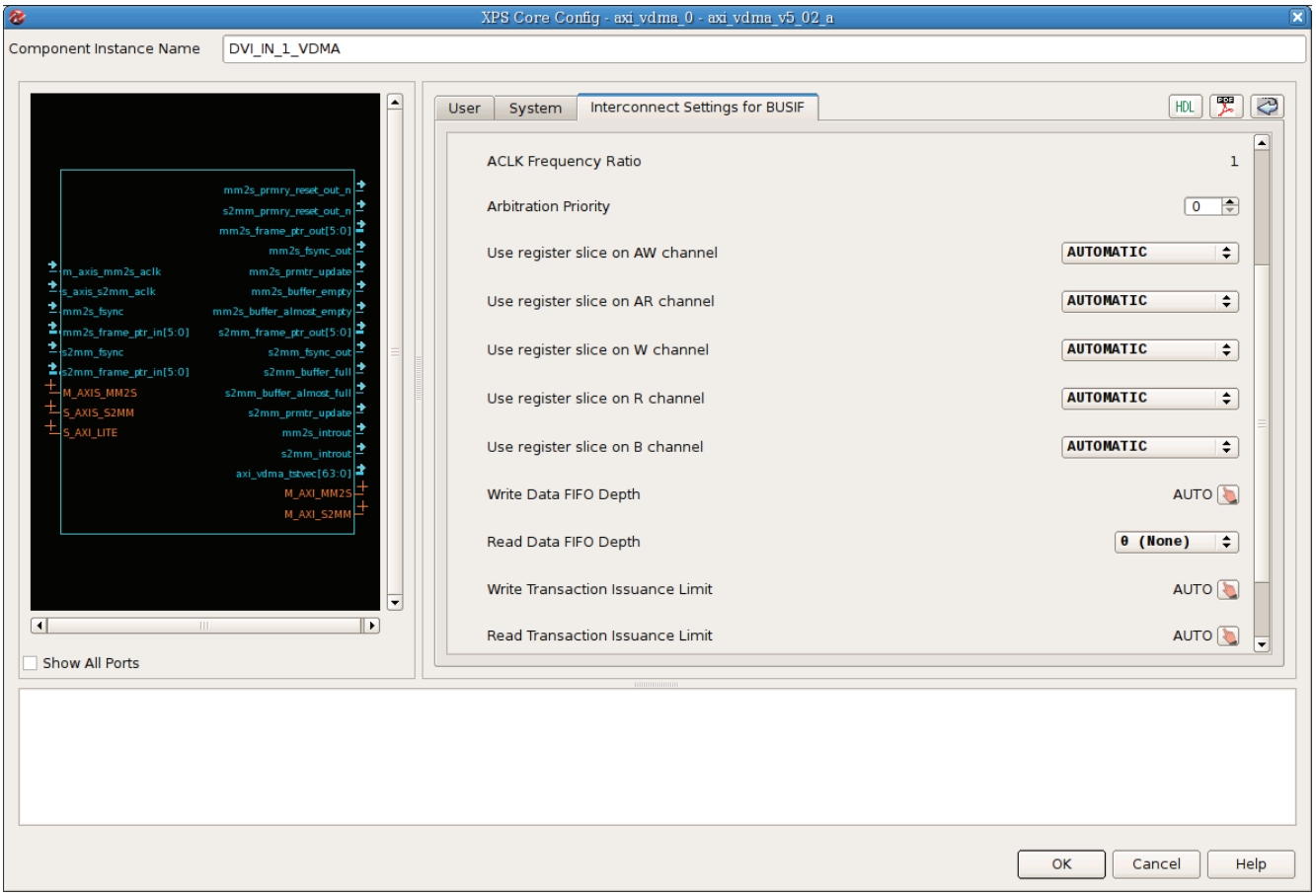


Figure 1-16: **Setting AXI_VDMA Parameters**

10. In the Instantiate and Connect IP dialog, which is automatically invoked, select **User will make necessary connections and settings** then click **OK**. The procedure to connect the bus interfaces will be covered next.

Connect the Bus Interfaces

1. Expand the DVI_IN_1_VDMA core in the System Assembly View (SAV) by clicking the respective **+**. Connect `M_AXI_S2MM` to `AXI_MM_VIDEO34` using the pull-down menu as shown in (Figure 1-17).

