International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Comparing the Two Different Generations of AMBA Based Protocols: AHB vs AXI

Medha Chhillar[1], Geeta Yadav[2], Neeraj Kr. Shukla[3]

[1-3]VLSI Design Group, Department of EECE, ITM University, Gurgaon (Haryana) India

Abstract: ***ARM Ltd formed in 1990 as an Intellectual Property company that designs microprocessor technology. These microprocessors form the heart of digital products that vary from mobile phones and digital cameras to automotive systems. AMBA (Advanced Microcontroller Bus Architecture) was introduced by ARM in 1996 as registered trademark and is an open-standard. Here, we talk about different types of AMBA buses and how they have evolved, thereby comparing two of its main protocols viz. AHB and AXI. Comparisons using various distinguishing parameters have been made and wherever required, explained using proper diagrams and tables. A few of the parameters include level of complexities between the AHB and AXI, flexibility, number of channels present, pipelining, etc.***

Keywords: AHB (Advanced High-performance Bus), AXI(Advanced eXtensible Interface), VIP(Verification Intellectual Property), pipelining, UVM(Universal Verification Methodology), SoC(System-on-Chip)

1. Introduction

ARM Ltd formed in 1990 as an Intellectual Property company that designs microprocessor technology. These microprocessors form the heart of digital products that vary from mobile phones and digital cameras to automotive systems and are widely used today. ARM provide standards in microprocessor architectures that are compatible with Windows and Linux and hence is widely accepted with a large network supporting design and development cycle[1]. ARM gave AMBA in the year 1996 which is an open-standard solution for connecting various functional blocks on an SoC. The first generation AMBA buses included ASB(Advanced System Bus) and APB(Advanced Peripheral Bus) followed by the second generation which introduced AHB. AHB is a high-performance bus. In 2003, ARM came up with another high-performance bus named AXI which is an interface. Also, for these standard protocols, we require verification IP's to get our design verified. VIPs ensure that the design under inspection is working as expected. There are various verification methodologies such as VMM, UVM, etc. UVM being the latest one. So, here we are comparing the two AMBA protocols viz. AHB and AXI.

2. Evolution of AMBA Protocols

AMBA protocols are broadly categorized into three generations. The first generation AMBA protocols are of two types viz. ASB and APB. The former is for high-performance system modules, and the latter is for low-power peripherals. Followed by the first generation, came the second generation which came up with an even more advanced bus protocol for high-performance. This protocol was named as AHB which stands for the Advanced High-performance Bus protocol. AHB supports system modules with high-clock frequency and high-performance. This bus acts as the backbone for high-performance systems and is also efficient in connecting various internal as well as external peripherals [2].
Followed by AHB, we have a third generation protocol which is the AMBA AXI namely the Advanced eXtensible Interface. It again has an advancement to AHB and supports high frequency as well as high-performance system designs for high-speed interconnect[3]. These generations are illustrated in brief as below.

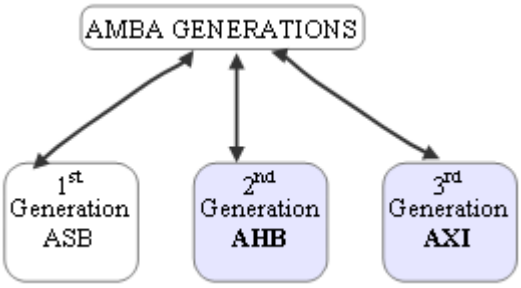

Figure 1: AMBA Generations in brief

3. Comparing the two Protocols: AHB vs AXI

3.1 Level of Complexities

These two protocols are complex in their own ways. The former is for supporting high-performance systems and the latter is an advanced interface. AHB is said to be complex in a way that it has the concept of multi-stage pipelining where the number of stages may vary from three to even eight.
On the other hand, when it comes to AXI pipelining is present, but with the use of register slices. These register slices are mainly used to add a required amount of delay which is essential for proper functioning of the VIP.

3.2 Key Features

AHB is a new generation bus which came after the APB to address the requirements of a high-performance system with high-clock frequency systems. The other features that AHB includes is:

- burst transfers
- split transactions
- single cycle bus master handover
- single clock edge operation

- non-tristate implementation
- wider data bus configurations(64/128 bits)[2]

AXI on the contrary is suitable for high-bandwidth and low-latency designs. Also, without using complicated bridges, AXI can enable high-frequency operations. The key features that distinguishes AXI from AHB are:

- separate phases for address/control and data transfer
- data transfers with no proper alignment using byte strobes is supported
- only the start address is required in case of burst transactions
- separate channels for both read and write data to enable low-cost Direct Memory Access (DMA)
- ability to issue multiple outstanding addresses
- out-of-order transaction completion
- easy addition of register stages to provide timing closure[3]

3.3 Throughput and Latency

AHB protocols supports a single channel in order to carryout all the data transactions to the connected peripherals. Hence, managing multiple masters requesting for a particular slave can be done using an arbitration algorithm. The arbiter manages these requests and the decoder decodes the responses received from slaves and sends them to the required masters. Here, when a particular transaction is going on, no other transaction can start unless the previous one is complete. This could be considered as a drawback which was handled with the introduction of another concept of split-transactions. In a split-transaction, priority of a request is considered. If one transaction is going on and simultaneously another master has requested for the bus which has a higher priority, then the current transaction is said to stop and split, thereby letting the other one with a higher priority to finish. After the completion, the execution goes back to the previous transaction.

