IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 3, Issue 10, May 2015)

Review of System – On- Chip Advanced eXtensible Interface (AXI) Bus Protocol

Dr. K.C. Mahajan[1], Mukesh K Yadav[2]

Professor, Technocrats Institute of Technology (Excellence), Bhopal Madhya Pradesh[1]

PhD Scholar (Electronics) Mewar University Chitodgarh, Rajasthan[2]

Abstract: **This paper gives a brief description of various on-chip bus protocols such as the Advanced Microcontroller Bus Architecture (AMBA) Advanced High-Performance bus (AHB) and Advanced Extensible Interface (AXI), Wishbone Bus, Open Core Protocol (OCP) and CoreConnect Bus. It gives a brief introduction of high performance system-on-chip bus protocol termed as an Extensible Interface (AXI). The AMBA AXI protocol supports high-performance, high-frequency system designs. It includes the optional extensions that cover signaling for low-power operation.**

Keywords: **Advanced Extensible Interface (AXI), Advanced Microcontroller Bus Architecture (AMBA), System-on-chip; Low power.**

I. INTRODUCTION

The Advanced Extensible Interface (AXI) is a part of the Advanced Microcontroller Bus Architecture (AMBA) which is developed by ARM (Advanced RISC Machines) company [1]. It is an On-Chip communication protocol. The AMBA AXI protocol supports high-performance, high-frequency system designs.

The AXI protocol:

It is suitable for high-bandwidth and low-latency designs, provides high-frequency operation without using complex bridges, meets the interface requirements of a wide range of components and suitable for memory controllers with high initial access latency , provides flexibility in the implementation of interconnect architectures is backward-compatible with existing AHB and APB interfaces.

II. LITERATURE REVIEW

The Advanced Microcontroller Bus Architecture (AMBA) is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs. It facilitates development of multi-processor designs with large numbers of controllers and peripherals. Since its inception, the scope of AMBA has, despite its name, gone far beyond micro controller devices. Today, AMBA is widely used on a range of ASIC and SoC parts including applications processors used in modern portable mobile devices like Smartphone's.

AMBA was introduced by ARM in 1996. The first AMBA buses were Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). In its second version, AMBA 2, ARM added AMBA High-performance Bus (AHB) that is a single clock-edge protocol. In 2003, ARM introduced the third generation, AMBA 3 [5], including AXI to reach even higher performance interconnect.

Advanced microcontroller bus architecture (AMBA) protocol family provides metric-driven verification of protocol compliance, enabling comprehensive testing of interface intellectual property (IP) blocks and system-on-chip (SoC) designs [1].

III. DIFFERENT SYSTEM-ON-CHIP BUS PROTOCOLS

The following are the bus protocols that are commonly used in industry. They are listed in as follows and some are explained in brief as under.

- AMBA
- AMBA AXI
- WISHBONE BUS
- OPEN CORE PROTOCOL
- CORECONNECT BUS
- MSBUS PROTOCOL

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 3, Issue 10, May 2015)

AMBA

The Advanced Microcontroller Bus Architecture (AMBA) specification by ARM defines an on-chip communications standard which is required for designing high-performance embedded microcontrollers. AMBA comprises of three buses which are the Advanced High-performance Bus (AHB), the Advanced System Bus (ASB) and the Advanced Peripheral Bus (APB) [3]. Fig. 1 shows a typical AMBA based architecture. It consists of a backbone bus which has the capability to sustain the external memory bandwidth where the on-chip memory, CPU and other Direct Memory Access (DMA) devices reside.

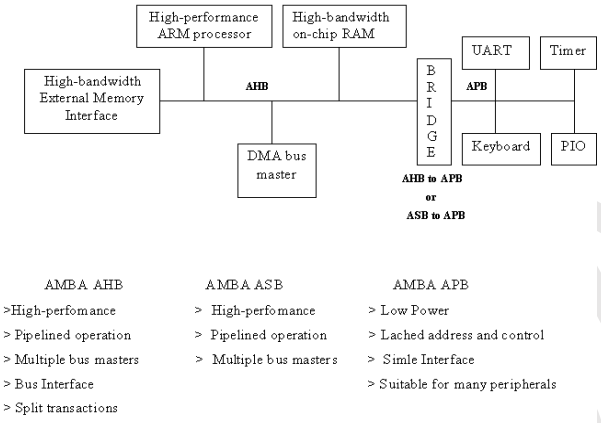

Fig 1: AMBA Architecture

The AHB is mainly used for high-performance, high clock frequency system modules. It acts as the high-performance system backbone bus. It supports multiple bus masters, provides high-bandwidth operation and supports efficient connection of processors, off-chip external memory interfaces and on-chip memories with low-power peripheral macrocell functions. It implements features such as burst transfers, split transactions, single-cycle bus master handover, single-clock edge operation, non-tristate implementation and wider data bus configurations (64/128 bits). AHB Master, AHB Slave, AHB Arbiter, and AHB Decoder are the main components of the AHB

AMBA AXI

The AMBA AXI protocol from ARM is intended at high-performance, high-frequency system designs and includes a number of features which are suitable for a high-speed submicron interconnects. The AXI protocol provides key features such as separate address/control and data phases, support for unaligned data transfers using byte strobes, burst-architecture and design of the systems in which based transactions with start address only issued, separate read and write data channels to ensure low-cost Direct Memory Access (DMA), ability to issue multiple outstanding addresses, out-of-order transaction completion and easy addition of register stages to provide timing closure [2]

AXI Protocol Specifications

A typical system consists of a number of master and slave devices connected together through the Interconnect. The 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 a slave.