   Note: Ignore error messages related to bus interface and port connections.

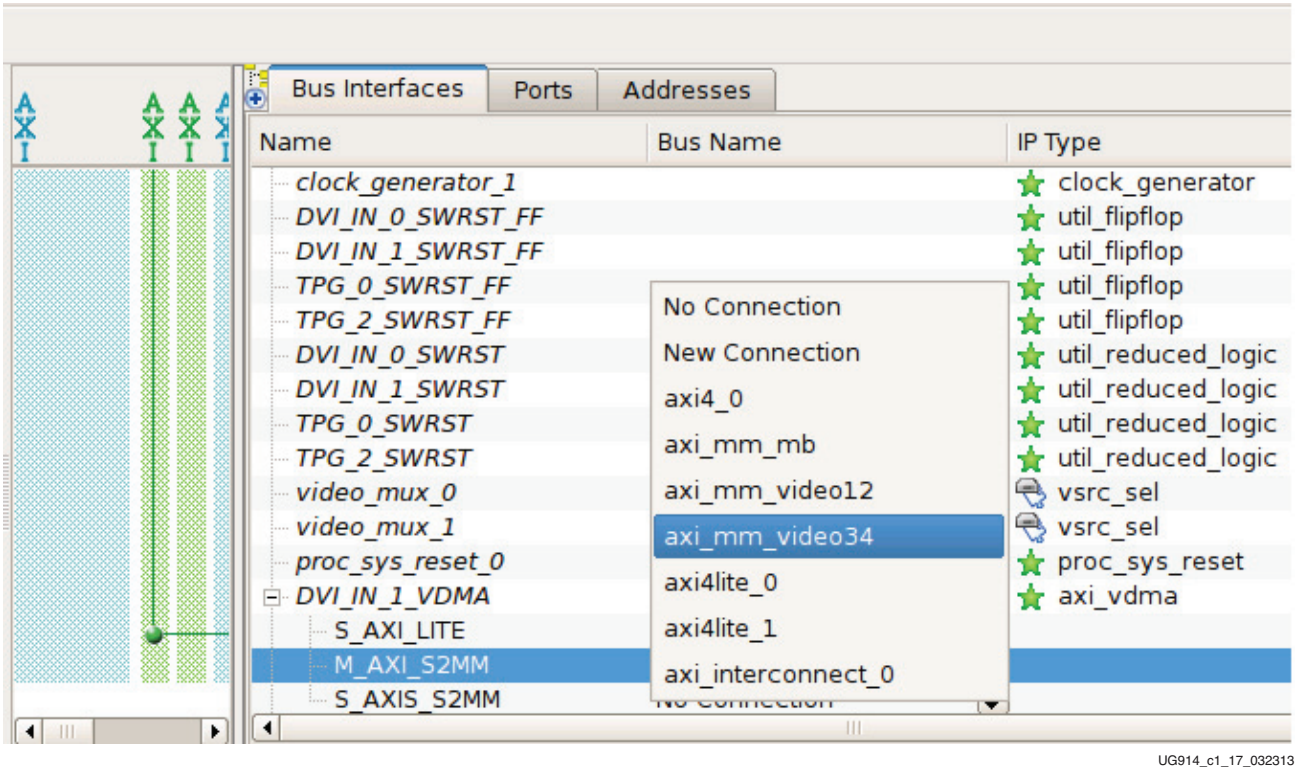


Figure 1-17: **Connecting DVI_IN_1VDMA Master Interface to AXI MM interconnect**

2. For S_AXI_LITE, click the **No Connection** box. The Connection Dialog box appears. In the left column, select **AXI4Lite_0** as shown in Figure 1-18 and click **OK**.

