XILINX®

WP417 (v1.1) July 22, 2013

Maximize System Performance Using Xilinx Based AXI4 Interconnects

By: Vinay Singh and Khang Dao

Delivering maximum performance FPGA/SoC solutions with the lowest total design cost often rests with the ability to rapidly combine and configure numerous design elements, such as on-chip logic blocks, processor cores, and a wide range of Xilinx and third-party IP design elements.

Xilinx® Plug-and-Play IP, together with its high performance, device-optimized, on-chip AXI4 interconnects, provides a simple yet powerful capability that can connect single or multiple sets of master and slave design blocks with a minimal amount of design effort.

Introduction

With today's ever-increasing design complexity, system designers are faced with numerous challenges — as well as many potential solutions. Key strategic design decisions made early in the development cycle often have far-reaching impact on the overall results achieved with the final design.

Some of the more important system design decisions include:

- Selecting FPGA/SoC product families with the best fit of on-chip processing cores
- Identifying a set of off-the-shelf IP blocks to be used with the design
- Implementing robust hardware interconnect solutions for both on-chip and external hardware resources

Xilinx offers many product solutions with single and multiple on-chip processing elements, such as multiple MicroBlaze™ processors or ARM® Cortex™-A9 processors, DMA engines, communications, video, and DSP IP. Evaluation Kits and Targeted Design Platforms are available to serve both as sample system designs and as advanced development platforms for the underlying FPGA/SoC selection. Understanding the breadth and depth of the surrounding ecosystem support can help reduce development time and associated design risks.

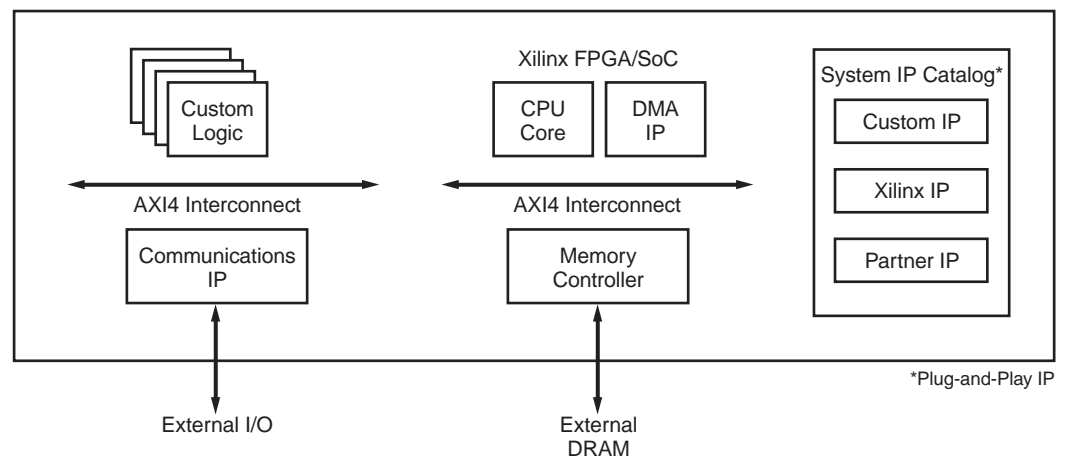Figure 1 shows a typical Xilinx FPGA/SoC-based design.

WP417_01_071613

Figure 1: **Typical Xilinx FPGA/SoC-Based Design**

System designers can also build their own custom IP blocks that complement their pre-built IP selections. After the basic design architecture has been chosen, the system designer must assemble all of the blocks together in the most efficient manner possible. To accelerate this process, Xilinx has adopted the industry-standard IP interface AMBA® Advanced eXtensible Interface 4 (AXI4) as a key element of the Plug-and-Play IP initiative.

Xilinx Plug-and-Play IP

To meet customer demand for a wide variety of IP cores for many markets and applications, Xilinx introduced the Plug-and-Play IP initiative to create an open and scalable infrastructure that employs a standard interface and tool flow for the development and deployment of IP. The Plug-and-Play IP initiative is the Xilinx response to the growing customer demand for system-level design using multiple IP cores from multiple sources. See WP379, *AXI4 Interconnect Paves the Way to Plug and Play IP.*

By using AXI4 as a standard interface, Xilinx Plug-and-Play IP help customers remove the overhead of integrating IP with different incompatible interfaces. The benefits to customers include:

- Increased productivity: The AXI4 specification eliminates the need for multiple legacy or custom interfaces to integrate IP from various sources. Because all IP shares a common interconnect, the designer can easily remove, add, or replace IP blocks within a design.

- Greater flexibility: The AXI4 specification accommodates a range of system requirements. It is inherently scalable, enabling system designers to optimize their designs for the highest possible F_{MAX}, maximum throughput, lower latency, smaller area, or some combination of those attributes. Xilinx provides a rich portfolio of AXI4 Interconnect and related infrastructure IP for customers to implement optimized systems for their specific applications.

- Greater IP availability: The AXI4 specification encourages and enables the Xilinx IP ecosystem and the ARM IP ecosystem to efficiently develop IP for use in Xilinx's FPGAs and All Programmable SoCs.

- Higher Quality IP: By leveraging industry-standard verification IP and methodologies around AXI4 interface, IP providers and customers can accelerate production of high quality, pre-verified IP or design components. This results in faster design validation and reduces system debug effort enabling shorter development cycles. To further assist customers and IP providers for creating and verifying IP with AXI4 interfaces, Xilinx also provides AXI Bus Functional Models (BFMs). See DS824, *AXI Bus Functional Models Data Sheet*. The BFMs, developed for Xilinx by Cadence Design Systems, support the simulation of custom-designed AXI masters and slaves using the AXI3, AXI4, AXI4-Lite, or AXI4-Stream interface. The BFMs are delivered as encrypted Verilog modules and can be simulated with Aldec Riviera-PRO, Cadence Incisive Enterprise Simulator, ISE® Design Tool Simulator, Vivado® Simulator, Mentor Graphics ModelSim, and Synopsys VCS.

