XILINX®

XAPP1160 (v3.0) July 03, 2014

AXI Chip2Chip Reference Design for Real-Time Video Application

Authors: Ravi Kiran Boddu and Pankaj Kumbhare

Summary

The LogiCORE™ IP AXI Chip2Chip is a soft Xilinx core that provides bridging between Advanced eXtensible Interface (AXI) systems for multi-device System-On-Chip solutions. This application note provides a setup demonstrating real-time video traffic across Kintex®-7 FPGA and Zynq®-7000 All Programmable (AP) SoC boards. The setup uses the AXI Chip2Chip core for connectivity across two Xilinx boards using FMC connector cables.

The reference design includes two embedded systems created with the Vivado IP integrator 2014.1 tool, which is part of the Vivado® Design Suite: System Edition. The axi_chip2chip_v4_2 core is the version used in the reference design. The design also includes software created with the Xilinx Software Development Kit (SDK). Complete IP integrator and SDK project files are provided with this application note to allow you to examine and rebuild this design or use them as a reference for a new design.

Introduction

The AXI Chip2Chip core functions like a bridge to connect two AXI-based systems for multi-device system-on-chip solutions (see *AXI Chip2Chip Product Guide* (PG067) [Ref 1]). The core bridges the AXI transactions in compliance with the AXI protocol specifications. The core provides a low pin count and high performance AXI chip-to-chip bridging solution. The reference design implements a video system in which the Test Pattern Generator (TPG) creates test patterns. Two instances of the core are instantiated, one as a master and one as a slave. The AXI Chip2Chip core in Master mode (Master C2C) and AXI Chip2Chip core in Slave mode (Slave C2C) interface with each other by utilizing FPGA I/O pins, as shown in Figure 1.

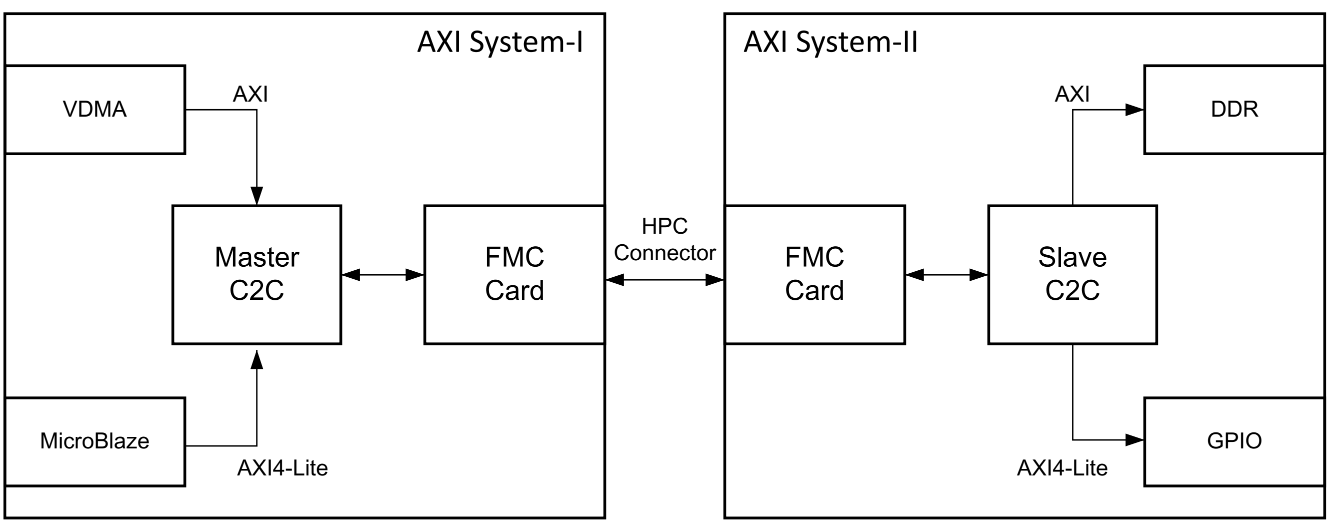


Figure 1: **Typical AXI Chip2Chip Core Interconnections**

The Master C2C has an AXI4 slave interface that is connected to an AXI master peripheral through an AXI interconnect. The Slave C2C has an AXI4 master interface that can be connected to an AXI slave peripheral through an AXI interconnect. By mapping the memory region of the AXI slave peripheral in System-II to the Master C2C, the AXI master peripherals in System-I can access the System-II slave peripherals.

In this application note, System-I contains the AXI Video Direct Memory Access (VDMA) reference design system [Ref 2] with the AXI Chip2Chip core in master mode replacing the AXI 7 series DDRx memory controller. System-II contains the AXI 7 series DDRx memory controller connected to the AXI Chip2Chip core in slave mode of operation through the AXI interconnect. System-I is designated the Master system because it contains the AXI Chip2Chip core in master mode. Similarly, System-II is designated the Slave system. Two reference designs are included: one design shows the connectivity between two Kintex-7 FPGA KC705 boards, and the other design shows the connectivity between a Kintex-7 FPGA KC705 board and a Zynq-7000 AP SoC ZC706 board.

Figure 2 is a block diagram of the reference design and its interconnections. The AXI VDMA on Board-A writes and reads the video data from the external memory on Board-B through the AXI4 interfaces of the Chip2Chip master and slave blocks. The MicroBlaze™ processor on Board-A configures the video peripherals attached to the AXI4-Lite interconnect on Board-A through the AXI4-Lite interfaces. Similar designs can be created with the AXI Chip2Chip core by replacing the AXI VDMA with any master that generates AXI transactions, and replacing external memory with any slave that receives AXI transactions.