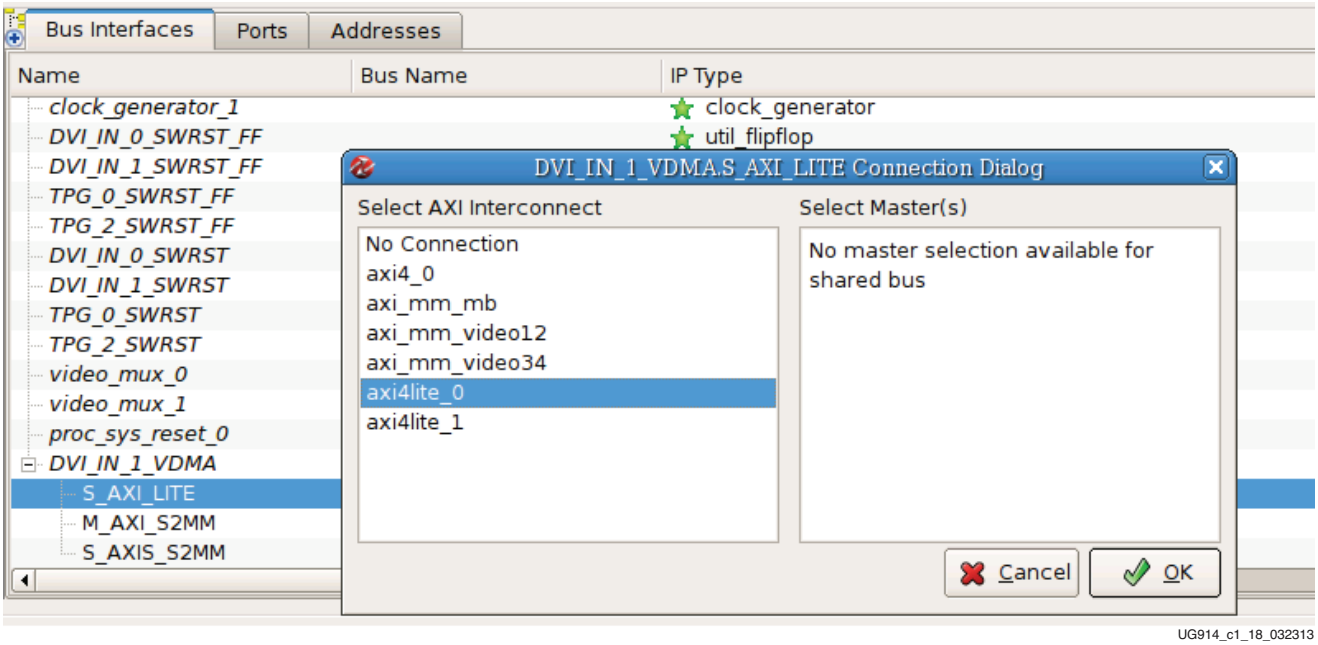


Figure 1-18: **Connecting DVI_IN_1VDMA Lite Slave Interface to AXI Lite interconnect**