Xilinx Plug-and-Play IP, ISE Design Suite, and Vivado Design Suite tools (shown in Figure 2) help designers assemble the various IP blocks into a system. The design tools enable customers and IP providers to provide access to pre-verified, high-quality IP blocks as part of an IP catalog for system designers.
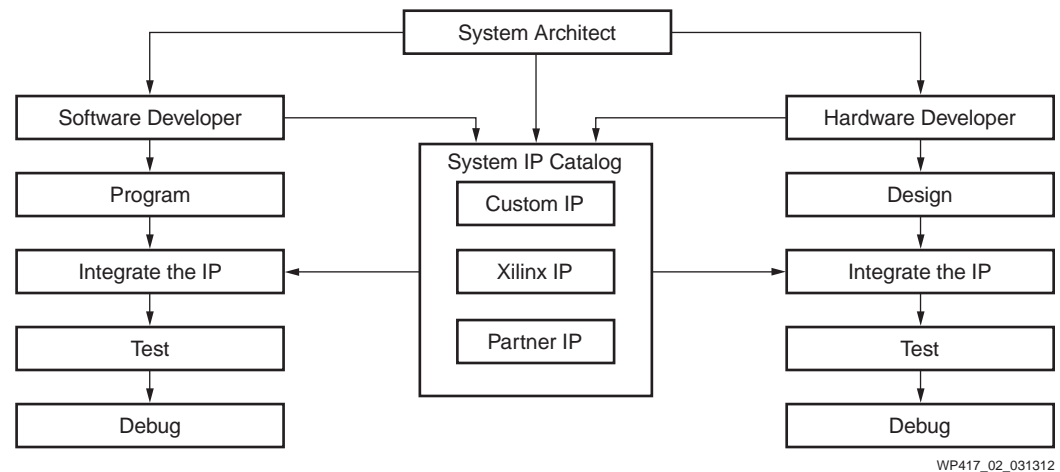
Figure 2: **System-Level IP Design Flow**

Most providers of the IP blocks pre-verify their designs, thus, helping the downstream system designers save additional verification effort. For IP blocks with a large number of underlying logic design and interfaces, this design time savings can be extremely valuable.

Sharing a common IP catalog and a standard interface ensures that the software and hardware design teams can build sub-systems and integrate them into larger systems. It also means that updates to any of the IP cores or the sub-systems can be readily assimilated into existing as well as new system designs.

Xilinx design tools such as Xilinx Platform Studio (XPS) and Vivado IP Integrator are designed to automate assembly of IP blocks and interconnect with the AXI4 interface into a system. XPS provides a graphical system assembly tool for connecting blocks of IPs together using AXI interface level connections to implement systems, with or without processors. XPS provides native support for AXI Interconnect IP, allowing customers to explore and tailor system topologies to their application requirements.

The Xilinx Plug-and-Play IP with ISE Design Suite and Vivado Design Suite tools provides a powerful design platform for system architects, hardware developers, and software developers.

AMBA AXI4 Overview

The AXI4 specification, based on the ARM, Ltd. Advanced Microcontroller Bus Architecture (AMBA), represents a major evolutionary step in interconnect technology for on-chip system design. AMBA was first introduced in 1996 to address a range of growing on-chip interconnect challenges. AMBA defined a set of open-standard protocols and interfaces used for on-chip interconnects including AXI.

The AMBA 4 and AXI4 specifications, introduced in March, 2010, were designed by and for the industry, with contributions from 35 of the industry's leading OEM, EDA, and semiconductor vendors, including Xilinx. The result is an industry-standard IP interface and interconnect architecture suitable for FPGAs/SoCs and ASICs. Xilinx introduced support for all AXI4 interfaces in ISE Design Suite v12.3. See Figure 3.
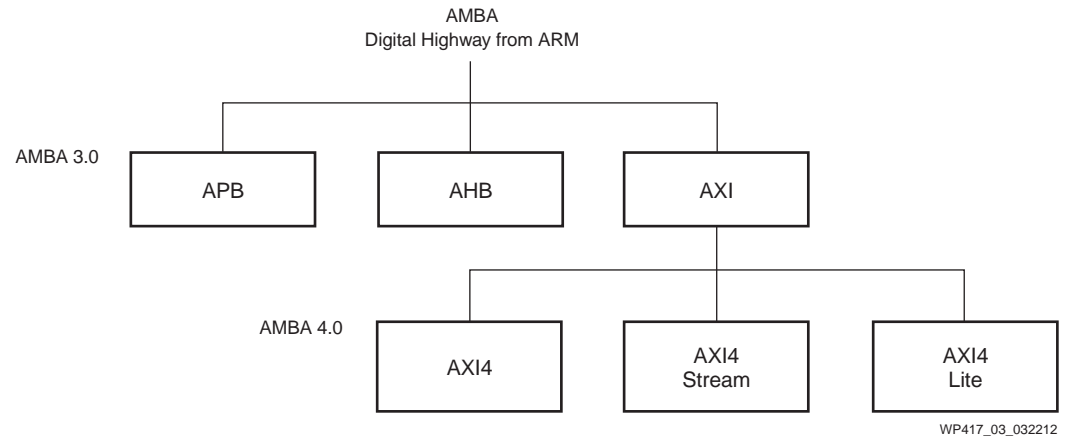
Figure 3: **AMBA AXI Family Tree**

Today's fourth generation of AXI, known as AXI4, is the direct result of a partnership effort between Xilinx and ARM. AXI4 comes in three distinct flavors:

- **AXI4 (Memory-Mapped)**, a robust memory-mapped interface designed to achieve maximum levels of on-chip performance. Allows variable bursts up to 256 data transfers per single address transfer.