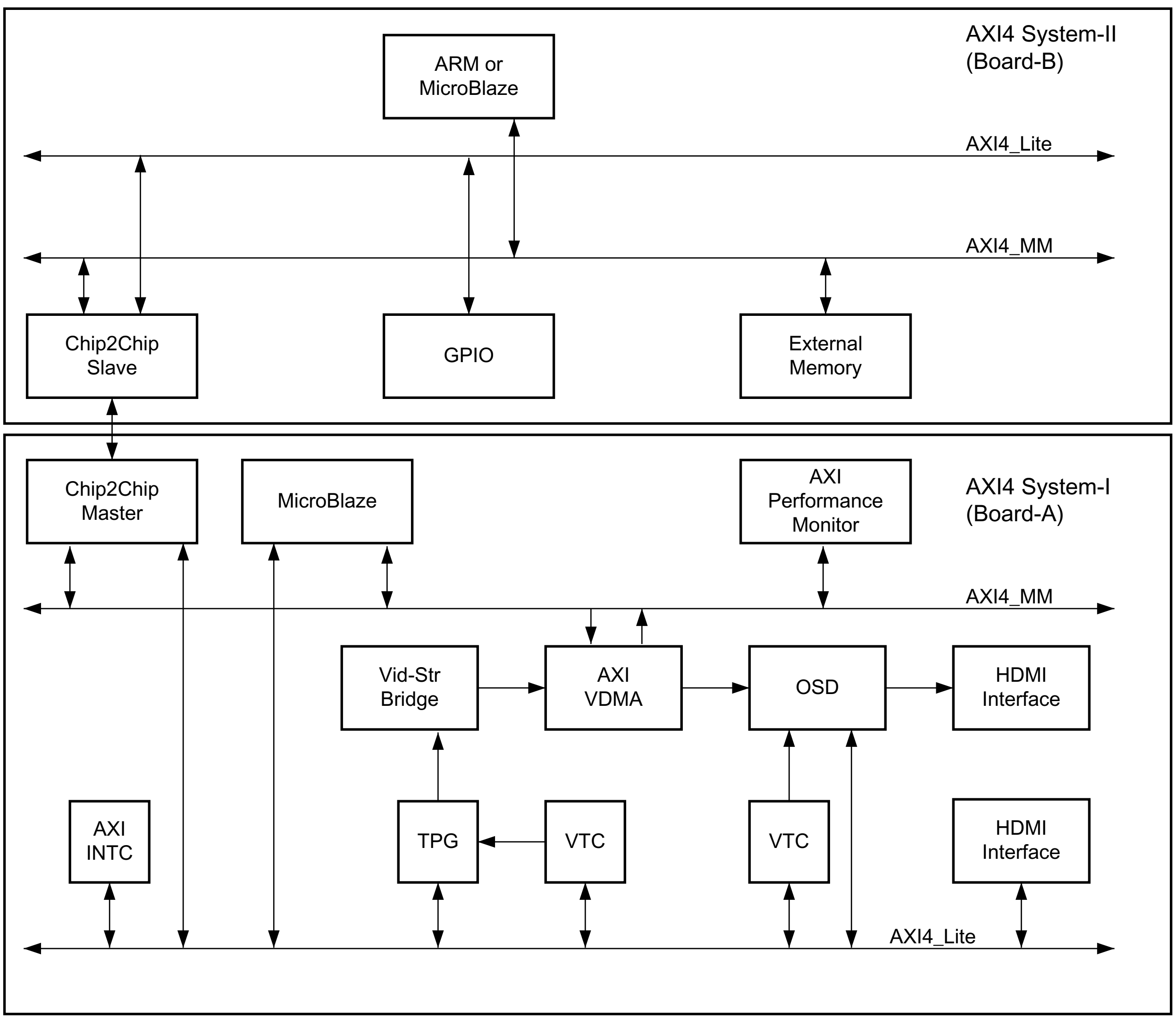


Figure 2: **Reference Systems Block Diagram**

Hardware and Software Requirements

The hardware requirements for this reference design are:

- Two Kintex-7 FPGA KC705 boards or one Kintex-7 FPGA KC705 board and one Zynq-7000 AP SoC ZC706 board
- Two USB Type-A to Mini-B 5-pin cables
- High quality HDMI™ cable (high quality required for proper color display)
- Two USB Type-A to Micro-B 5-pin cables
- Display monitors supporting configurable resolutions (tested with Dell LCD monitor U2410f with HDMI to HDMI cable)
- FMC-to-FMC connector cable

  Note: FMC connector cable can be purchased from [Ref 9].
- Xilinx Vivado Design Suite 2014.1 (System Edition)

Reference Design Specifics

Two reference designs are included in the application note. One design shows the chip-to-chip connectivity across two Kintex-7 FPGA boards, and the other design shows it across a Kintex-7 FPGA board and a Zynq-7000 AP SoC board. Each reference design includes a Master system and a Slave system.

The Master systems of both reference designs are similar and include the following cores. Table 1 lists the address mapping of the peripherals.

- MicroBlaze Processor
- AXI Chip2Chip Bridge
- AXI Interconnect
- Clock Generator
- Processor System Reset (proc_sys_reset)
- AXI IIC
- AXI Interrupt Controller
- Video Timing Controller (VTC)
- Test Pattern Generator (TPG)
- AXI Video Direct Memory Access (VDMA)
- AXI Performance Monitor
- AXI On-Screen Display (OSD)
- HDMI Interface cores

The Slave system on the Kintex-7 FPGA includes the following cores. Table 2 lists the address mapping of the peripherals.

- AXI 7 series FPGA Memory Controller
- AXI Chip2Chip Bridge
- MicroBlaze™ Processor
- AXI Interconnect
- Clock Generator
- Processor System Reset

The Slave system on the Zynq-7000 AP SoC includes the following cores. The processing system (PS) is configured to include the UART and the DDR. The DDR is accessed through the HP0 port.

- AXI Interconnect
- AXI Chip2Chip Bridge (in slave mode)
- Clock Generator
- Processor System Reset

Note: The Zynq-7000 AP SoC processing system (PS) is not included in Figure 2.

Table 1: **System-I Address Map**

Peripheral	Instance	Interface Type	Base Address	High address	Board
axi_chip2chip	axi_chip2chip_0	s_axi	0xC0000000	0xDFFFFFFF	Kintex-7 FPGA
axi_chip2chip	axi_chip2chip_0	s_axi	0x20000000	0x3FFFFFFF	Zynq-7000 AP SoC
axi_chip2chip	axi_chip2chip_0	s_axi_lite	0x50000000	0x5000FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_gpio	axi_gpio_0	s_axi	0x40000000	0x4000FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_iic	axi_iic_1	s_axi	0x40800000	0x4080FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_intc	axi_intc_1	s_axi	0x41200000	0x4120FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_perf_mon	axi_perf_mon_0	s_axi	0x44A50000	0x44A5FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_timer	axi_timer_1	s_axi	0x41C00000	0x41C0FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_uartlite	axi_uartlite_1	s_axi	0x40600000	0x4060FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
axi_vdma	axi_vdma_1	s_axi_lite	0x44A10000	0x44A1FFFF	Kintex-7 FPGA
lmb_bram_if_cntlr	lmb_bram_if_cntlr_1	slmb	0x00000000	0x0001FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
mdm	mdm_1	s_axi	0x41400000	0x41400FFF	Kintex-7 FPGA, Zynq-7000 AP SoC
v_cresample	v_cresample_0	ctrl	0x44A60000	0x44A6FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
v_osd	v_osd_1	ctrl	0x44A20000	0x44A2FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
v_rgb2ycrcb	v_rgb2ycrcb_0	ctrl	0x44A40000	0x44A4FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
v_tc	v_tc_1	ctrl	0x44A30000	0x44A3FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC
v_tpg	v_tpg_1	ctrl	0x44A00000	0x44A0FFFF	Kintex-7 FPGA, Zynq-7000 AP SoC

Table 2: **System-II Address Map**

Peripheral	Instance	Interface Type	Base Address	High address	Board
axi_bram_ctrl	axi_bram_ctrl_0	s_axi	0xE0000000	0xE000FFFF	Kintex-7 FPGA
axi_uartlite	axi_uartlite_0	s_axi	0x40600000	0x4060FFFF	Kintex-7 FPGA
dlmb_bram_if_cntlr	dlmb_bram_if_cntlr	slmb	0x00000000	0x0001FFFF	Kintex-7 FPGA
axi_intc	microblaze_0_axi_intc	s_axi	0x41200000	0x4120FFFF	Kintex-7 FPGA
mig_7series	mig_7series_0	s_axi	0xC0000000	0xDFFFFFFF	Kintex-7 FPGA
ps ddr	processing_system7_0	s_axi	0x20000000	0x3FFFFFFF	Zynq-7000 AP SoC
gpio	axi_gpio_1	s_axi	0x50000000	0x5000FFFF	Zynq-7000 AP SoC

Hardware System Specifics

This section describes the configuration of the AXI Chip2Chip core. For information on hardware system specifics for VDMA configuration and other video-related IP cores, see the *AXI VDMA Reference Design Application Note* (XAPP742) [Ref 2]. For information on AXI system optimization and design trade-offs, see the *Vivado Design Suite: AXI Reference Guide* [Ref 3].

This application note assumes general knowledge of the IP integrator feature. See *Vivado Design Suite Tutorial: Designing IP Subsystems Using IP Integrator* (UG995) [Ref 4] and *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994) [Ref 5] for more information about the IP integrator feature.

Configuring the AXI System-I

This section describes how to configure the AXI System-I.

