Design of AXI Coherency Extensions (ACE) to APB Bridge: Review

Ashish jhariya, Deepika soni
[1]M. Tech. Scholar, [1], Asst. Prof. [2]
GGITS, Jabalpur

Abstract: Microprocessor performance has improved rapidly these years. In contrast memory latencies and bandwidths have improved little. The result is that the memory access time is the bottleneck which limits the system performance. In case of larger system design which requires more number of I/O ports and more memory capacity the system designer may interface external I/O ports and memory with the system. We are using advanced microcontroller bus architecture with its advanced high performance bus. The Advanced Microcontroller Bus Architecture (AMBA) is a widely used interconnection standard for System on Chip (SoC) design. In order to support high-speed pipelined data transfers, AMBA supports a rich set of bus signals, making the analysis of AMBA-based embedded systems a challenging proposition. The goal of this is to synthesize and simulate complex interface bridge between Advanced eXtensible Interface (AXI) and Advanced Peripheral Bus (APB) known as AXI2APB Bridge. This also involves the Back notation for Synthesized of Bridge module and to perform Functional and Timing Simulation using Xilinx ISE. In this we are using ace to APB interfacing via a bridge and using parallel operations which we are divided the large program or modules into a smaller one. Here used parallel pipeline to fast operation. In the thesis we are proposing to design a very high performance ace to APB and vice versa bridge and to achieve our proposed work we have used pipelining for data transfer.

I. INTRODUCTION

The Advanced Microcontroller Bus Architecture (AMBA) is a widely used interconnection standard for System on Chip (SoC) design [1].In order to support high-speed pipelined data transfers AMBA supports a rich set of bus signals, making the analysis of AMBA-based embedded systems a challenging proposition. The AMBA specification [2] has become a de-facto standard for the semiconductor industry, it has been adopted by more than 95% of ARM's partners and a number of IP providers. The specification has been successfully implemented in several ASIC designs. Since the AMBA interface is processor and technology independent, it enhances the reusability of peripheral and system components across a wide range of applications. The AMBA specification has been derived to satisfy the following four key requirements.

(i) To facilitate the right-first-time development of Embedded Microcontroller Products with one or more CPUs or signal processors.

(ii) To be technology-independent and ensure that highly reusable peripheral and system macro cells can be migrated across a diverse range of IC processes and be appropriate for full-custom, standard cell and gate array technologies.

(iii) To encourage modular system design to improve processor independence, providing a development road-map for advanced cached CPU cores and the development of peripheral libraries.

(iv) To minimize the silicon infrastructure required supporting efficient on-chip and off-chip communication for both operation and manufacturing test. This paper is to design the AMBA based AXI2APB Bridge which interfaces ace and APB buses. It is required to bridge the communication gap between low bandwidth peripherals on APB with the high bandwidth ARM Processors and/or other high-speed devices on AXI. This is to ensure that there is no data loss between ace to APB or APB to ace data transfers. In our work we intend to use Verilog HDL (Hardware Description Language) for designing the RTL (Register Transfer Level) code[3]. Synthesis and Simulation is done using Xilinx[4].

II. TYPICAL AMBA BASED MICROCONTROLLER

An AMBA-based microcontroller typically consists of a high-performance system backbone bus (AMBA ace or AMBA ASB), able to sustain the external memory bandwidth on which the CPU, on-chip memory and other

Direct Memory Access (DMA) devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high performance bus is a bridge to the lower bandwidth APB, where most of the peripheral devices in the system are located. AMBA APB provides the basic peripheral [5] macro cell communications infrastructure as a secondary bus from the higher bandwidth pipelined main system bus. Such peripherals typically:

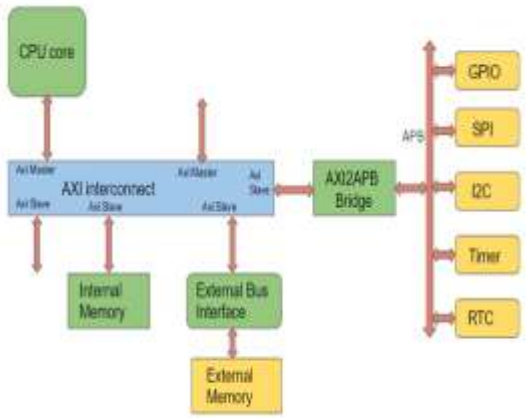

Figure-1. AMBA based Simple Microcontroller

(i) Have interfaces which are memory-mapped registers
(ii) Have no high-bandwidth interfaces
(iii) Are accessed under programmed control.