- **AXI4-Lite**, a lightweight, single-transaction memory-mapped interface. It is a smaller logic footprint, a subset of the AXI4 interface, used for accessing control registers and low-performance peripherals.

- **AXI4-Stream**, used for high-speed streaming applications that do not require an address. Data burst size can be unlimited.

Xilinx supports IP with all three forms of AXI4 interfaces, but for the remainder of this white paper, the focus is on the full memory-mapped form, known simply as "AXI4." The AXI4 interface consists of five separate channels, as shown in Figure 4.
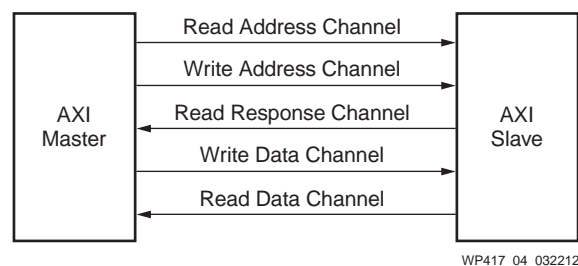

Figure 4: **Standard AXI4 Master-Slave Interface**

Data can move in both directions between the master and slave simultaneously, and data burst lengths can vary up to 256 data transfers. For an AXI4 Read transaction, the master device issues information to the slave indicating the source location and data burst length. Pipelining support allows a master to request multiple Read transfers before the slave has completed even the first Read Data response.

AXI4 Write cycles operate in a similar fashion, using a separate set of Write Address and Write Data channels. An AXI4 Write Response channel is used to communicate completion information so that write transfers can be pipelined and tracked throughout the system to improve data throughput.

A set of AXI4 read or write transactions can be grouped and tagged by IDs. Transactions with the same ID (separate for read and write transactions) are

completed in the order they were issued. Transactions between different IDs can be completed "out of order." ID usage and out-of-order execution support is optional and can be tailored according to the capabilities and system requirements of master, slave, and interconnect IP blocks.

Note: The processing system block in the Zynq®-7000 devices utilize AXI3 memory mapped interfaces. AXI3 is a subset of AXI4 and Xilinx tools automatically insert the necessary adaptation logic to translate between AXI3 and AXI4.

AXI4 Interconnect Overview

In its most basic form, an AXI4 interface connects one master to one slave device. For device and system architectures that require more than one master or more than one slave, Xilinx provides AXI Interconnect and infrastructure IP to manage communication between them. While traditional interfaces specify a bus-style shared datapath for communication, AXI4 provides greater communication flexibility by defining the interface as point to point. This freedom within the specification paves the way for a multitude of Interconnect architectures that can scale to deliver far greater system performance than system implementations using traditional shared-bus designs.

Xilinx's specific device-optimized AXI Interconnect IP is designed to deliver excellent performance and scalability without requiring the system designer to implement and verify all the commonly-used building blocks of an AXI4-compliant interconnect. AXI4 Interconnect provides switches, arbiters, buffers, decoders, adapters, transaction management logic, etc. This allows system designers the time to focus on their application's system topology and IP assembly rather than designing and optimizing the logic required for managing basic AXI4 traffic.

Xilinx's AXI4 Interconnect is highly configurable to allow users to balance and tune for their design goal, i.e., area, timing, throughput, and latency. This flexibility of implementation enables system designs around AXI4 to be scalable, from smaller devices needing only low or medium performance to larger devices requiring high performance across large numbers of interconnected IP blocks.

The AXI4 protocol allows master or slave devices to choose from a range of characteristics, such as data flow control, data widths, burst lengths, clock domains, etc. To implement heterogeneous systems, the AXI Interconnect features a range of converters (buffers, data widths, protocols, clocks, etc.) for enabling plug-and-play interfacing among master and slaves with differing interface characteristics.

For example, Figure 5 shows a system implementing an AXI4 IP-based Multiport Memory Controller (MPMC), connecting two master devices of 32-bits at 50 MHz and 64-bits at 100 MHz data width, sharing a single 64-bit-wide, 150 MHz, DDR3-based external memory controller. To implement this system, the AXI Interconnect is automatically configured to perform width and clock conversion.

To take full advantage of the memory controller's throughput, the interconnect also needs to be configured to employ buffering at both master side interfaces of the interconnect. Without buffering, the slower master reduces the data rate of the memory controller's datapath during its transfers, reducing overall memory bandwidth utilization.
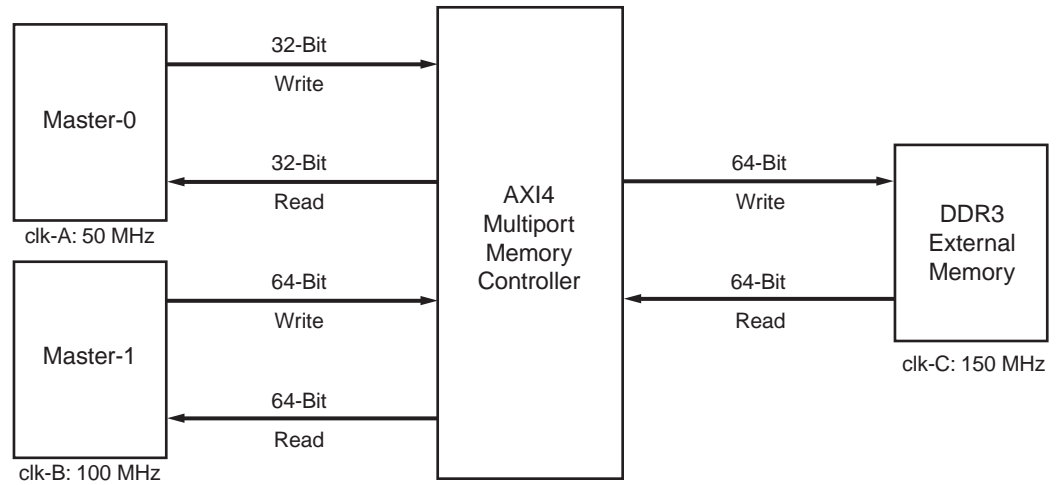
Figure 5: **AXI4 Dual-Master, Single-Slave Shared Memory Interface**