3. Connect S_AXIS_S2MM to DVI_IN_1_S_AXIS_S2MM using the pull-down menu as shown in Figure 1-19.

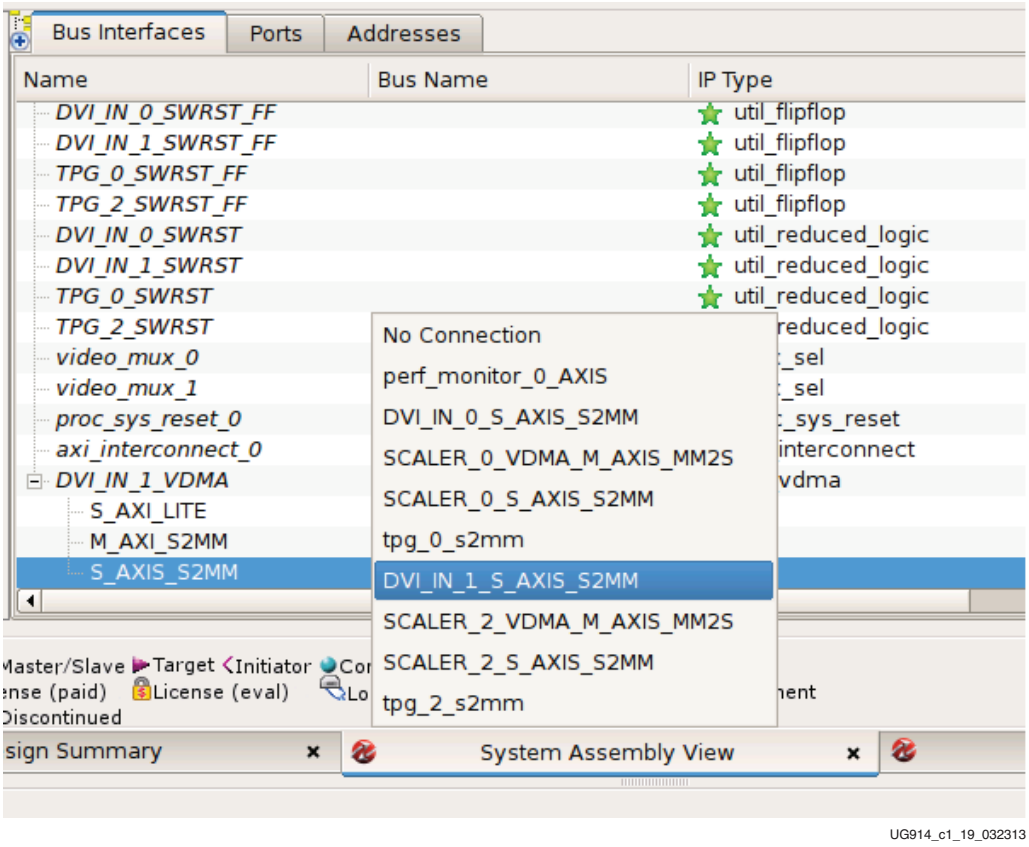


Figure 1-19: **Connecting DVI_IN_1VDMA Stream Slave Interface to Video Streaming Master**

4. Expand the axi2axi_video34 core in the Bus Interface tab of System Assembly View of the XPS GUI by clicking the respective +.
5. Click the **Bus Name** column for the S_AXI row. This brings up the master list dialog for the axi2axi_video34 connector.
6. Select **DVI_IN_1_VDMA.M_AXIS_S2MM** and click **OK** (see Figure 1-20).