AXI on the other hand, is a multi-channel bus with separate channels for read address, write address, read data, write data and for response. The address and response channels are to improve pipeline of multiple requests[4]. Here, we do not have the concept of split-transaction as separate channels are present for separate operations. But, here we do have the concept of out-of-order execution. In OO execution, we can have multiple requests being executed with no particular order. The reason being, with each transaction, the address information is also sent. If we have the address of each byte of data in a transfer, then it becomes easy to change the order. But this is not the case with AHB, where only the start address of burst is sent.

3.4 Timing Diagrams

The AHB data transfer can be illustrated in figure 2 using a simple burst transaction showing different addresses at every cycle inside the burst. Also, a clear difference can be seen from figure 3 where another burst transfer with only start address can be seen. From these two figures we can clearly differentiate between the two. Also, to summarize the differences a table (Table 1) has been created for a better understanding.

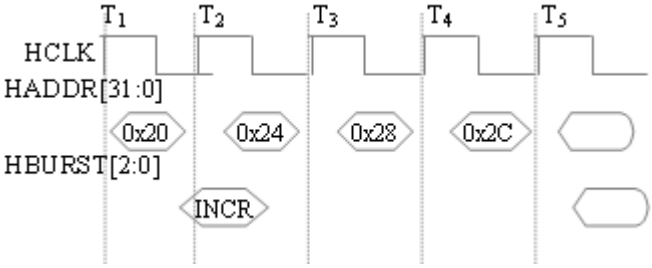

Figure 2: AHB Burst Transaction with Addresses at each clock-cycle

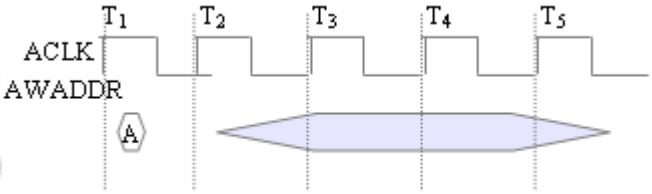

Figure 3: AXI Burst Transaction with only start Address

Table 1: AHB vs AXI

Parameters	*AMBA 2.0 AHB*	*AMBA 3 AXI*
Channels	Single-channel	5 separate channels
Address in Burst Mode	Requires address of every data item to be transmitted	Address of only first data item is sent
Transaction Scheme	OO(Out-of-Order)	SPLIT and RETRY
Burst mode	Fixed for memory mapped I/O peripherals	No fixed burst mode
Exclusive data access support	Yes	No
Low-power clock control interface	Yes	No

3.5 Summary

AHB is a high-performance bus whereas AXI is an extensible interface. The former is a single-channel bus protocol while the latter is a multi-channel bus. All the bus masters share a single bus in case of AHB while in case of AXI separate channels are provided for read address, read data, write address, write data, and write response. Bus latencies starts lower in AHB than the AXI. AXI consumes approximately 50% more power as compared to AHB[5].

4. Conclusion

ARM has given great products to the world, and the AMBA protocols is one of them. In this paper, two generations of AMBA protocols have been explained, and hence compared. There are other protocols also which have not been covered as AHB and AXI are the most commonly used bus protocols. The paper started with the evolution of the AMBA protocols and headed with a brief description to a few protocols mentioned in the paper, and finally ended with a clear comparison of the two protocols (AHB vs AXI) with a table. Main parameters for comparison included throughput and latency, level of complexities, number of channels, etc. Further studies are being made and we are

trying to create verification IP's using UVM methodology for AHB and AXI and soon we would be able to come up with even more clearer comparisons.

References

[1] Ioannis Skazikis, "The ARM Processor Architecture," Matr-Nr. 637868. (Technical Report Style)
[2] "AMBA™ Specification (Rev 2.0)," in May 13th 1999, Issue A, © Copyright ARM Limited 1999. (Book Style)
[3] "AMBA AXI Protocol® v1.0 Specification" in June 16th 2003, Issue A and March 19th 2004, Issue B, Copyright © 2003, 2004 ARM Limited. (Book Style)
[4] Deepak Shankar. "Comparing AMBA AHB to AXI Bus using System Modeling," [Online]. Available:http://www.designreuse.com/articles/24123/amba-ahb-to-axi-bus-comparison.html. [Accessed: Mar. 5, 2014]. (Technical Report Style)
[5] "Difference Between AHB and AXI," [Online]. Available: http://www.differencebetween.net/technology/difference-between-ahb-and-axi/ [Accessed: Apr. 10, 2014]. (General Internet Site)

Author Profile

Medha Chhillar received the B.Tech degree in Electronics and Communication Engineering from Kurukshetra University in 2012 and is currently pursuing M.Tech in VLSI Design from ITM University. Her interests have been in verification, and free and open source eda. She is now doing her internship with CoVerify Systems Technology LLP.

Geeta Yadav received the B.Tech degree in Electronics and Communication Engineering from Maharshi Dayanand University in 2012 and is currently pursuing M.Tech in VLSI Design from ITM University. Her interests have been in free and open source eda, and verification. She is now doing her internship with CoVerify Systems Technology LLP.