III. OVERVIEW OF AMBA BUSES

The Advanced Microcontroller Bus Architecture (AMBA) is ARM's no-cost, open specification[2], which defines an on-chip communications standard for designing high performance Embedded Microcontrollers. Three distinct buses are defined within the AMBA specification:
(i) Advanced eXtensible Interface (AXI)
(ii) Advanced high-performance bus (AHB)
(iii) The Advanced System Bus (ASB)
(iv) The Advanced Peripheral Bus (APB).

A. Advanced eXtensible Interface (AXI)

AXI is a new generation of AMBA bus, which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation. AMBA ace [6] implements the features required for high-performance, high clock frequency systems including:
• High performance
• Pipelined operation
• Multiple bus masters
• Burst transfers
• Single-cycle bus master handover
• Non-tri state implementation
• Wider data bus configurations(64/128bits).

Bridging between this higher level of bus and the current ASB/APB can be done efficiently to ensure that any existing designs can be easily integrated.
An AMBA ace design may contain one or more bus masters typically a system would contain at least the processor and test interface. However, it would also be common for a Direct Memory Access (DMA) or Digital Signal Processor (DSP) to be included as bus masters. The external memory interface, APB Bridge and any internal memory are the most common ace slaves. Any other peripheral in the system could also be included as an ace slave. However, low-bandwidth peripherals typically reside on the APB.

B. Advanced peripheral bus (APB)

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA) hierarchy [7] of buses and is optimized for minimal power consumption and reduced interface complexity.
The AMBA APB should be used to interface to any peripherals which are low-bandwidth and do not require the high performance of a pipelined bus interface. The latest revision of the APB ensures that all signal transitions are only related to the rising edge of the clock. This improvement means the APB peripherals can be integrated easily into any design flow.
Features of APB:
• Low power
• Latched address and control
• Simple interface
• Suitable for many peripherals

These changes to the APB also make it simpler to interface it to the new Advanced High-performance Bus (AHB).

IV. OPERATION OF AXI2APB BRIDGE

The AXI2APB interfaces ace and APB. It buffers address, controls and data from the AXI, drives the APB peripherals and return data along with response signal to the AHB. The AXI2APB interface is designed to operate when ace and APB clocks have the any combination of frequency and phase [8]. The AHB2APB performs transfer of data from ace to APB for write cycle and APB to ace for Read cycle [2].

A. Features of AXI2APB Bridge

Interface between AMBA Advanced eXtensible Interface (AXI) and AMBA peripheral bus (APB) [2], provides latching of address, controls and data signals for APB peripherals. Supports for the following
• APB compliant slaves and peripherals.
• Peripherals which require additional wait states.

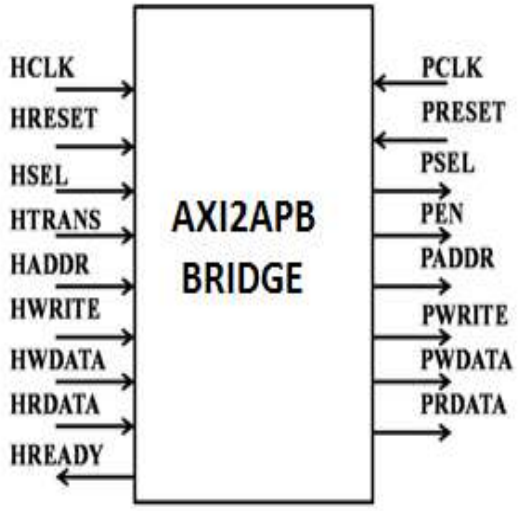

Figure 2. Pin details of AXI2APB Bridge

B. ACE Response: The sub-module ace Response sequences the way that the AXI2APB responds to ace requests. Valid commands are forwarded to control transfer for action. Invalid commands are not forwarded and an error message is generated. It operates on ace CLOCK and RESET. The control Transfers block in Fig. 3 transfers ace control signal to the APB access with appropriate delays inserted to map the pipelined ace protocol to the two cycle APB protocol. It ensures that only one request is presented to the APB access while it is processing a request. It operates on ace CLOCK and RESET.