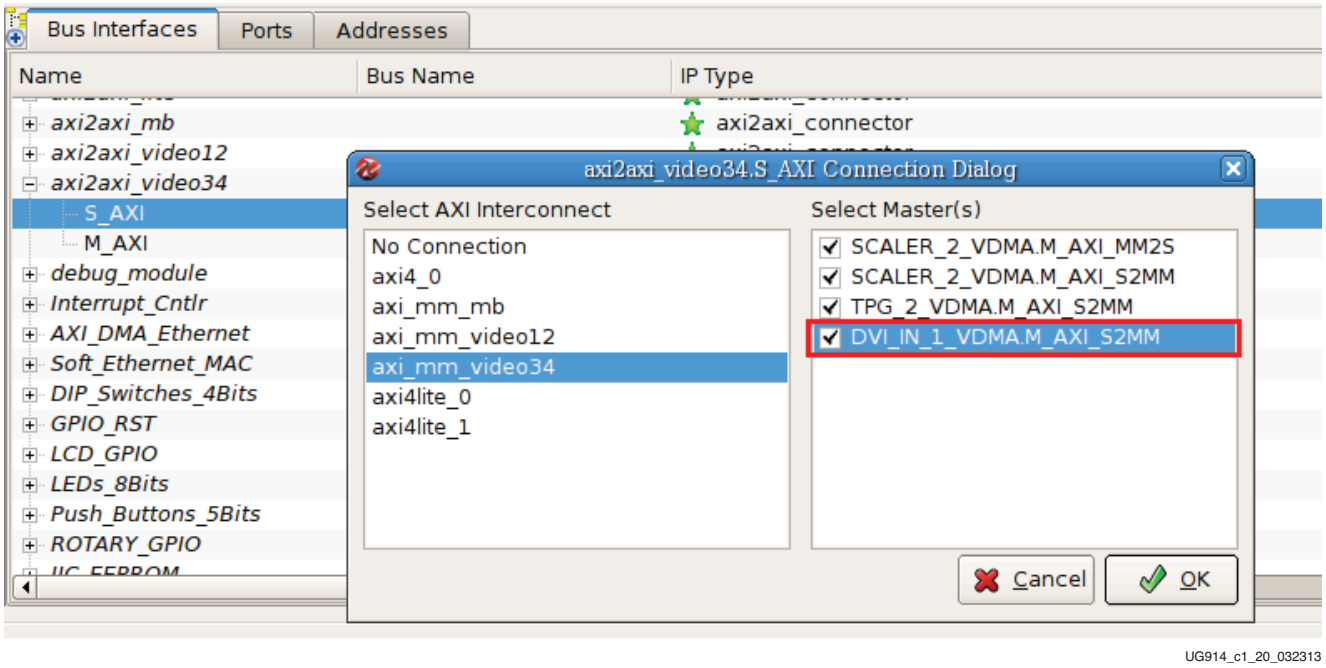


Figure 1-20: **Adding DVI_IN_1VDMA as AXI MM Master to AXI2AXI Connector**

Connect the Ports

The section covers connecting clock pins, interrupts, and signals for the DVI_IN_1_VDMA. In the Port Filters section of the XPS window, ensure that **Defaults** is selected (refer to Figure 1-8, page 15).

1. Click the **Ports** tab at the top of the System Assembly View.
2. Expand the DVI_IN_1_VDMA core in the System Assembly View by clicking the respective **+**.
3. In the name column of the port list for **s_axis_s2mm_aclk**, select the net name **sys_clk_s** using the pull-down menu.
4. Expand (BUS_IF) S_AXI_Lite by clicking the respective **+**. In the name column for **s_axi_lite_aclk**, select the net name **sys_clk_axilite_s** using the pull-down menu in the Net column.
5. Expand (BUS_IF) M_AXI_S2MM by clicking the respective **+**. In the name column for **m_axi_s2mm_aclk**, select the net name **sys_clk_s** using the pull-down menu.
6. Expand (BUS_IF) S_AXIS_S2MM by clicking the respective +. In the name column for **s_axis_s2mm_tdata**, enter the new connection name, **0x00 & dvi2axi_tdata_1**.
7. In the name column for **s2mm_fsync_out**, enter the new connection name, **fsync_from_dvi_in_1_vdma**.
8. In the name column for **s2mm_introut**, enter the new connection name, **DVI_IN_1_VDMA_s2mm_introut**.
9. In the name column for **s2mm_prmry_reset_out_n**, enter the new connection name, **DVI_IN_1_S_AXIS_S2MM_RESET_OUT_N**.
10. Expand the Interrupt_Cntlr core in the System Assembly view by clicking the respective **+**.

11. Click the **Net** column for the INTR row. This brings up the Interrupt Connection Dialog.
12. In the left column select **DVI_IN_1_VDMA:s2mm_introut** and click the right arrow (see Figure 1-21).