When larger numbers of masters and slaves are connected to a common AXI4 Interconnect, the resulting congestion and resource utilization can reduce overall system performance and increase latency. To better implement such systems, customers can employ a topology of multiple, simpler AXI4 Interconnects. Multiple interconnects can also be applied to enable a greater level of parallelism than with a single interconnect solution.

Figure 6 illustrates a system involving seven master devices and seven slave devices, where every master does not need to communicate with every slave.
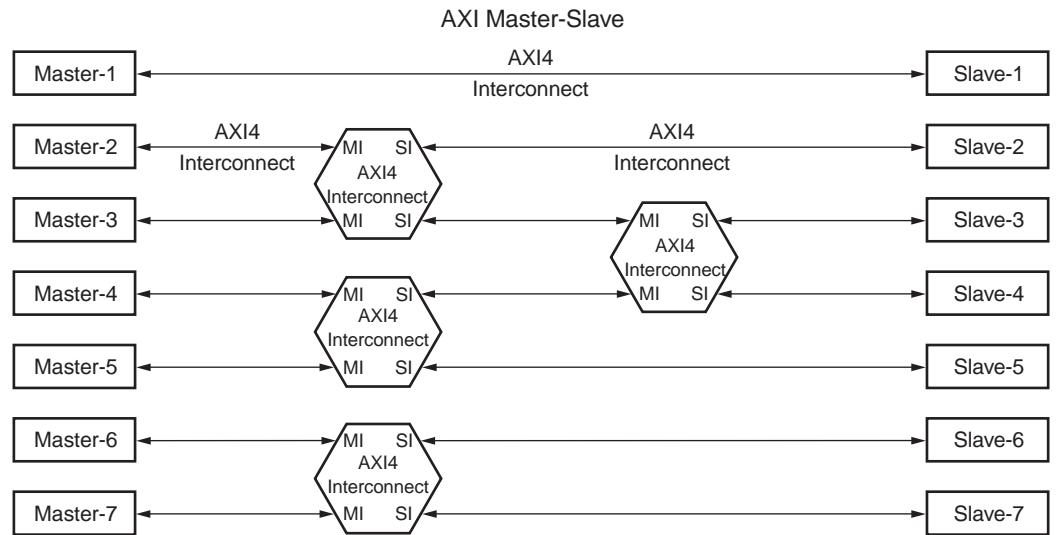
Figure 6: **Complex Multi-Master, Multi-Slave AXI4 Fabric Switching**

Master-1 only communicates with Slave-1 and can, therefore, connect point-to-point. Masters 6 and 7 only need to communicate with Slaves 6 and 7 and can, thus, use a single 2-by-2 AXI Interconnect. Masters 2 through 5, along with Slaves 2 through 5, use a cascaded series of three 2-by-2 AXI4 Interconnects to balance their communication and performance requirements.

Given the flexibility of the AXI4 Interconnect, IP designers can choose a hierarchy of cascaded interconnects rather than a single monolithic interconnect (i.e., three tailored 2-by-2 interconnects in the previous example vs. one less efficient 4-by-4 interconnect). The system architect must still pay attention to the number of interconnects traversed between a given master-slave pair to meet system latency requirements.

Along with Plug-and-Play IP, the Xilinx design tools ensure that the MPMC example or systems with multiple interconnects can be scaled to integrate additional standard AXI4-compatible blocks. The development of a custom memory controller with one or more interconnect switches is no longer needed. The overall design and development time can be substantially reduced for designs that rely on numerous master devices sharing memory or memory-mapped spaces!

AXI4 Interconnect Performance Trade-Offs

High performance AXI4 designs can employ pipelining and longer burst length to improve system throughput. However, this must be balanced against master and slave design effort and resource utilization.