AXI Chip2Chip Master Instance (master_c2c)

The AXI Chip2Chip core provides two modes of operation: master and slave. In the master mode, the core can be configured as a slave for one or more AXI master peripherals. In the slave mode, the core can be configured as a master for one or more AXI slave peripherals. The core can be configured to act in independent or common clocking mode. In independent clocking mode, the physical layer interface can be operated at a higher or lower frequency compared to the AXI clock. In the common clocking mode, latencies due to clock domain crossing latencies are reduced.

An AXI data width of 32 or 64 bits can be selected based on the system requirements. The Chip2Chip PHY Type and PHY width determine the number of I/O pins used for device-to-device interfacing. Compact 2:1 and 4:1 options reduce the number of I/O pins needed.

In the Kintex-7 FPGA to Kintex-7 FPGA design, the 64-bit AXI Chip2Chip master instance is configured for independent clocking mode with the physical layer operated at a frequency of 200 MHz. In the Kintex-7 FPGA to Zynq-7000 AP SoC design, the 32-bit AXI data width is used to reduce the number of I/Os in the design. The Chip2Chip AXI-Lite interface is configured to act as the AXI master. The master Chip2Chip has two AXI masters: VDMA MM2S and S2MM channels. Therefore, the AXI ID width of the master Chip2Chip is one. The PHY type is configured as SelectIO™ DDR with Compact 1:1 PHY width to obtain a good data rate for transmitting and receiving 1080p real-time video traffic signals. The AXI WUSER width is set to one bit.

Figure 3 shows how the parameters are set for the Chip2Chip master instance in the IP integrator.

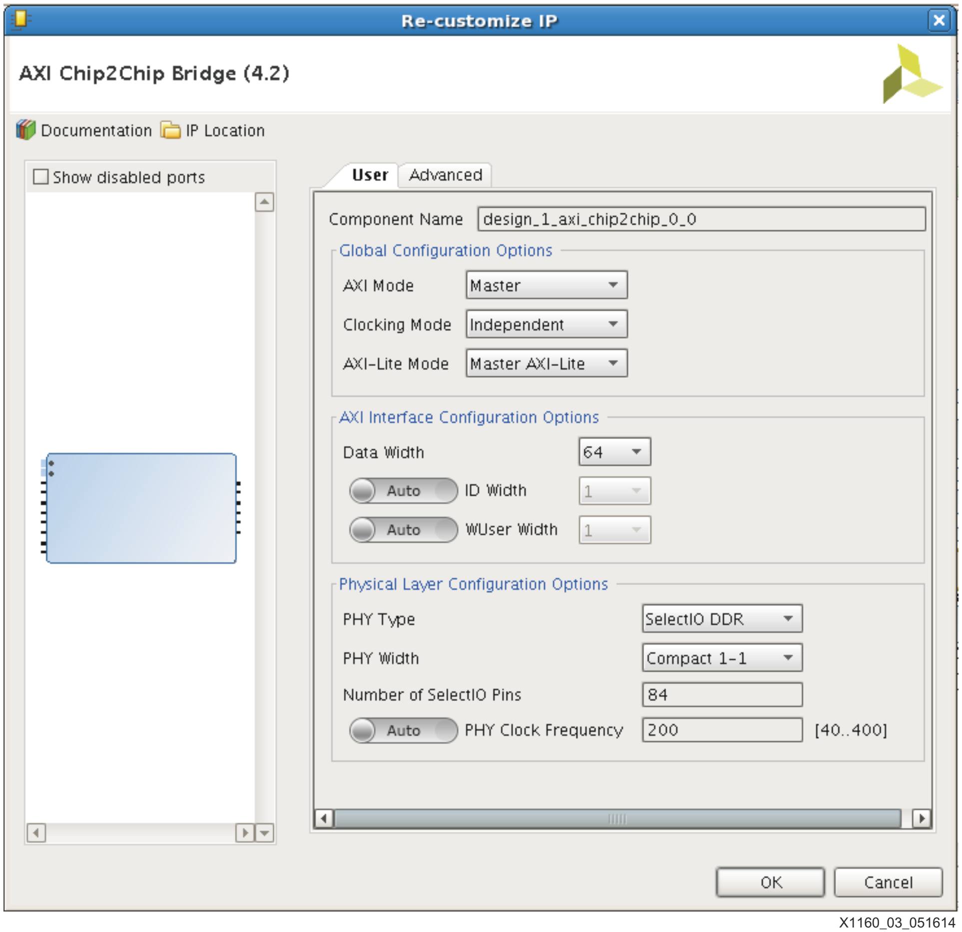


Figure 3: **AXI Chip2Chip Configuration in Master Mode**

AXI Performance Monitor

The LogiCORE™ AXI Performance Monitor core measures major performance metrics for the AMBA® AXI system. The core consists of a slave AXI4-Lite interface for the registers for access by the processor. The AXI performance monitor core only monitors the read and write channels between the AXI slave and the AXI Interconnect. The core does not modify or change any of the AXI transactions it is monitoring.

The core is capable of measuring various performance metrics, such as total read byte count, write byte count, read requests, write requests, and write responses. Count start and count end conditions come from the processor through the register interface. The global clock counter of the core measures the number of clocks between the count start and count end events. The counters used for the performance monitor can be configured for 32 or 64 bits through the register interface. Final user-selectable metrics can also be read through the register interface.

In the reference design, the slave AXI interface of the master AXI Chip2Chip core is monitored and the performance metrics are reported.

Configuring the AXI System-II

This section describes how to configure the AXI Chip2Chip core for the AXI System-II.

AXI Chip2Chip Slave Instance (Slave_c2c)

Figure 4 shows how the parameters are set for the slave Chip2Chip instance. The values of all the parameters except the Chip2Chip mode are identical to the master_c2c instance in the AXI System-I. In general, the AXI Chip2Chip slave configuration parameters AXI Data Width, ID Width, WUSER Width, Chip2Chip PHY Type, PHY Width, and Chip2Chip PHY clock frequency should match the respective parameters of the AXI Chip2Chip master configuration.

