Third Generation of AMBA (AXI) implementation using Verilog

[1]Anjali P, [2] Asha Rani A.R

[1]*Asst. Professor, Dept. of ECE, Dayananda Sagar Academy of Technology and Mgmt., Bengaluru*
[2]*Asst. Professor, Dept. of ECE, Dayananda Sagar Academy of Technology and Mgmt., Bengaluru*

***Abstract*—**A System on Chip (SoC) is an integrated circuit that integrates all components of a computer or other electronic systems. It integrates advanced peripherals like GPU, Wi-Fi module with microprocessor or microcontroller. AMBA is one among SoC designs which has higher scope in present electronic world. In 1996, ARM introduced Advanced Microcontroller Bus Architecture (AMBA). As of now ARM has made four improvisations to the AMBA. Namely Advanced High performance Bus (AHB) in 1999, Advanced Extensible Interface (AXI) in 2003, AXI4 also known as ACE in 2010-11, AMBA Coherent Hub Interface (CHI) in 2013. This paper describes the third generation of AMBA i.e. Advanced Extensible Interface (AXI) and also the advantages of AXI over previous versions of AMBA.

***Keywords*—** SoC, AMBA, AHB, AXI**.**

I. INTRODUCTION

Embedded System is microcontroller based, software-driven, reliable, real time control system, designed to perform a specific task. It can be thought of as a computer hardware system having software embedded in it. Advanced Microcontroller Bus Architecture describes a generic set of buses for use in SoCs, application-specific integrated circuits (ASICs), and traditional microcontroller.

AMBA is the most widely used as the on-chip bus in the SoC designs. In the first version of AMBA, there are two parts, namely Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). In APB no pipelining is present. It is sometimes optimized for reduced interface complexity and minimal power consumption for supporting peripheral functions. This bus can also be used in union with either ASB or AHB. Advanced High-performance Bus (AHB) is the advanced version of ASB which uses full duplex parallel communication whereas APB uses massive memory-I/O accesses.

Advanced Extensible Interface (AXI) is the third generation of AMBA whose features are:

- High bandwidth in low-latency designs are supported.
- It enables high frequency operation without using any complex bridges.
- Interface requirements of a wide range of components are met.
- It is best fitted for memory controllers with high initial access latency.
- In AXI one can feel flexibility in designing and implementation of interconnect architectures.
- Backward compatible with existing AHB and APB interfaces.

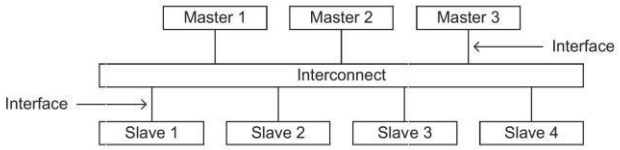
This paper has been organized in three different sections. The first section describes the architecture, channel and signal descriptions of AXI. The second section describes master write operation whereas, the third section describes the advantages of AXI over the previous versions of AMBA

II. ARCHITECTURE OF AXI

An AMBA AXI system consists of a number of master and slave devices connected together through interconnect. In practical AMBA AXI consists of overall 16 masters and 16 slaves connected through interconnect. AXI protocol provides a single interface definition for the interfaces.

- Between a master and the interconnect
- Between a slave and the interconnect
- Between a master and the slave.

The typical block diagram of AMBA AXI is shown below.



An AXI protocol is bus based and defines totally five different channels, namely

- Write Address Channel
- Write Data Channel
- Write Response Channel
- Read Address channel
- Read Data channel

Handshake Mechanism in AXI

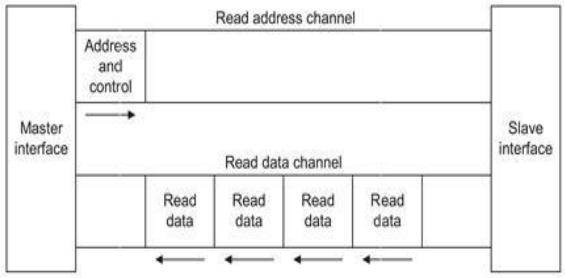
Each of five independent channels consists of a set of information signals and uses a two way *VALID* and *READY* handshake mechanism. The information source uses the valid signal to show a valid data or control information is available on the channel. The ready signal is used by the destination to show when it can accept the data. The read data channel and the write data channel include one more handshake signal *LAST* to indicate when the transfer of final data item within a transaction takes place.

Read and Write Address Channels

Both Read and Write transactions have their own address channels. This address channel carries all of the required address and control information for a transaction

Read Data Channel

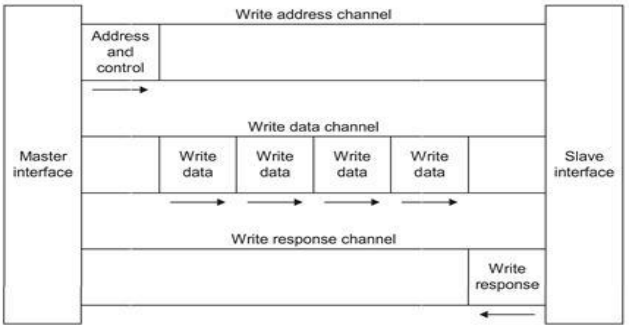
The Read Data and any Read response information is conveyed by the Read Data channel from the Slave Back to Master.