Figure 7 and Figure 8 show the system throughput results when three masters and three slaves utilize different pipelining and burst lengths characteristics across a common AXI4 Interconnect. In Figure 7, the datapath throughput using a 4-beat burst length cannot fill more than half the possible bandwidth, even when transactions are pipelined. This is caused by the congestion of servicing three masters in the interconnect's address channel control logic. Changing the data burst length to 16, as shown in Figure 8, allows interconnect utilization to reach over 70% when transactions are not pipelined, increasing to 100% when transactions are pipelined. Using longer burst lengths allows a greater ratio of data transfers per address transfers. Using pipelining allows the subsequent address transfers to occur during previous data transfer phases. Therefore, effective use of pipelining with longer burst lengths improves the throughput of the system.
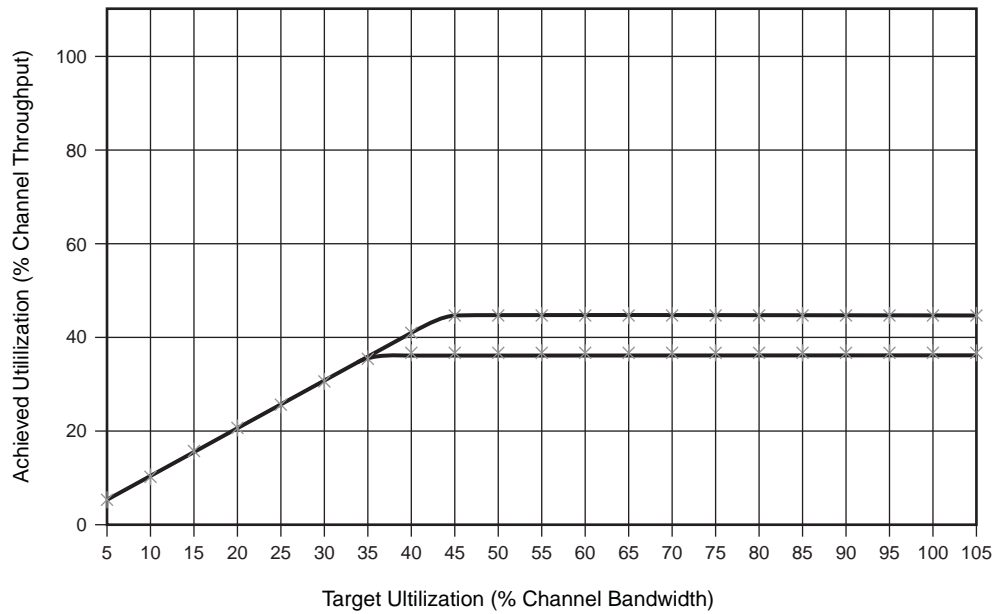
Figure 7: **Interconnect Utilization (Burst Length = 4)**
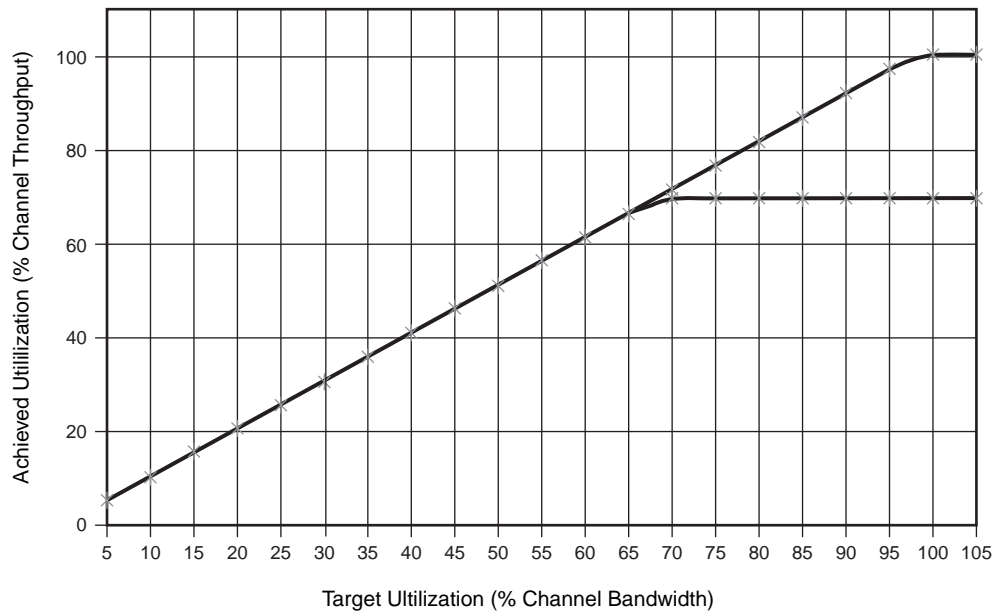
Figure 8: **Interconnect Utilization (Burst Length = 16)**

AXI4 systems that do not use pipelining (which simplifies the design) or shorter burst lengths might not be able to reach the desired level of performance. However, implementing longer burst lengths might not match the native length of data, thus, incurring added logic to form the longer bursts.

There are other considerations for tuning and optimizing AXI4 systems, for example, matching clocks and data widths across the system. Users must validate a system design implementation to see that performance requirements are met when more complex AXI4 design architectures are employed. For more information about AXI4

system optimization, refer to the AXI System Optimization: Tips and Hints chapter in UG761, *AXI Reference Guide*.

These trade-off examples demonstrate the importance of designing end-point masters and slaves to employ the best possible data transfer characteristics, matching them to the topology of the AXI4 Interconnects to achieve overall system latency, performance, and area requirements.