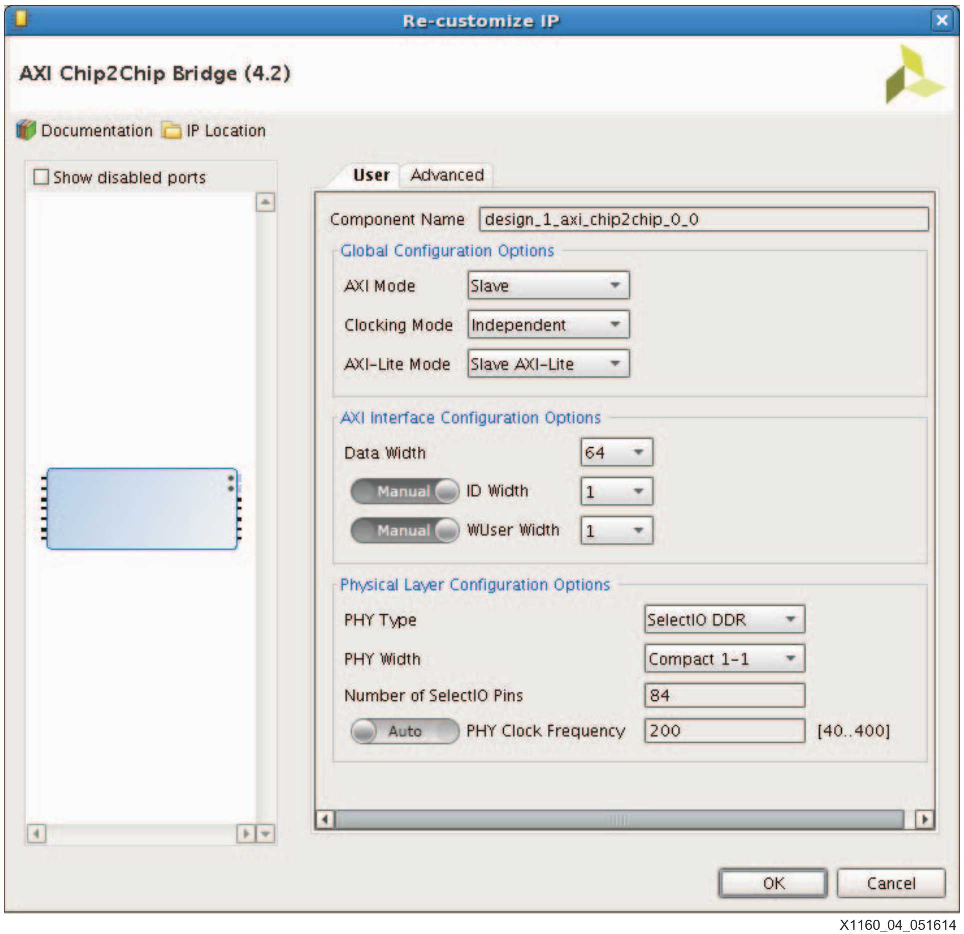


Figure 4: **AXI Chip2Chip Configuration in Slave Mode**

Configuring the Memory System in Kintex-7 FPGA and Zynq-7000 AP SoC

In the Slave system on the Kintex-7 FPGA, the AXI 7 series memory controller is used for interfacing the DDR3 SDRAM device. The AXI interface is 64-bits running at 200 MHz. The core is configured for a write/read acceptance of two and 512-deep write/read. FIFOs are enabled for the port of the AXI interconnect connected to the memory controller. See the *Zynq-7000 SoC and 7 Series Devices Memory Interface Solutions User Guide* (UG586) [Ref 6] for more details on the core.

In the Slave system on the Zynq-7000 AP SoC, the AXI Chip2Chip slave instance connects to the high-performance (HP) slave AXI interface of the PS. The HP port enables a high throughput datapath between the AXI masters in the programmable logic (PL) and the DDR3 memory of the PS.

Software Applications

The application software for the system running on Board-A configures the AXI4-Lite slaves on Board-A. See the *AXI VDMA Reference Design Application Note* (XAPP742) [Ref 2] for more details on the software functionality.

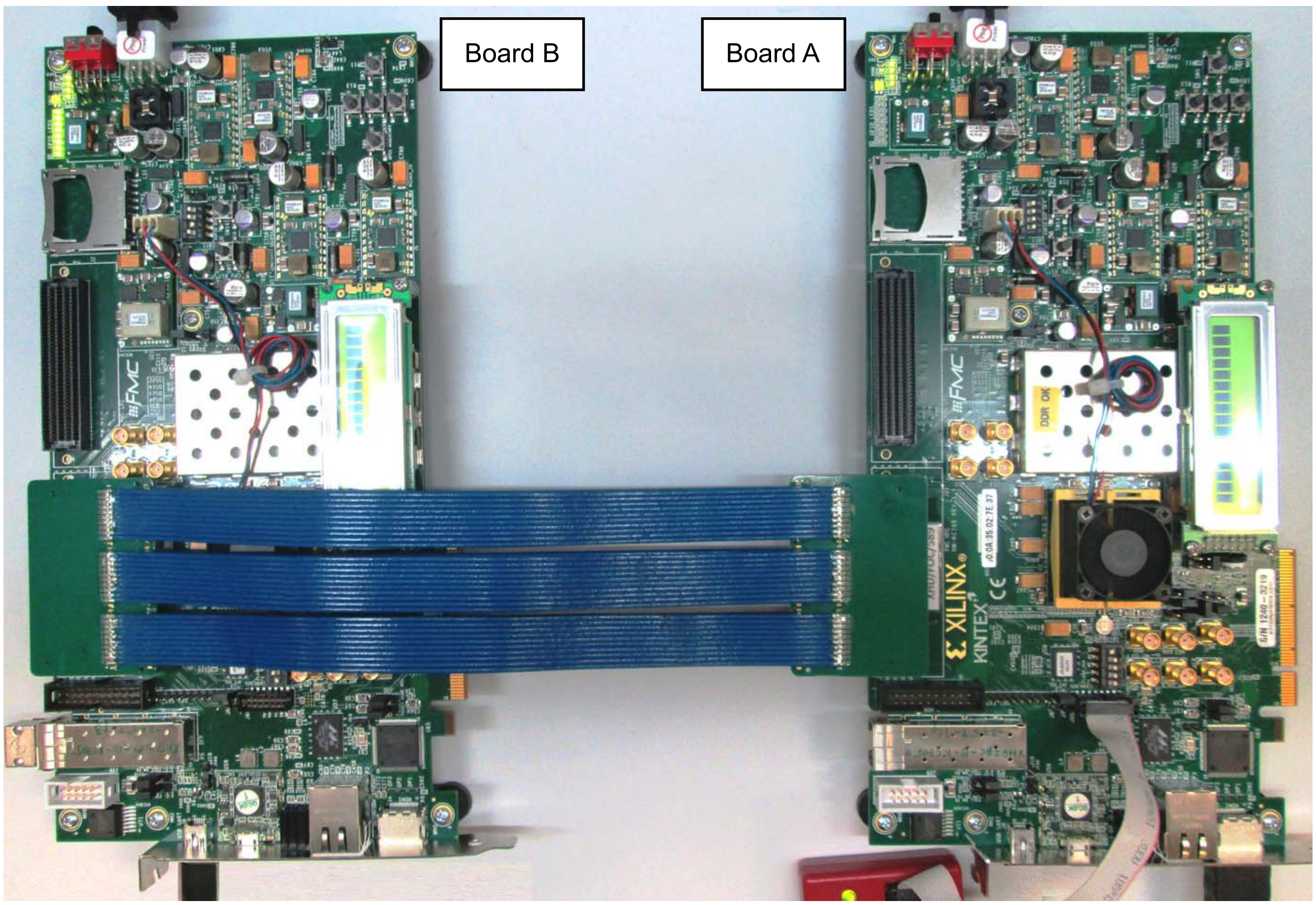
Executing the Reference Design

This section provides instructions for executing the reference designs on hardware.

Kintex-7 FPGA KC705 Board to Kintex-7 FPGA KC705 Board

To execute the reference system on the Kintex-7 FPGA KC705 board to Kintex-7 FPGA KC705 board:

1. Connect the KC705 boards with the FMC-to-FMC HPC connector cable as shown in Figure 5.



X1160_05_051614

Figure 5: **Kintex-7 FPGA KC705 Board to Kintex-7 FPGA KC705 Board Setup**

2. Connect the KC705 HDMI video output of one of the boards to a video monitor capable of displaying a 1920 x 1080p 60 Hz video signal.

   Note: This board is referred to as Board-A and the other board as Board-B in the remaining steps.
3. Connect a USB cable from the host PC to the USB UART port on Board-A.
4. Connect the power supply cable to both boards.
5. Set power ON on both boards.