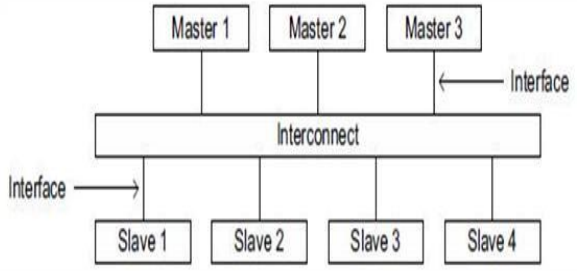

Fig 2: AXI System Topology

The AXI protocol is burst-based and defines the following independent transaction channels: Read Address Channel, Read Data Channel, Write Address Channel, Write Data Channel and the Write Response Channel.

An address channel carries control information that describes the nature of the data to be transferred. The data is transferred between master and slave using either:

A write data channel to transfer data from the master to the slave. In a write transaction, the slave uses the write response channel to signal the completion of the transfer to the master.

A read data channel to transfer data from the slave to the master.

The AXI protocol permits address information to be issued ahead of the actual data transfer. It supports

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 3, Issue 10, May 2015)

multiple outstanding transactions. It also supports out-of-order completion of transactions.

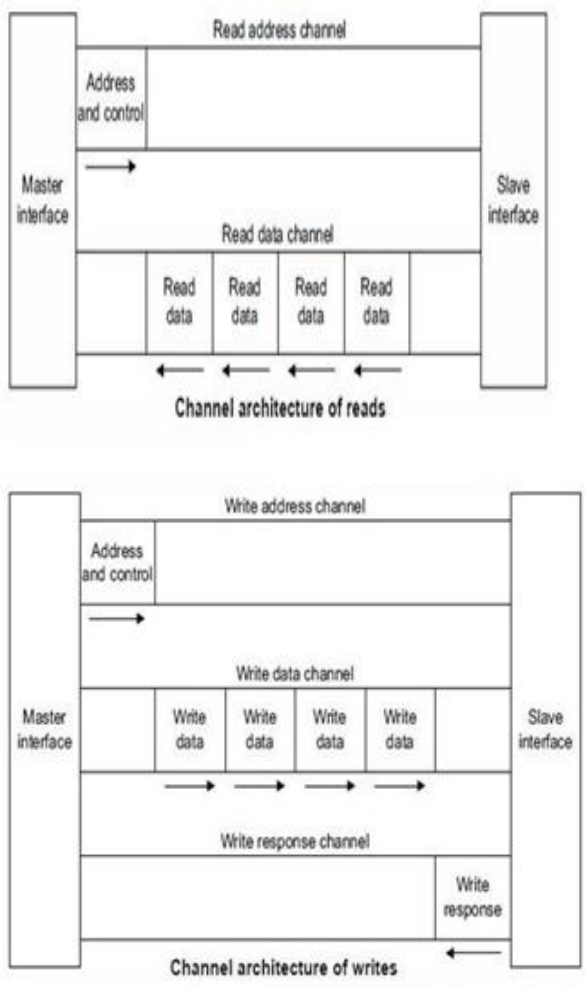

Fig 3: AXI Channel Architecture

IV. AXI CHANNELS

Each of the independent channels consists of a set of information signals (address, data or control) and **VALID** and **READY** signals that provide a two-way handshake mechanism [2].

The information source uses the **VALID** signal to show when valid address, data or control information is available on the channel. The destination uses the **READY** signal to show when it can accept the information. Both the read data channel and the write data channel also include a **LAST** signal to indicate the transfer of the final data item in a transaction.

Read and write address channels

Read and write transactions each have their own address channel. The appropriate address channel carries all of the required address and control information for a transaction.

Read data channel

The read data channel carries both the read data and the read response information from the slave to the master, and includes:

- The data bus, that can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits wide
- A read response signal indicating the completion status of the read transaction.

Write data channel

The write data channel carries the write data from the master to the slave and includes:

- The data bus, that can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits wide
- A byte lane strobe signal for every eight data bits, indicating which bytes of the data are valid.

Write data channel information is always treated as buffered, so that the master can perform write transactions without slave acknowledgement of previous write transactions.

Write response channel

A slave uses the write response channel to respond to write transactions. All write transactions require completion signaling on the write response channel.

V. CONCLUSION

In this brief, we have done the study of the AXI interface on-chip protocols along with their features, architectures and read & write channels. Using different interface techniques system speed improve and calculate other parameters.

REFERENCES

[1] AMBA Specification, Rev 2.0, ARM, 1999.
[2] AMBA® AXI™ and ACE™ Protocol Specification 2011.
[3] Wishbone Bus Architecture – A Survey and Comparison, IJVLSICS Vol. 3, No.2, April 2012.
[4] A Review of System-On-Chip Bus Protocols, IJAREEIE, Vol. 4 Issue 1, January 2015.
[5] Verification of AMBA based AXI 4 Slave Interface, Krithi B et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 3135-3137.