An AXI4-Based Video Application Example

Video processing designs can take advantage of AXI4 Interconnects. Video processing applications that use frame buffers need access to a shared external memory to store data between processing stages (see Figure 9). AXI4 Interconnects and associated AXI4 IP, such as Video DMA (VDMA) engines and DDR3 Memory Controllers, enable systems to be quickly assembled. The user can then leverage the flexibility and scalability of the AXI based system to tailor it for the processing, frame buffering, and performance requirements. In a typical video system architecture, the primary video stream enters at the Source block on the left, and exits at the I/O interface on the right side — possibly as an external DVI PHY, DisplayPort, or HDMI™ connection, or possibly as a simple LVDS panel interface using the standard I/O interface support provided by the Kintex®-7 FPGA. Intermediate processing stages can mix on-chip line buffers with off-chip frame buffers to manage data flow between the stages.
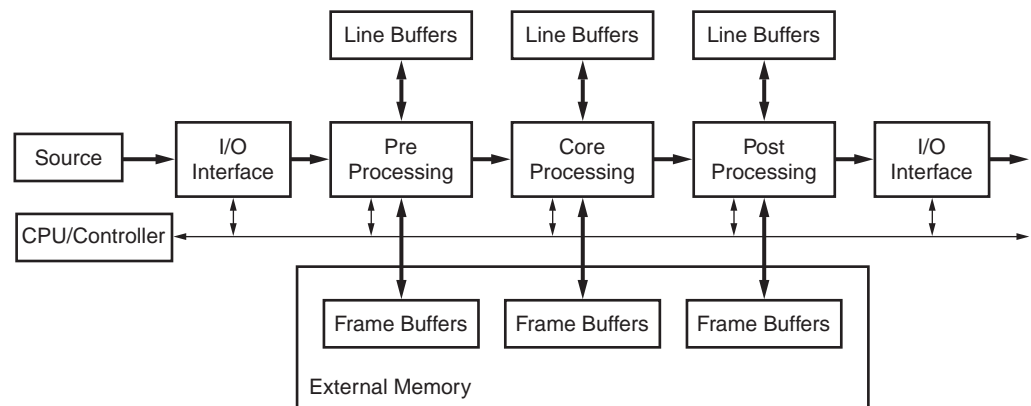

WP417_09_032212

Figure 9: **Advanced Video System Architecture**

In the following example, 16 streams of 1920x1080 60/75p video information with up to 32 bits/pixel can be handled by a single 64-bit DDR3 interface in the Kintex-7 FPGA. As shown in Figure 10, this video design is built around several key AXI4 Interconnects:

- Two cascaded AXI4 Interconnects (AXI4-MM0/AXI4_MM1) are used by the on-chip MicroBlaze processors, eight VDMA blocks, and a DDR3 memory controller to implement an AXI-based MPMC to be used as various frame buffers for each of the video streams.

- Two AXI4-Lite Interconnects (AXI_Lite, AXI_Lite_Video) are used to connect the video IP control register interfaces and peripherals to the MicroBlaze processor controlling the system.

In addition, this application also uses a series of Test Pattern Generators (TPGs) as well as a Video Timing Control (VTC) block to generate video data traffic and video timing controls for the system. All of the video streams are multiplexed via the On-Screen

Display (OSD) block, and from there to the external HDMI video PHY. An attached AXI4 Performance Monitor is used to analyze AXI4 traffic performance.
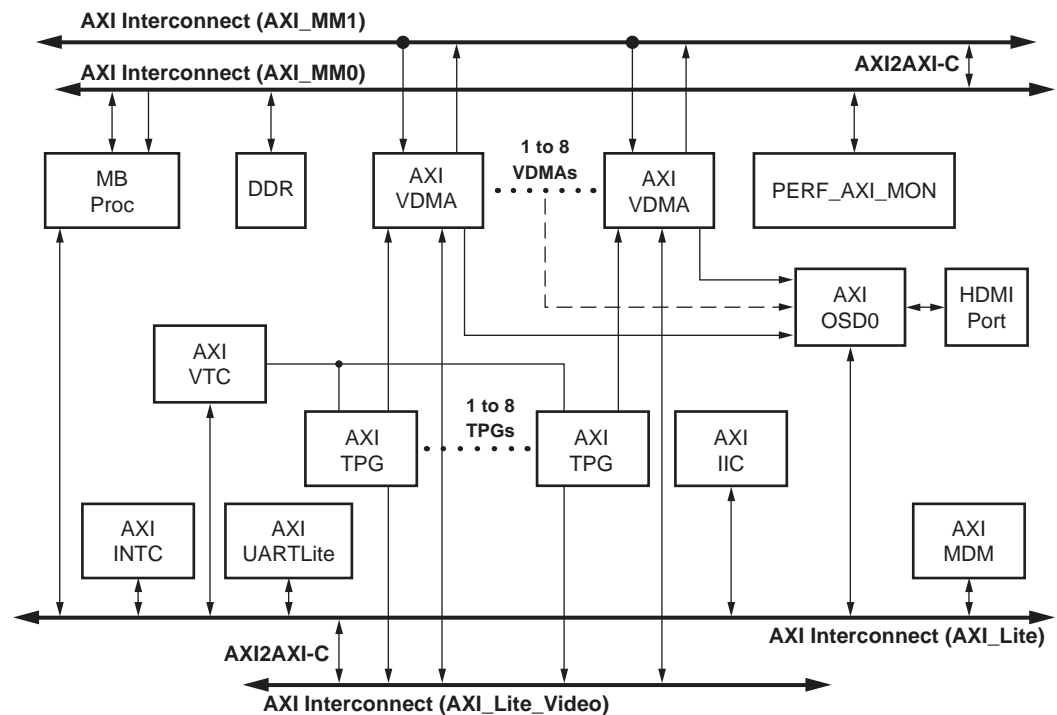
Figure 10: **Video Application AXI4 Topology (Kintex-7 FPGA Based)**

Each 1920x1080 75p video stream can operate with 32-bit pixels, forming a stream that runs at ~622 MB/s (~ 5 Gb/s). Video traffic generated by the eight TPG IP cores creates an aggregate read/write bandwidth (in and out of the frame buffers), equivalent to 16 streams through memory, or nearly 9.95 GB/s (80 Gb/s)! This 18-port shared memory controller is used to support the 16 VDMA channels (each with one Read and one Write interface) plus the data and instruction cache master interfaces on the MicroBlaze processor.