6. Start a terminal program (for example, HyperTerminal) on the host PC with these settings:
   a. Baud Rate: **9600**
   a. Data Bits: **8**
   a. Parity: **None**
   a. Stop Bits: **1**
   a. Flow Control: **None**
7. Connect the JTAG cable to Board-B.
8. In the command shell or terminal window, change directories (using a 32- or 64-bit command prompt) to the slave download directory:
   ```
   % cd <unzip_dir>/c2c_selectio/kintex-kintex/ready_to_download/slave
   ```
9. Start the Xilinx Microprocessor Debugger (XMD):
   ```
   % xmd
   ```
10. Download the bitstream file to Board-A:
    ```
    XMD% fpga -f design_1_wrapper.bit
    ```
11. Stop XMD:
    ```
    XMD% exit
    ```
12. Connect the JTAG cable to Board-A.
13. Change directories to the master download directory:
    ```
    % cd <unzip_dir>/c2c_selectio/kintex-kintex/ready_to_download/master
    ```
14. Start XMD:
    ```
    % xmd
    ```
15. Download the bitstream file to Board-B:
    ```
    XMD% fpga -f design_1_wrapper.bit
    ```
16. Connect the processor:
    ```
    XMD% connect mb mdm
    ```
17. Disable reset of the entire system on download of the software:
    ```
    XMD% debugconfig -reset_on_run system disable
    ```
18. Reset the processor:
    ```
    XMD% rst -processor
    ```
19. Download the Executable and Linkable Format (ELF) processor code file:
    ```
    XMD% dow app_master.elf
    ```
20. Run the software to execute the reference system:
    ```
    XMD% run
    ```
21. Select a resolution and a frame rate from the UART menus.
22. Select one of the patterns from the pattern menu.
23. Enter the numbers to see the different patterns in the menu or the DDR bandwidth through the AXI performance monitor.

Kintex-7 FPGA KC705 Board to Zynq-7000 AP SoC ZC706 Board

To execute the reference system on the Kintex-7 FPGA KC705 board to the Zynq-7000 ZC706 board:

1. Connect the KC705 board to the ZC706 board with the FMC-to-FMC connector cable interfacing the HPC connector pins as shown in Figure 6.

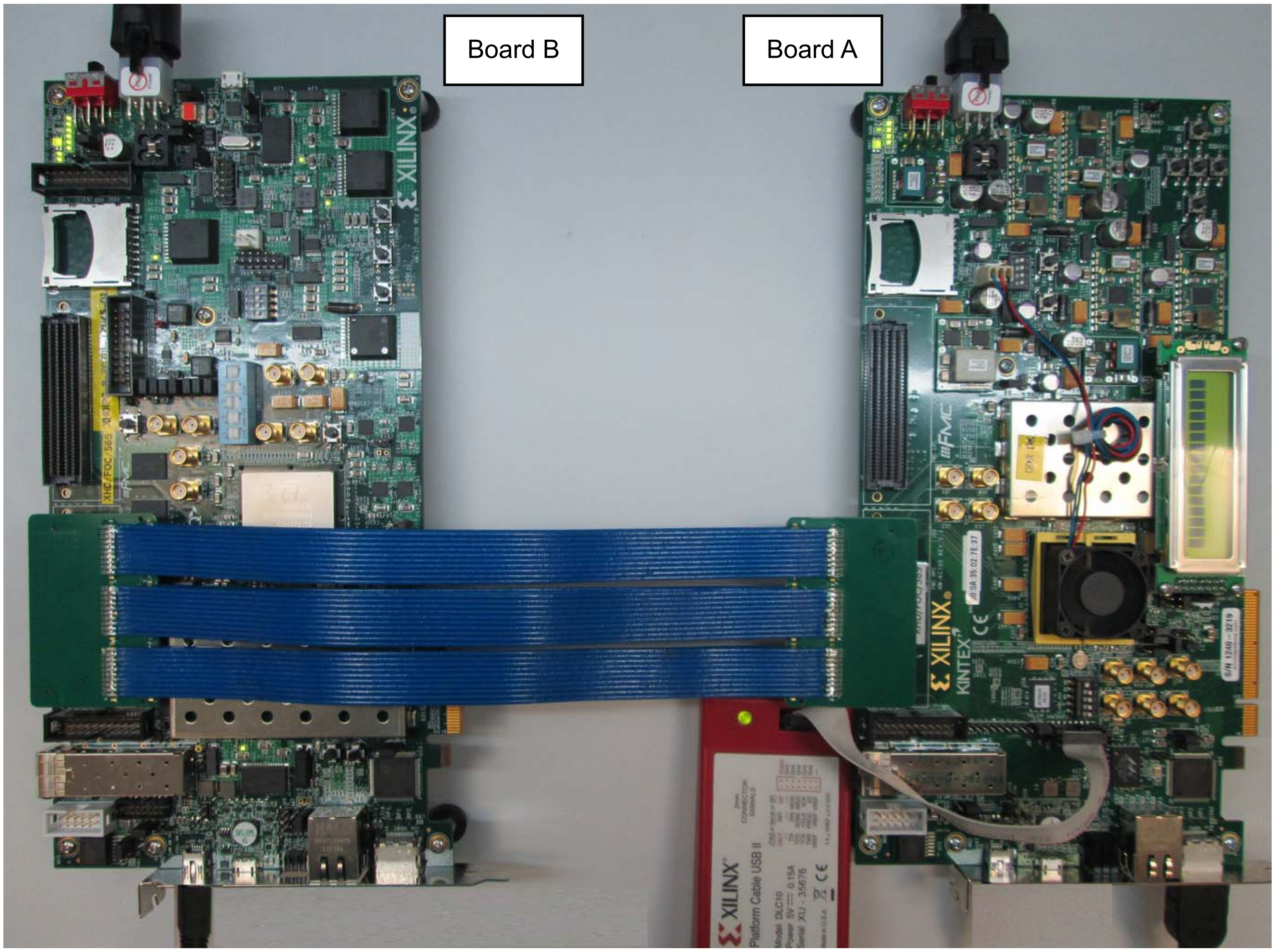


Figure 6: **Kintex-7 FPGA KC705 Board to Zynq-7000 AP SoC ZC706 Board Setup**

2. Connect the KC705 HDMI video output to a video monitor capable of displaying a 720 x 480p 60 Hz video signal.

   Note: This board is Board-A and the other board is Board-B in the following steps.
3. Connect a USB cable from the host PC to the USB UART port on Board-A.
4. Connect the power supply cable to both boards.
5. Set power ON on Board-B.
6. Connect the JTAG cable to Board-B.
7. In the command shell or terminal window, change directories (using a 32- or 64-bit command prompt) to the slave download directory:

```
% cd <unzip_dir>/c2c_selectio/kintex-zynq/ready_to_download/slave
```

8. Start the Xilinx Microprocessor Debugger (XMD):

   ```
   % xmd
   ```

9. Start a terminal program (for example, HyperTerminal) on the host PC with these settings:
   a. Baud Rate: **9600**
   b. Data Bits: **8**
   c. Parity: **None**
   d. Stop Bits: **1**
   e. Flow Control: **None**

10. Source the Tcl file on Board-B:

    ```
    XMD% source slave.tcl
    ```

11. Stop XMD:

    ```
    XMD% exit
    ```

12. Connect the JTAG cable and set power ON on Board-A.

13. Change directories to the master download directory:

    ```
    % cd <unzip_dir>/c2c_selectio/kintex-zynq/ready_to_download/master
    ```