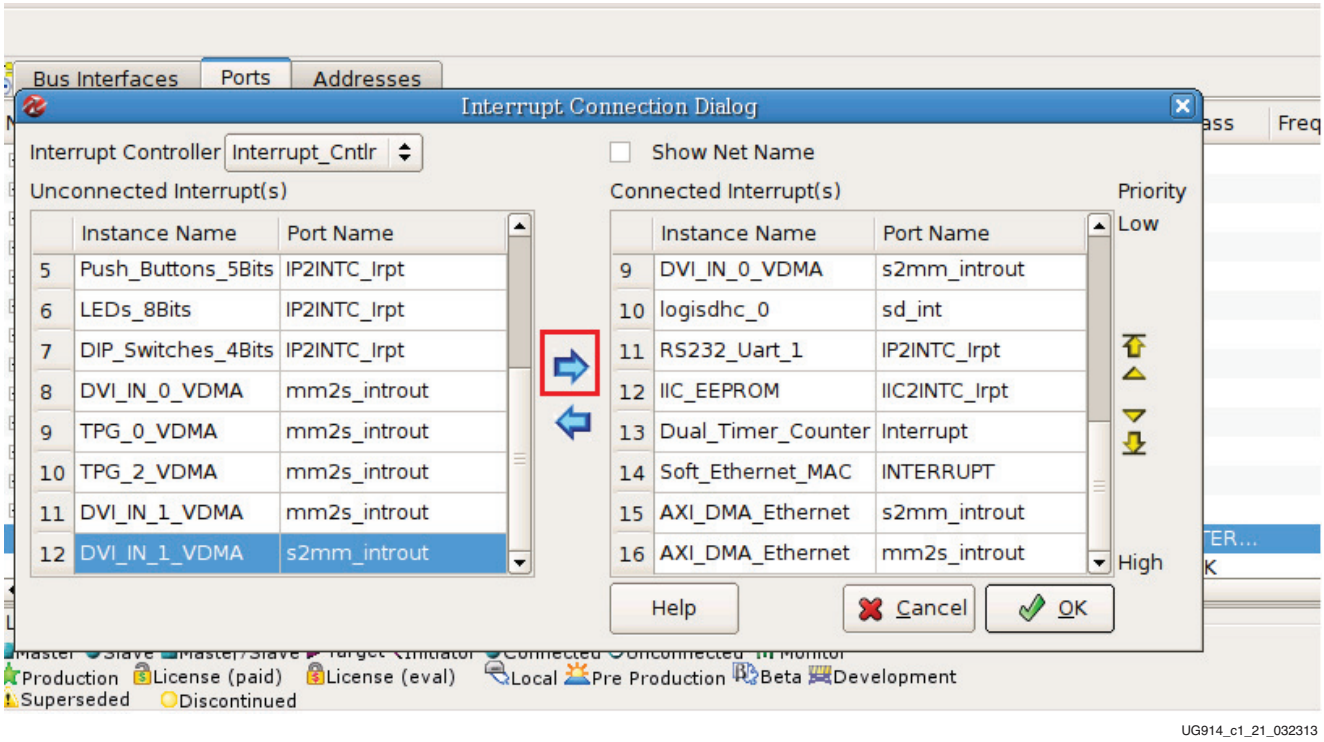


Figure 1-21: **Adding DVI_IN_1_VDMA_S2MM Interrupt to Interrupt controller**

13. In the right column select **DVI_IN_1_VDMA:s2mm_introut** and click the up arrow under Priority Low to set this interrupt priority to **5** and click **OK** (see Figure 1-22).