The DDR3 PHY is configured for a 64-bit native data width with a memory clock speed of 800 MHz (1,600 MHz data rate).The primary internal AXI4 slave interface of the memory controller is operated at 200 MHz, using an AXI4 data width of 512 bits, supporting a theoretical maximum data bandwidth of 12.8 GB/s. The system implementation shows the total bandwidth of this design measures out at ~9.95 GB/s. This amounts to roughly 78% of the theoretical maximum, all using off-the-shelf AXI4 IP blocks!

For more information on the configuration and operation of the VTCs, TPGs, and OSD in this application, see XAPP741, *Designing High-Performance Video Systems with the AXI Interconnect.*

Targeted Design Platform Using AXI4

Xilinx development kits provide out-of-the box design solutions that significantly cut development time and enhance productivity. Xilinx takes it one step further with Targeted Design Platforms – the industry's most comprehensive development kits. The kits come complete with evaluation boards, Vivado Design Suite tools, IP cores, reference designs, and FPGA Mezzanine Card (FMC) support. In essence, a Targeted Design Platform serves as a higher-order evaluation vehicle, in many cases coming pre-configured with the design elements that are to be used in the customers' final design.

For designers wishing to explore AXI4 operation and performance, the Kintex-7 FPGA KC705 Evaluation Kit is an excellent starting point (Figure 11). See the Xilinx Kintex-7 FPGA KC705 Evaluation Kit page: http://www.xilinx.com/products/boards-and-kits/EK-K7-KC705-G.htm.
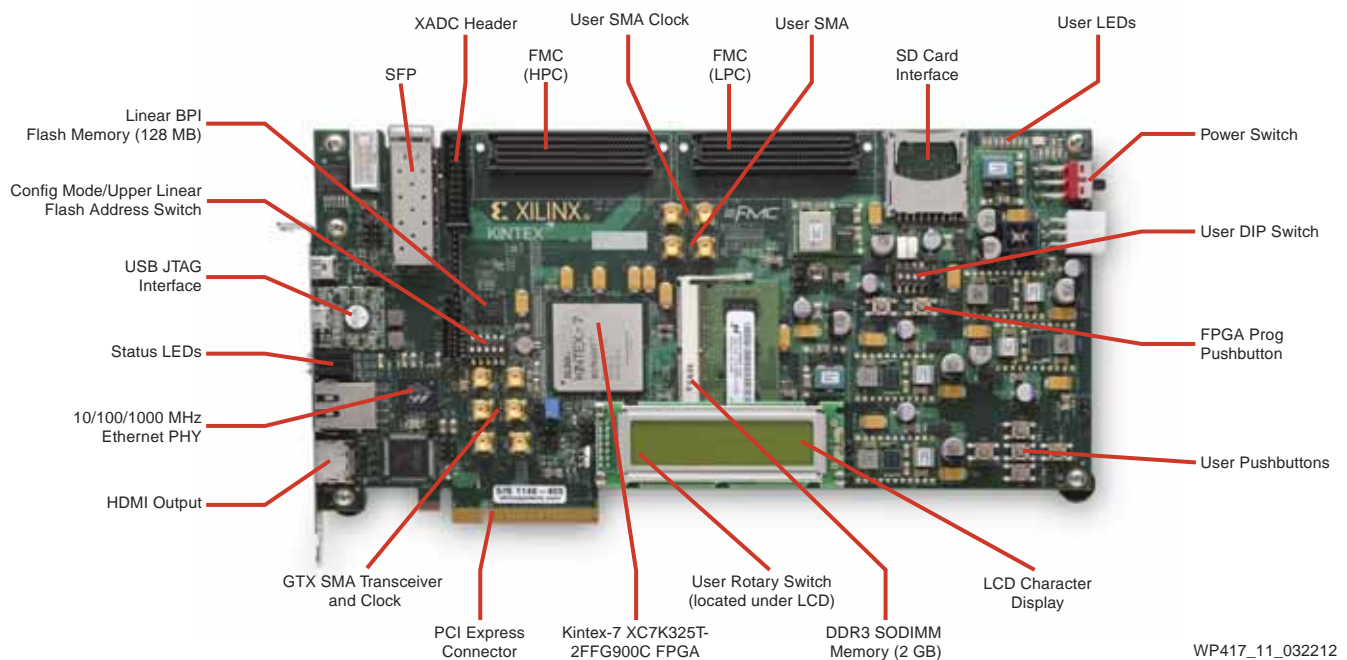
Figure 11:   **Kintex-7 FPGA KC705 Evaluation Board**

As shown in Figure 11, this evaluation kit comes equipped with an excellent hardware platform, surrounding the Kintex-7 FPGA with the interfaces and controls often used in full system solutions:

- I/O Interfaces: FMC connectors; USB/JTAG; Ethernet; HDMI connector; PCI Express®; SD Card; SFP
- LCD Character Display, Status LEDs, switches, and pushbuttons
- DDR3 SODIMM

The KC705 Evaluation Board comes pre-configured with a base targeted reference design, as shown in Figure 12. Also see UG882, *Kintex-7 FPGA Base Targeted Reference Design.* This Targeted Reference Design demonstrates a number of features including:

- Integrated Endpoint block for PCI Express
- Northwest Logic's Packet DMA accessing a Xilinx MPMC using memory controller IP and AXI4 Interconnect
- Multiple virtual FIFOs (VFIFO), connected to memory using AXI4 Interconnects

The Kintex-7 FPGA's integrated Endpoint block for PCI Express and the packet DMA are responsible for the data transfers between the host system and the Endpoint card. Data to and from the host are stored in a virtual FIFO built around the DDR3 memory. This multiport virtual FIFO abstraction layer around the DDR3 memory is used to move traffic efficiently without the need to manage addressing and arbitration on the memory interface. It also provides a larger depth when compared to storage implemented using on-chip memory.