14. Start XMD:

    ```
    XMD% xmd
    ```

15. Download the bitstream file to Board-A:

    ```
    XMD% fpga -f design_1_wrapper.bit
    ```

16. Connect the processor:

    ```
    XMD% connect mb mdm
    ```

17. Disable reset of the entire system on download of the software:

    ```
    XMD% debugconfig -reset_on_run system disable
    ```

18. Reset the processor:

    ```
    XMD% rst -processor
    ```

19. Download the ELF processor code file:

    ```
    XMD% dow app_master.elf
    ```

20. Run the software to execute the reference system:

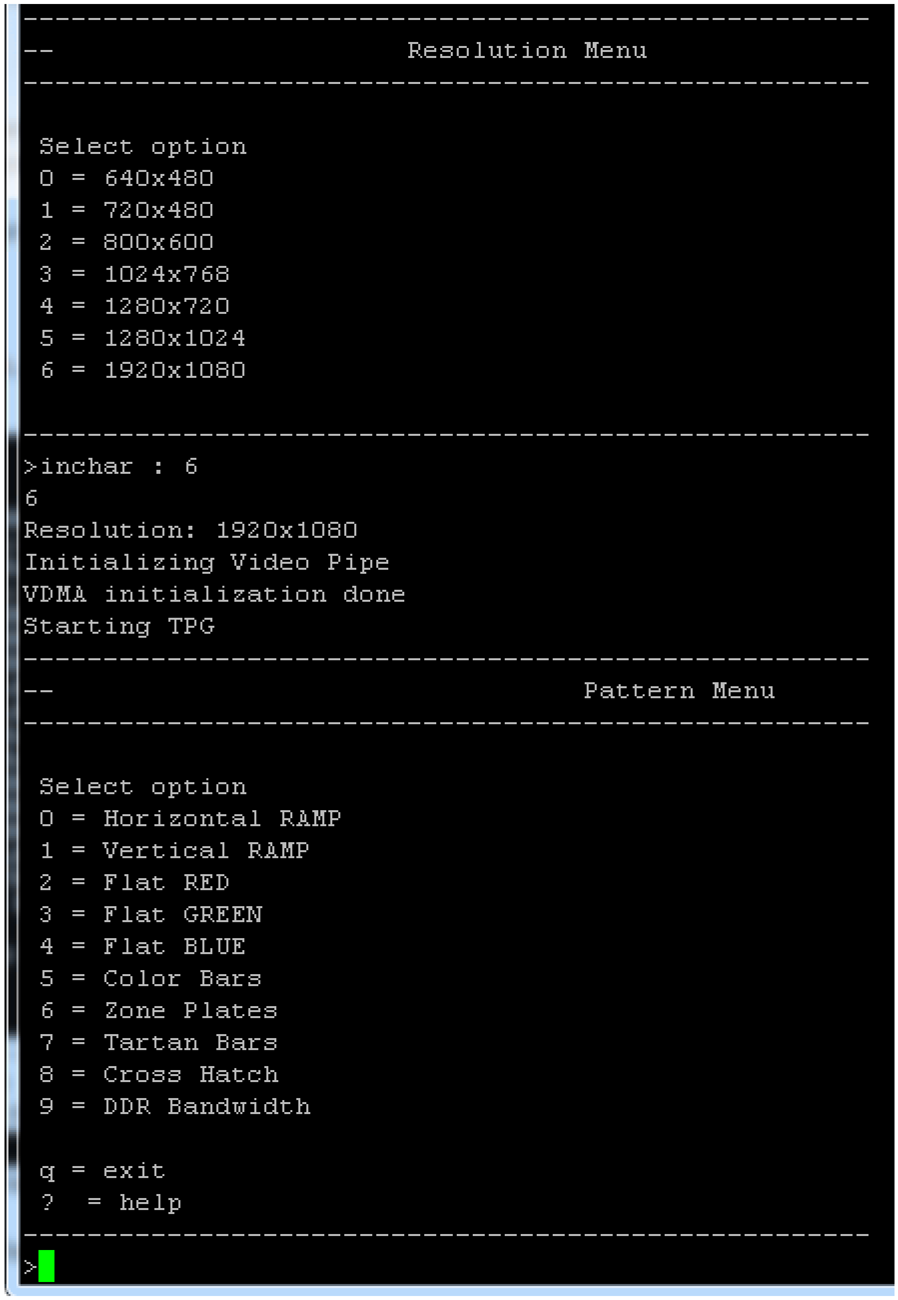
    ```
    XMD% run
    ```

21. Select one of the patterns from the pattern menu.

22. Enter the numbers to see the different patterns in the menu or the DDR bandwidth through the AXI performance monitor.

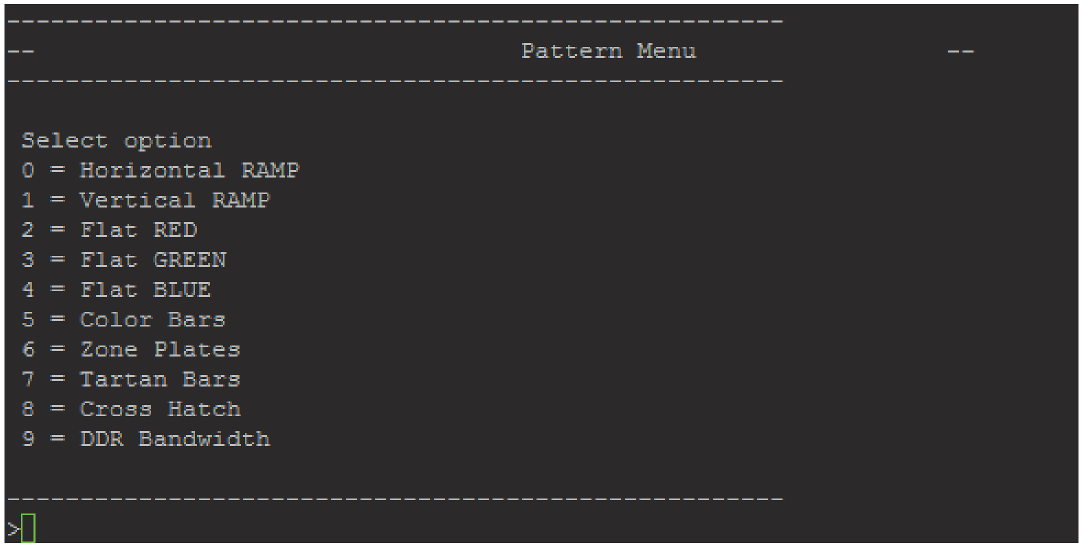
Results from Running Hardware and Software

In the Kintex-7 FPGA KC705 board to Kintex-7 FPGA KC705 board setup, resolutions 640 x 480 to 1920 x 1080 are showcased. In the Kintex-7 FPGA KC705 board to Zynq-7000 AP SoC ZC706 board setup, an AXI data width of 32 bits is chosen for the AXI Chip2Chip configuration and a resolution of 720 x 480 is chosen by default in the software. The HyperTerminal screen displays the output shown in Figure 7 and Figure 8.



X1160_07_051614

Figure 7: **HyperTerminal Menu for Selecting Resolution**



X1160_08_051614

Figure 8: **HyperTerminal Menu for Selecting Video Pattern**

You can choose from eight options displayed on the HyperTerminal screen:

0 – Shows the horizontal ramp on LCD

1 – Shows vertical ramp on LCD

2 – Shows flat red on LCD

3 – Shows flat green on LCD

4 – Shows flat blue on LCD

5 – Shows color bars on LCD

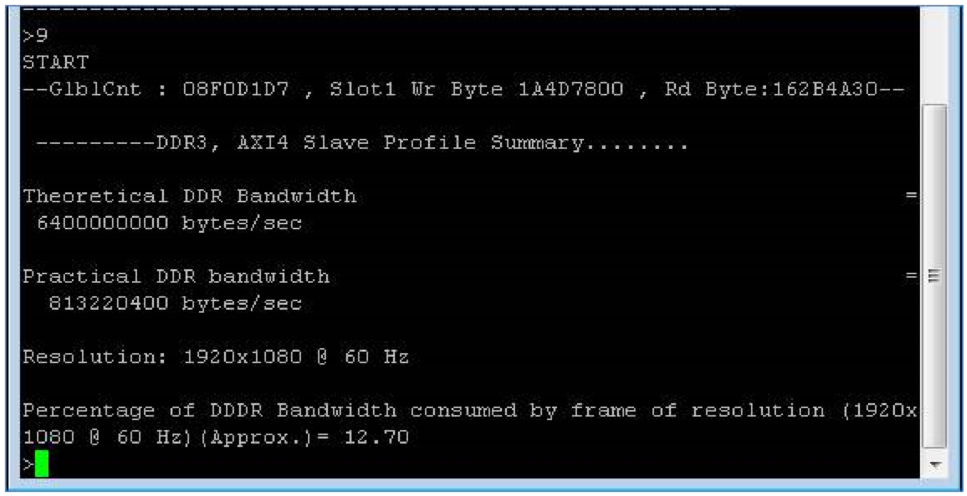
6 – Shows zone plates on LCD

7 – Shows tartan bars on LCD

8 – Shows cross hatch on LCD

9 – Displays performance related metric

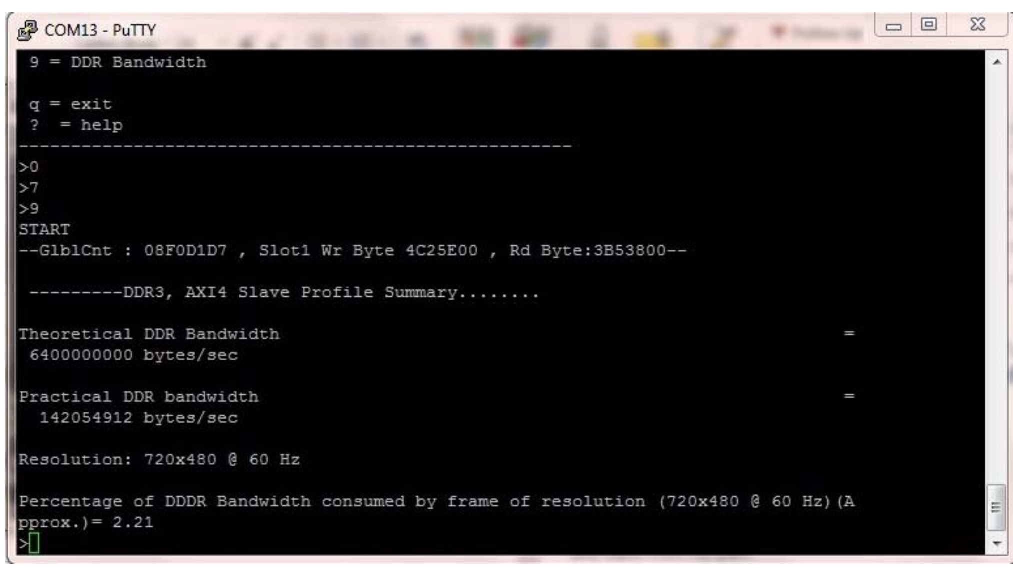
Figure 9 shows the performance data output for the Kintex-7 FPGA KC705 board to Kintex-7 FPGA KC705 board setup. The theoretical bandwidth available in the chosen Chip2Chip configuration is 1.5 GB/s. The bandwidth needed for 1920 x 1080 video traffic is 0.747 GB/s. DDR on a slave system can support a bandwidth of 6.4 GB/s The percentage of available slave DDR bandwidth used is 12.70% (= 0.813/6.4). The numbers in the UART log might vary from the showcased logs.



X1160_09_051614

Figure 9: **Kintex-7 Device to Kintex-7 Device Performance Data**

Figure 10 shows the performance data output for the Kintex-7 FPGA KC705 board to Zynq-7000 AP SoC ZC706 board setup. The theoretical bandwidth available in the chosen Chip2Chip configuration is 0.75 GB/s. The bandwidth needed for 720 x 480 video traffic is 0.125 GB/s. DDR on a slave system can support a bandwidth of 6.4 GB/s. The percentage of available Chip2Chip bandwidth used is 2.21% (= 0.142/6.4). The numbers in the UART log might vary from the showcased logs.



X1160_10_051614

Figure 10: **Kintex-7 Device to Zynq-7000 AP SoC Performance Data**

Rebuilding Hardware Designs

This section describes how to rebuild the hardware designs. Before rebuilding the projects, ensure that the licenses for AXI OSD and AXI Timebase are installed. To obtain evaluation licenses for the AXI Timebase or AXI OSD, see the website for the On-Screen Display LogiCORE IP [Ref 7] or the Video Timing Controller LogiCORE IP [Ref 8].

Rebuilding Master Design

To rebuild the master design:

1. Start the Vivado Design Suite.
2. Open the applicable file:

   For Kintex-7 FPGA:

   `<unzip_dir>/c2c_selectio/kintex-kintex/HW/master/project_1/project_1.xpr`

   For Zynq-7000 AP SoC:

   `<unzip_dir>/c2c_selectio/kintex-zynq/HW/master/project_1/project_1.xpr`
3. Select **Flow** -> **Generate Bitstream** to generate a bitstream for the system.
4. Select **Device Configuration** -> **Update Bitstream** to initialize the block RAM with a bootloop program to ensure the processor boots up with a stable program in memory.

Rebuilding Slave Design

To rebuild the slave design:

1. Start the Vivado Design Suite.
2. Open the applicable file:

   For Kintex-7 FPGA:

   `<unzip_dir>c2c_selectio/kintex-kintex/HW/slave/project_1/project_1.xpr`

   For Zynq-7000 AP SoC:

   `<unzip_dir>/c2c_selectio/kintex-zynq/HW/slave/project_1/project_1.xpr`
3. Select **Flow** -> **Generate Bitstream** to generate a bitstream for the system.
4. Select **Device Configuration** -> **Update Bitstream** to initialize the block RAM with a bootloop program to ensure the processor boots up with a stable program in memory.

Compiling Software and Running Design with SDK

The Xilinx SDK is a software development environment that supports all Xilinx FPGA architectures.

Compiling Software

To compile the software:

1. Start the SDK.
2. Select **Workspace Launcher** -> **Workspace**.
3. Navigate to and select:

   For Kintex-7 FPGA:

   `<unzip_dir>c2c_selectio/kintex-kintex/SW/master/SW/`

   For Zynq-7000 AP SoC:

   `<unzip_dir>/c2c_selectio/kintex-zynq/SW/master/SW/`
4. Click **OK**.
5. Click **Finish**.

The BSP and software applications start to compile.

Note: The process can take up to five minutes.

You can now modify existing software applications and create new software applications using the SDK.

Design Characteristics

The resource utilization of the Kintex-7 FPGA to Kintex-7 FPGA reference designs is listed in Table 3.

Table 3: **Resource Utilization of Kintex-7 FPGA to Kintex-7 FPGA Reference Designs**

Mode	Slice LUTs	Slice Regs	Memory	DSP	I/O	Clocking
Master System	26,650	32,600	81	15	116	10
Slave System	24,976	26,200	113	0	206	10

The resource utilization of the Kintex-7 FPGA to Zynq-7000 reference designs is listed in Table 4.

Table 4: **Resource Utilization of Kintex-7 FPGA to Zynq-7000 AP SoC Reference Designs**

Mode	Slice LUTs	Slice Regs	Memory	DSP	I/O	Clocking
Master System	26,028	32,289	80	15	90	10
Slave System	2,904	4,160	4	0	63	7

Reference Design

The reference designs have been fully verified and tested on hardware boards. The design includes details on the functions of the AXI Chip2Chip IP core. The designs have been successfully implemented using the Xilinx Vivado Design Suite.

The reference design files for this application note can be downloaded from:

https://secure.xilinx.com/webreg/clickthrough.do?cid=202414

Table 5 shows the reference design matrix.