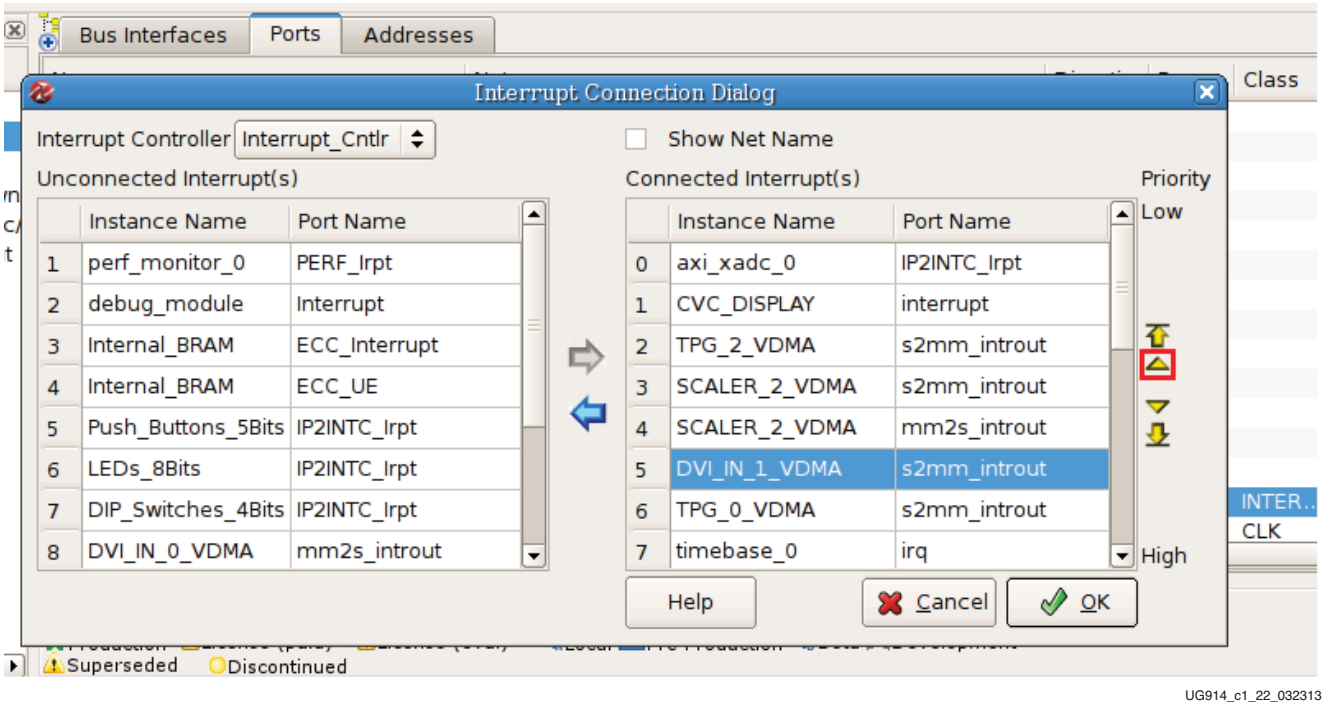


Figure 1-22: **Setting DVI_IN_1_VDMA_S2MM Interrupt priority**

14. As shown in Figure 1-23 Double click **axi_mm_video34** to open the configuration wizard
15. Select **Master Read/Write Settings** and click the **Master/Slave Specific Settings** tab.
16. Set Write FIFO Delay for DVI_IN_1_VDMA.M_AXI_S2MM to **1** and press **OK**.