Channel architecture of reads

Write Data Channel

The Write Data from the Master to the Slave is conveyed by the Write Data Channel. The data bus can be 8, 16, 32, 64,128, 256, 512 or 1024 bits wide. One byte lane strobe for every 8 data bits indicating which byte of data bus are valid.



Channel architecture of writes

Write Response Channel

The way for Slave to respond to the Write transaction is provided by Write response channel. The completion signal occurs once for each burst, not for each individual data transfer within the bus.

III. MASTER WRITE OPERATION

The Master sends the address to the slave where the Data is to be written through the AWADDR. If the Slave is ready to accept the data then the master sets the different write ID for different data through the signal AWID. Then it sends the data with the signal AWDATA. AWID helps in performing out of order transaction. The Interface diagram for Master is as shown below

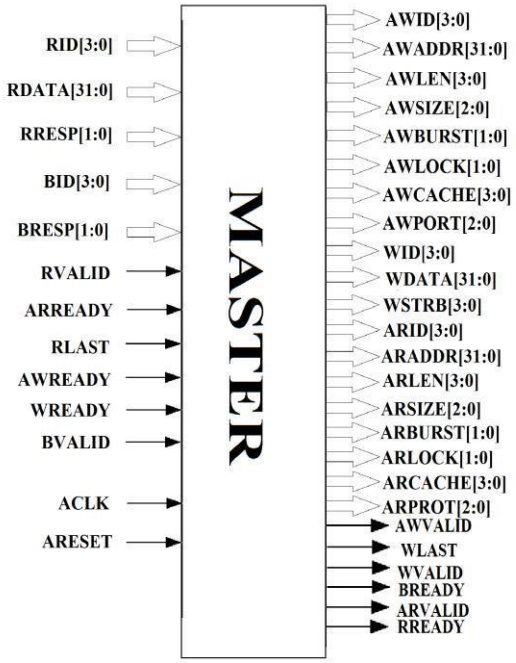


Figure: Depicts the Interfacing diagram for Master

The Finite State Machine (FSM) for the Write operation performed by the Master is as follows,

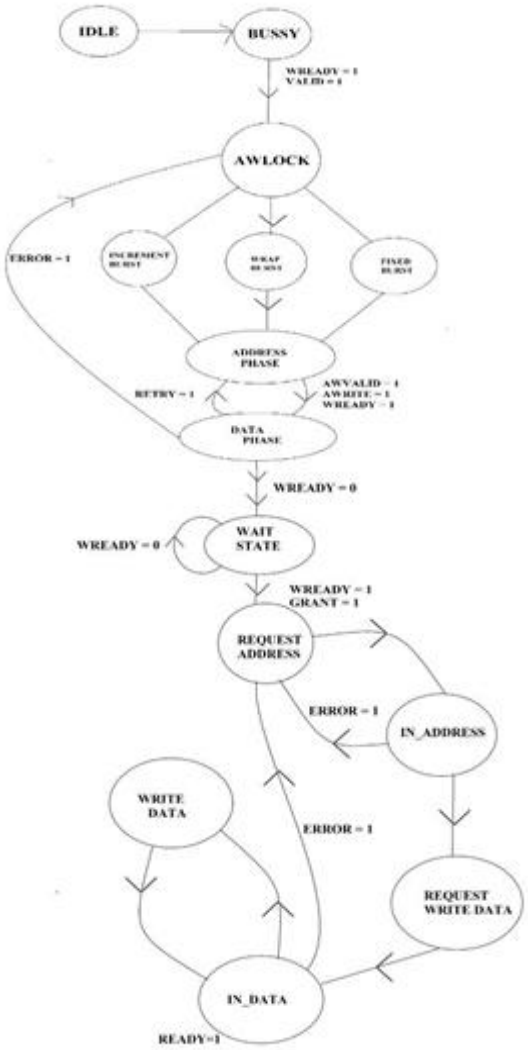


Fig: FSM for master read operation

VI. ADVANTAGES OF AXI OVER PREVIOUS VERSIONS

Advantages of the AMBA 3 AXI protocol include:

- Independently acknowledged address and data channels
- Out-of-order completion of bursts
- Exclusive access (atomic transaction)
- System level cache support
- Access security support
- Unaligned address & byte strobe
- Static burst, which allows bursts to FIFO memory
- Low power mode

The AMBA 3 AXI architecture differs significantly from previous AMBA protocols with the introduction of channels.

The Master and Slave in AXI communicate in a transaction-oriented communication. Each transaction consists of address, data, and response transfers on their corresponding channels.Every transfer identifies itself as part of a specific transaction by its transaction ID tag. Transactions may complete out-of-order and transfers belonging to different transactions may be interleaved. The out of order transactions can be sorted out at the destination.

V. CONCLUSION

After AXI, the latest versions of AMBA, ACE and ATB have been introduced while the older versions of AMBA are being used in some less demanding architecture. Further we are trying to implement the AXI bus in our future studies about AMBA AXI.

REFERENCES

[1] AMBA specification, Rev 2.0 by ARM, May 1999.
[2] AMBA AXI protocol specification, version 1.0 by ARM, March 2004.
[3] AMBA AXI protocol specification ,version 2.0 by ARM, March 2010.
[4] Priyanka Gandhani and Charu Patel, “Moving from AMBA AHB to AXI Bus in SoC Designs: A Comparative Study”, August 2011.