Table 5: **Reference Design Matrix**

Parameter	Description
General	
Developer name	Ravi Kiran Boddu and Pankaj Kumbhare
Target devices (stepping level, ES, production, speed grades)	Kintex-7 FPGA and Zynq-7000 AP SoC
Source code provided	Yes
Source code format	VHDL/Verilog
Design uses code/IP from existing Xilinx application note/reference designs or third-party sources.	Reference designs provided for Vivado IP 2013.4 (see *AXI VDMA Reference Design Application Note* (XAPP742) [Ref 2])
Simulation	
Functional simulation performed	Simulation not supported

Table 5: **Reference Design Matrix *(Cont'd)***

Parameter	Description
Timing simulation performed	Simulation not supported
Test bench used for functional and timing simulations	Simulation not supported
Test bench format	Simulation not supported
Simulator software/version	Simulation not supported
SPICE/IBIS simulations	Simulation not supported
Implementation	
Synthesis software tools/version	Vivado Design Suite 2014.1
Implementation software tools/versions used	Vivado Design Suite 2014.1
Static timing analysis performed?	Yes (passing timing with Vivado implementation tools)
Hardware Verification	
Hardware verified?	Yes
Hardware platform used for verification	Two KC705 boards and one ZC706 board

Utilization and Performance

Table 6 lists the device resource utilization for the master and slave instances of the AXI Chip2Chip IP core in the Kintex-7 FPGA to Kintex-7 FPGA reference design. Table 7 lists the device resource utilization for the master and slave instances of the AXI Chip2Chip IP core in the Kintex-7 FPGA to Zynq-7000 SoC reference design.The information in these tables is from the Design Summary tab in the Vivado Design Suite under the Design Overview > Module Level Utilization report selection. The utilization information is approximate due to cross-boundary logic optimizations and logic sharing between modules.

Table 6: **Module Level Resource Utilization of AXI Chip2Chip Instances in Kintex-7 FPGA to Kintex-7 FPGA Reference Design**

IP Core	Mode	Slice LUTs	Slice Regs	Memory	DSP	I/O	Clocking
AXI Chip2Chip	Master	1,814	2,909	5	0	N/A	3
AXI Chip2Chip	Slave	1,660	2,729	5	0	N/A	3

Table 7: **Module Level Resource Utilization of AXI Chip2Chip Instances in Kintex-7 FPGA to Zynq-7000 SoC Reference Design**

IP Core	Mode	Slice LUTs	Slice Regs	Memory	DSP	I/O	Clocking
AXI Chip2Chip	Master	1,528	2,497	4	0	N/A	3
AXI Chip2Chip	Slave	1,508	2,437	4	0	N/A	3

Note: Slices can be packed with basic elements from multiple IP cores and hierarchies. Therefore, a slice is counted in every hierarchical module that each of its packed basic elements belong to, which results in some double counting of slice counts when adding up the slice counts across modules.

The Kintex-7 FPGA to Kintex-7 FPGA board setup uses the 64-bit AXI configuration in compact 1:1 DDR mode with the physical layer at 250 MHz. The AXI Chip2Chip core should be configured so that its theoretical throughput (see Equation 1) is higher than the average traffic sent as input to the master AXI Chip2Chip core:

$$\frac{3 \times AXIDataWidth}{4 \times MuxingRatio} \times PHYFrequency \qquad \text{Equation 1}$$

For example, for a 32-bit AXI data width configuration with compact 1:1 and DDR PHY type, and the PHY operating at 250 MHz, the theoretical throughput of the core is 750 MB/s. Thus, this configuration cannot support a frame resolution of 1920 x 1080, which needs a bandwidth of 0.995 GB/s. For the ZC706 board, the available number of I/O pins in the FMC HPC connector is less than that needed for the 64-bit compact 1:1 DDR mode. Consequently, a lower resolution of 720 x 480 is showcased with 32-bit configuration of the AXI Chip2Chip core.

Note: Muxing ratio indicates the Chip2Chip PHY width parameter, which is 1 for compact 1:1, 2 for compact 2:1, and 4 for compact 4:1. Equation 1 applies to systems with burst length=1. The design implements a priority encoding scheme for multiplexing AW, AR, and W (or AR and B) data of the AXI4 interface. That is, for systems with a larger burst length, when a defined slot is empty, data from an available channel is multiplexed and transmitted (for example, if the AW and AR channels do not have data, then data from the W channel is transmitted). Hence, the theoretical bandwidth for systems with larger burst lengths will be better than the value in Equation 1 and the ¾ factor could be ignored.

Table 8 lists the number of input and output I/Os needed for the different Chip2Chip core configurations. The shaded rows show the selected configurations for the reference design setup. If a lower data rate or pin count is needed, Table 8 and Equation 1 can be used to determine the appropriate configuration.

Table 8: **FPGA I/O Utilization for Different Configurations of AXI Chip2Chip Core**

AXI Data Width	Chip2Chip PHY Type	Muxing Ratio	Number of Input I/Os	Number of Output I/Os
32	SelectIO SDR	Compact 4:1	19	19
		Compact 2:1	31	31
	SelectIO DDR	Compact 4:1	10	10
		Compact 2:1	16	16
		Compact 1:1	29	29
64	SelectIO [technology, interface] SDR	Compact 4:1	26	26
		Compact 2:1	45	45
	SelectIO DDR	Compact 4:1	14	14
		Compact 2:1	23	23
		Compact 1:1	42	42

Note: The highlighted 32-bit configuration is chosen for the Kintex-7 FPGA to Zynq-7000 SP SoC reference design and the highlighted 64-bit configuration for the Kintex-7 FPGA to Kintex-7 FPGA reference design.

References

This document uses the following references:

1. PG067, *AXI Chip2Chip Product Guide*
2. XAPP742, *AXI VDMA Reference Design Application Note*
3. UG1037, *Vivado Design Suite: AXI Reference Guide*
4. UG995, *Vivado Design Suite Tutorial: Designing IP Subsystems Using IP Integrator*
5. UG994, *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator*
6. UG586, *Zynq-7000 SoC and 7 Series Devices Memory Interface Solutions User Guide*
7. On-Screen Display LogiCORE IP
   www.xilinx.com/products/intellectual-property/EF-DI-OSD.htm
8. Video Timing Controller IP
   www.xilinx.com/products/intellectual-property/EF-DI-VID-TIMING.htm

9. FMC connector cable
   www.samtec.com/standards/vita.aspx

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
03/07/2013	1.0	Initial Xilinx release.
07/03/2013	2.0	Added AXI-Lite support. Updated for software moving from the master board to the slave board. Updated ISE Design Suite to 14.5. Updated Figure 2 description. Updated Figure 1, Figure 2, Figure 3, Figure 4, and Figure 5. Updated Table 1, Table 2, Table 3, Table 4, and Table 5. Added Figure 9, Figure 10, Table 6, Table 7, and Table 8. Updated Reference Design Specifics, Hardware System Specifics, Software Applications, Kintex-7 FPGA KC705 Board to Kintex-7 FPGA KC705 Board, Kintex-7 FPGA KC705 Board to Zynq-7000 AP SoC ZC706 Board, and Design Characteristics.
07/03/2014	3.0	Updated Figure 1, Figure 2, Figure 3, Figure 4, Figure 7, Figure 8, and Figure 10. Updated Table 1, Table 2, Table 3, Table 4, Table 5, Table 6, and Table 7. Updated Introduction, Hardware and Software Requirements, Reference Design Specifics, Hardware System Specifics, Software Applications, Executing the Reference Design, Rebuilding Hardware Designs, Compiling Software and Running Design with SDK, Design Characteristics, Reference Design, and Utilization and Performance. Updated from ISE Design Suite to Vivado Design Suite throughout. ***Note:*** This application note is no longer applicable to the ISE Design Suite.

Notice of Disclaimer