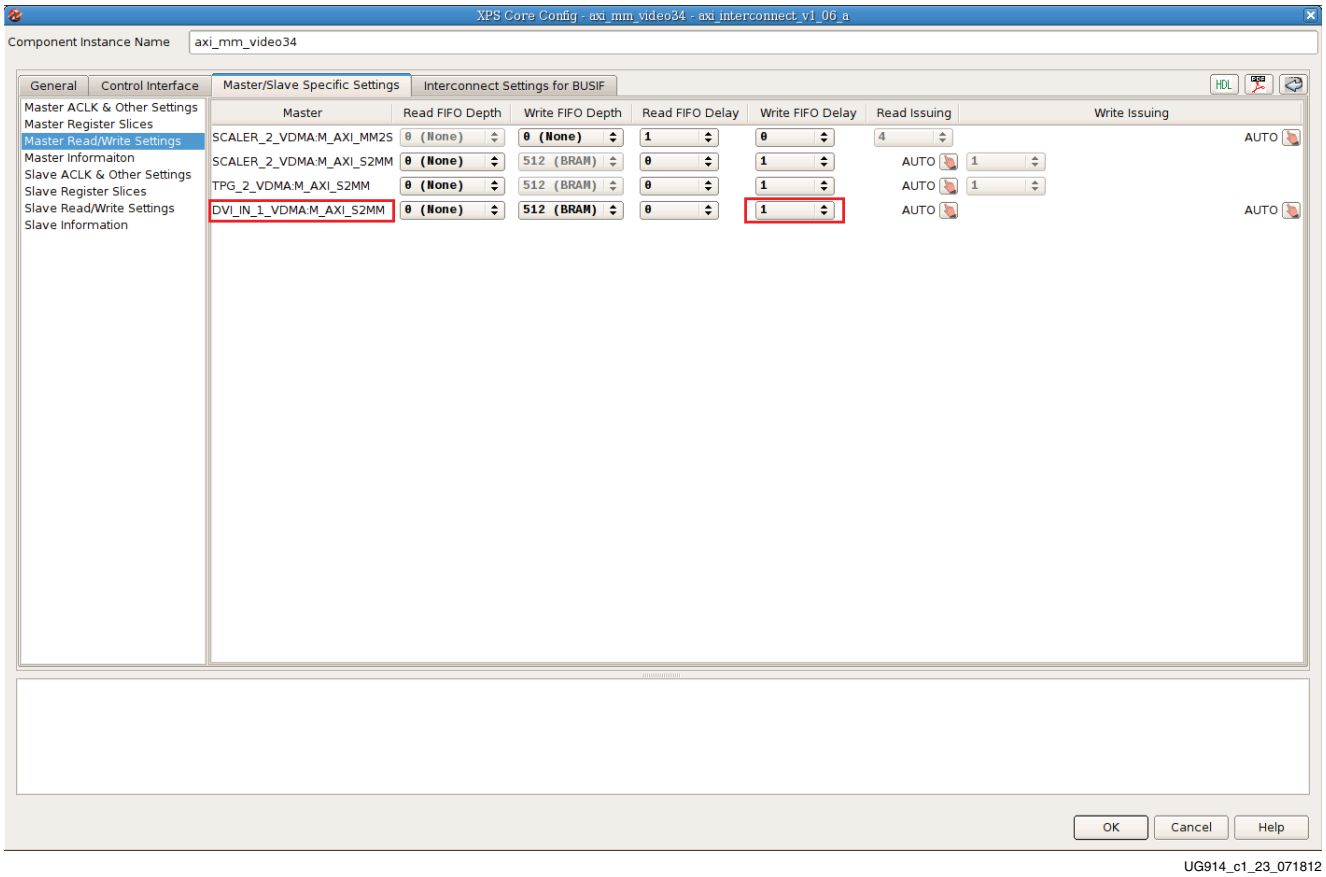


Figure 1-23: **DVI_IN_1_VDMA Interconnect settings**

At this point in the tutorial, additional IP from the Xilinx IP Catalog has successfully been added to the KC705 embedded kit MicroBlaze processor subsystem.

Implement and Test the Design

The design is ready to be synthesized, mapped, placed and routed in the chosen FPGA architecture.

1. Close the XPS project by selecting **File > Exit**.
2. Under Project Manager/Sources, click **system_top - STRUCTURE**.
3. In the Flow Navigator, double click **Generate Bitstream** under the Program and Debug section, as shown in Figure 1-12, page 19.
4. Once the bit file is generated, Export hardware to SDK as explained in Export the BIST Hardware Platform to SDK, page 19.
5. Connect the FMC cards, HDMI cables, Ethernet cable and USB cables as shown in UG913, *Getting Started with the Kintex-7 FPGA KC705 Embedded Kit*, Kintex-7 FPGA Video Demonstration Hardware Setup.
6. Configure the FPGA with the generated bit file, `KC705_Embedded_Kit/Tutorial_Sandbox/HW/Video_Demo/MicroBlaze_Processor_Subsystem/download.bit`

7. Load and run the software ELF file, `KC705_Embedded_Kit/Video_Demo/ready_for_download/Video_Demo.elf`
8. Open the webpage at IP 192.168.1.10 which shows the video demo web server running on the MicroBlaze processor subsystem.
9. Select different menu options on the web page and check the video output and data throughput for different possible options.

Note: For more details on Video Demo testing, different possible menu options and data throughput values refer to UG913, *Getting Started with the Kintex-7 FPGA KC705 Embedded Kit*.

Summary

The tasks accomplished in this tutorial are listed here:

- Loaded and executed a design from a pre-built bitstream and ELF
- Rebuilt a design
- Added IP from the Xilinx IP catalog to an embedded system

Next Steps

Next Step	Refer To
Develop the Software Platform	UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*
Advanced Debugging	UG029, *ChipScope Pro Software and Cores User Guide* and UG111, *Embedded System Tools Reference Manual*, GNU compiler tools
Simulate the Embedded System	UG683, *EDK Concepts, Tools, and Techniques*

XILINX

Appendix A

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

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

www.xilinx.com/support/myalerts.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

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.

Further Resources

The most up to date information related to the KC705 board and its documentation is available on the following websites.

The Xilinx Kintex-7 FPGA Embedded Kit product page:

www.xilinx.com/products/boards-and-kits/DK-K7-EMBD-G.htm

The Kintex-7 FPGA Embedded Kit - Known Issues and Release Notes Master Answer Record:

www.xilinx.com/support/answers/52970.htm

These documents provide supplemental material useful with this user guide:

- UG913, *Getting Started with the Kintex-7 FPGA KC705 Embedded Kit*
- UG683, *EDK Concepts, Tools, and Techniques*
- DS669, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Data Sheet*
- UG111, *Embedded System Tools Reference Manual*
- UG915, *AXI Interface Based KC705 Embedded Kit MicroBlaze Processor Subsystem Software Tutorial*
- UG029, *ChipScope Pro Software and Cores User Guide*

- UG631, *ISE Design Suite 14: Release Notes, Installation, and Licensing*
- UG810, *KC705 Evaluation Board for the Kintex-7 FPGA User Guide*
- UG081, *MicroBlaze Processor Reference Guide Embedded Development Kit*

References

AMBA AXI4-Stream Protocol Specification:

infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0051a/index.html