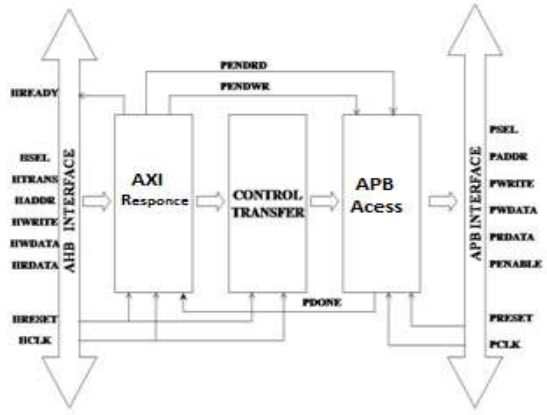

Figure 3. Internal architecture of the bridge

C. APB Access

The APB access generates the control signals on the APB for read and writes cycles. It operates on APB CLOCK and RESET. The APB Bridge is the only bus master on the AMBA APB. In addition, the APB Bridge is also a slave on the higher-level system bus. The bridge unit converts system bus transfers into APB transfers and performs the following functions:

• Latches the address and holds it valid throughout the transfer.
• Decodes the address and generates a peripheral select (PSEL). Only one select signal can be active during a transfer.
• Drives the data onto the APB for a write transfer.
• Drives the APB data onto the system bus for a read transfer.
• Generates a timing strobe, PENABLE, for the transfer.

V. DESIGN OF AXI2APB BRIDGE

AHB2APB Bridge operates on HCLK and APB access sub module operates on PCLK. ace response and Control transfer is together termed as ace interface and APB access is termed as APB interface to ensure the correct generation of suitable control signals and address we use three internal signals in the bridge module namely:
PENDWR (Pending Write)
PENDRD (Pending Read)
PDONE (Peripheral operation done)

The capture of address & control for Write or Read operation is done when HREADY, HTRANS and HSEL are valid. READY is the only signal that is the output from the bridge to ace master to cope up the communication between ace and APB. Hence the generation of HREADY signal is very significant in the bridge module. By using the internal signals PENDWR and PENDRD and double synchronized signal (Double synchronization is explained later in this section) PDONE, HREADY generation is made easy to capture the next control for Write or Read operation from ace to APB. Since the sub modules operate on different clock domains namely HCLK and PCLK, there is a need for interfacing these clock domains. Any two systems are considered asynchronous to each other:
• When they operate at two different frequency
• When they operate at same frequency, but at two different clock phase angles

This interfacing is difficult in the sense that design becomes asynchronous at the boundary of interface, which results in setup and hold time violation, Meta stability and unreliable data transfers. Hence we need to go out for special design and interfacing techniques. In such a case if we need to do data transfer, there are very few methods to achieve this namely:
• Handshake signalling method
• Asynchronous FIFO

Both have its own advantages and disadvantages. In our paper we have used Handshake signalling Method. In Handshake signalling method the ace interface sends data to APB interface based on the handshake signals PENDWR (or PENDRD) and

PDONE signals. The protocol for this uses the same method that is found with 8155 chip used with 8085 based on handshake signals Request and Acknowledge.