By using the AXI4 IP-based MPMC, the Targeted Reference Design system is capable of sustaining up to 10 Gb/s throughput end-to-end.
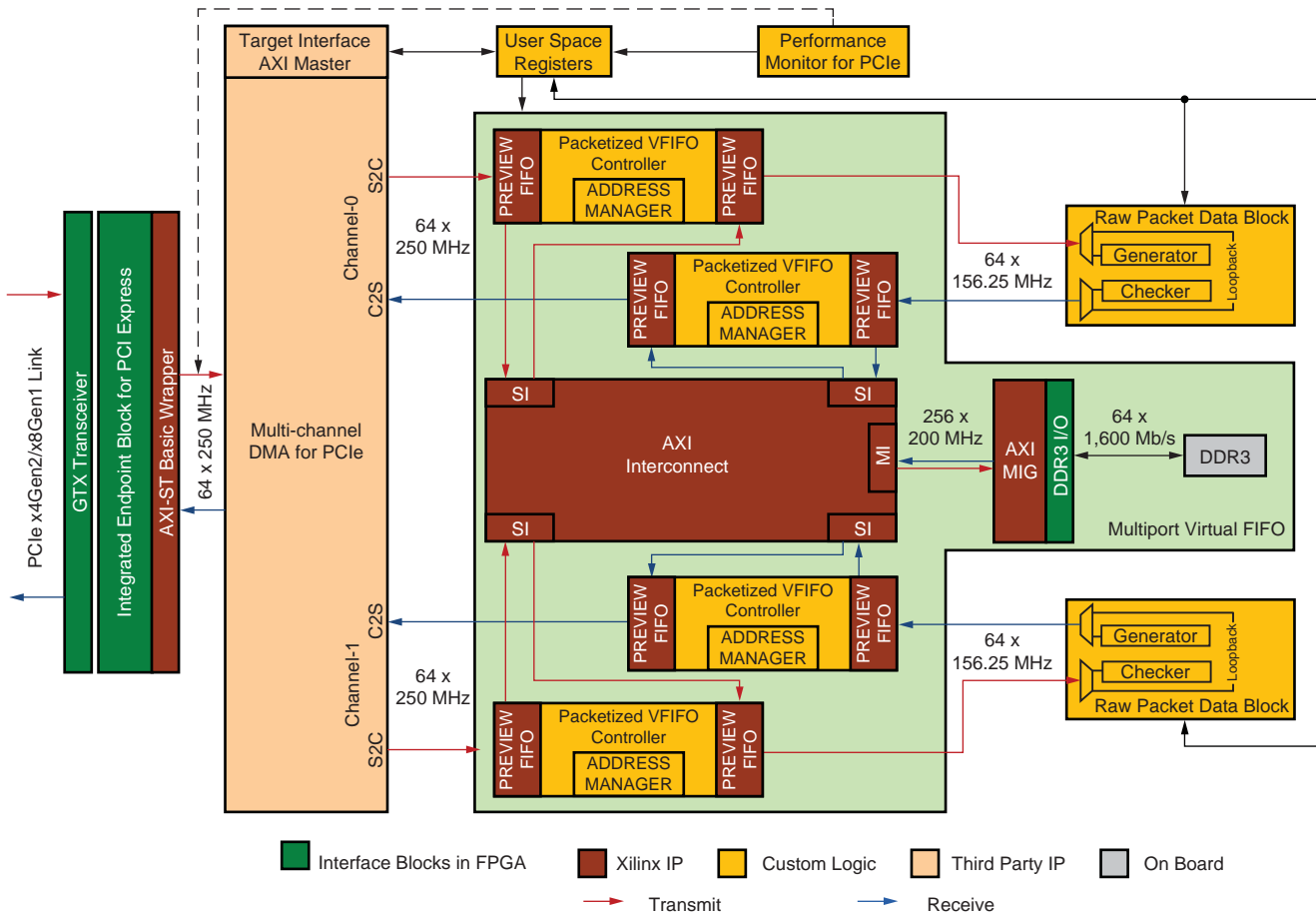
Figure 12:   **Kintex-7 FPGA Based Targeted Reference Design Block Diagram**

Summary

The video application example and the Kintex-7 FPGA Targeted Reference Design demonstrate the type of high-performance, AXI4-based systems that can be realized using Xilinx's ISE and Vivado Design Suite tools, along with Plug-and-Play IP.

By creating systems that pay attention to the general design guidelines described in this white paper, users can leverage the flexibility and scalability of AXI4 Interconnects to implement a wide range of systems. Faster time to market is a competitive advantage; choosing Xilinx FPGAs, Targeted Design Platforms, AXI4 IP, and industry-standard AXI4 support provides that competitive edge.

Related Information

Xilinx Website Links

For more information, go to:

1. Xilinx products and services: xilinx.com
2. AMBA Open Specifications:
   http://www.arm.com/products/system-ip/amba/amba-open-specifications.php
3. Xilinx AXI4 Support: http://www.xilinx.com/ipcenter/axi4.htm
4. Related Xilinx and third-party IP:
   http://www.xilinx.com/products/intellectual-property/index.htm
5. Xilinx Targeted Design Platforms:
   http://www.xilinx.com/products/targeted_design_platforms.htm

Application Notes

6. XAPP521, *Bridging Xilinx Streaming Video Interface with the AXI4-Stream Protocol*
7. XAPP739, *AXI Multi-Ported Memory Controller*
8. XAPP741, *Designing High-Performance Video Systems in 7 Series FPGAs with the AXI Interconnect*

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
03/22/12	1.0	Initial Xilinx release.
07/22/13	1.1	Added Zynq device and Vivado Design Suite information.

Notice of Disclaimer