VI. PROTOCOL

AHB interface asserts the PENDWR (or PENDRD) signal, makes the APB interface to accept or to send the data on the data bus.APB interface asserts the PDONE signal, asserting that it has accepted or sent the data. This method is straightforward but it has got loop holes. when APB interface samples the ace interface‘s PENDWR (or PENDRD) line and ace interface samples APB interface’s PDONE line, they are done with respect to their internal clock, so there will be setup and hold time violation. To avoid this we use double stage synchronizers, which are immune to meta stability to a good extent. The double synchronizers for PENDRD and PDONE will be same as double synchronizers for PENDWR with the only difference that synchronizer for PDONE is made to operate on HCLK unlike PENDWR and PENRD which operate on PCLK. If we do the double synchronizing, then the transfer rate comes down, due to the fact that a lot of clock cycles are wasted just handshaking.

VII. LITERATURE WORK

Chenghai et al [1] ARM introduced the Advanced Microcontroller Bus Architecture (AMBA) 4.0 specifications in March 2010, which includes advanced eXtensiable Interface (AXI) 4.0. AMBA bus protocol has become the de facto standard SoC bus. That means more and more existing IPs must be able to communicate with AMBA 4.0 bus. Based on AMBA 4.0 bus, we designed an Intellectual Property (IP) core of Advanced Peripheral Bus (APB) Bridge, which translates the AXI4.0-lite transactions into APB 4.0 transactions. The bridge provides an interface between the high-performance AXI bus and low-power APB domain**.**

Varsha vishwarkamaet al [2] Microprocessor performance has improved rapidly these years. In contrast memory latencies and bandwidths have improved little. The result is that the memory access time is the bottleneck which limits the system performance. In case of larger system design which requires more number of I/O ports and more memory capacity the system designer may interface external I/O ports and memory with the system. In this paper we are using advanced microcontroller bus architecture with its advanced high performance bus. AMBAAXI provides parallel communications with multi master bus management, high clock frequency, high performance systems for data transfer operation from the memory interfaced with the master or slave peripheral devices. AMBA AXI supports on chip communications standard for designing high-performance embedded microcontrollers.

VIII. CONCLUSION

The RTL Simulation of AXI2APB Bridge has been verified and validated by using suitable test benches namely ace Driver/Monitor and APB Driver/Monitor. The AXI2APB Bridge has been successfully synthesized by the extraction of Synthesized Netlist with unit delays and verified by comparing the Gate level Simulation with RTL Simulation results. The Back Annotation of AXI2APB Bridge has also been successfully completed by the extraction of Synthesized Netlist with suitable delays & verified by the comparison of Gate level simulation with RTL simulation results. Thus AXI2APB Bridge is a standalone solution to extract the advantages of newly developed ARM based AMBA ace bus by bridging the common gap between ace and the existing APB bus.

REFERENCES

[1] ARM, "AMBA Protocol Specification 4.0", www.arm.com, 2010.
[2] Ying-Ze Liao, "System Design and Implementation of AXI Bus", National Chiao Tung University, October 2007.
[3] Clifford E. Cummings, "Coding And Scripting Techniques for FSM Designs with Synthesis-Optimized, Glitch-Free Outputs," SNUG (Synopsys Users Group Boston, MA 2000) Proceedings, September 2000.
[4] Clifford E. Cummings, “Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs,” SNUG 2001
[5] Chris Spear, "SystemVerilog for Verification, 2nd Edition"Springer,www.springeronline.com, 2008.
[6] Lahir, K., Raghunathan A., Lakshminarayana G., “LOTTERYBUS: anew high-performance communication architecture for system-on-chip designs,” in Proceedings of Design Automation Conference, 2001.
[7] Sanghun Lee, Chanho Lee, Hyuk-Jae Lee, “A new multi-channel on chip-bus architecture for system-on-chips,” in Proceedings of IEEE international SOC Conference, September 2004.
[8] Martino Ruggiero, Rederico Angiolini, Francesco Poletti, Davide Bertozzi, Luca 86
[9] Benini, Roberto Zafalon, “Scalability Analysis of Evolving SoCInterconnect Protocols,” Int. Symposium on System-on-Chip, 2004. Lukai Cai, Daniel Gajski, “Transaction level modeling: an overview,” in Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, October 2003.
[10] Min-Chi Tsai, “Smart Memory Controller Design for Video Applications,” Master thesis: National Chiao Tung University, July 2006.

