Research Paper

Volume : 1 | Issue : 6 | Nov 2012 • ISSN No 2277 - 8179

Engineering

KEYWORDS : AMBA, AHB, AXI

Design of AMBA 3.0 (AXI) Bus based SoC

D.Neelavathi

Guru Nanak Institute of Technology, Student, M.Tech, VLSI-SD, E.C.E Dept., Hyderabad, India

ABSTRACT

System-on-a-Chip (SoC) design has become more and more complexly. Because difference functions components or IPs (Intellectual Property) will be integrated within a chip. The challenge of integration is "how to verify on-chip communication properties". Although traditional simulation-based on-chip bus protocol checking bus signals to obey bus transaction behavior or not, however, they are still lack of a chip-level dynamic verification to assist hardware debugging. We proposed a rule based synthesizable AMBA AXI protocol checker. The AXI protocol checker contains 44 rules to check on-chip communication properties accuracy. In the verification strategy, we use the Synopsys VIP (Verification IP) to verify AXI protocol checker. In the experimental results, the chip cost of AXI protocol checker is 70.7K gate counts and critical path is 4.13 ns (about 242 MHz) under TSMC 0.18um CMOS 1P6M Technology. | The benefits of using rule-based design include improving observability, reducing debug time, improving integration through correct usage checking, and improving communication through documentation. In the final purpose, increasing design quality while reducing the time-to-market and verification costs. We anticipate that the AMBA AXI protocol checking technique will be more and more important in the future. Hence, we propose a synthesizable AMBA AXI protocol checker with an efficient verification mechanism based on rule checking methodology. There are 44 rules to check the AMBA AXI protocol that provide AXI master, slave, and default slave protocol issues.

I.INTRODUCTION

AMBA AXI [1] introduced by ARM Ltd in 1996. Widely used on - chip bus in System-on–chip (SoC) designs. AMBA protocol is an open-standard on-chip interconnect specification for the connection and management of functional blocks in a SoC [2]. AMBA enables IP–reuse: IP reuse is an essential component in reducing SoC development cost and time scale [3-7].

The Flexibility IP reuse requires a common standard while supporting a wide variety of SoCs with a different power, performance and area requirements. With its AXI, AHB, APB interface protocols, AMBA3.0 has a flexibility to match every requirement [8, 9]

The multilayer architecture acts as a cross-bar switch between masters and slaves. Compatibility to speed up design of high integration micro-controller and reuse the peripherals between systems [11].

The AMBA specification defines the Advanced System Bus (ASB) [12] in 1997 and Advanced Peripheral bus (APB) in 1999. ASB has the following features: High Bandwidth, High Performance and High Speed Bus. APB supports for Burst transfer, Separate Address and Data buses, Support for pipe - lined operation, Support for multiple bus masters, Support for multiple slave devices.

Advanced High-Performance Bus (AHB) has the following features: high speed, high bandwidth bus architecture. It is Support for split transactions, Support for burst transfers, Multiple bus masters , Pipe-lined operations, Single clock edge protocol, Large bus-widths (64/128 bit) . The AMBA 3 specification (2003) defines Advanced eXtensible Interface (AXI), Advanced High-performance Bus (AHB - Lite & AHB Multi – Layer), Advanced Trace Bus (ATB), Advanced System Bus (ASB), Advanced Peripheral Bus (APB).

In 2002, AHB Multi – Layer was introduced which is based on AHB protocol that allows for parallel access paths between multiple masters and slaves in the system, Fully compatible with the current AHB specification., Full multi layer AHB consists of a bus layer for each of the bus masters, with each layer connected to every slave through slave multiplexer. Reduces latencies and increases the bus bandwidth available to multi-master systems.

In 2002, AHB - Lite [12] was introduced which is Subset of full AHB specification and is used in designs with single bus master. AHB-Lite removes the protocol required for multiple bus masters.

Addressing Options, the master begins each burst by driving transfer control information and the address of the first byte in the transfer. It is the responsibility of the slave to calculate the address of the subsequent transfers in the burst. Bursts must not cross 4KB boundaries.

The comparison features with AMBA 3 AXI and AMBA 2 AHB [12] Protocol are:

- AHB has Fixed pipeline for the address and data transfers Bi - directional link with the complex timing relationships, Separate address for every data item, Only one transaction at a time, Fixed pipeline for address and data, Only one transaction at a time.
- The AXI has the five independent channels for R/W, addr/data and response. Each channel is unidirectional burst based - one address per burst. Multiple outstanding transactions, Out of order data ssimultaneous read and writes. AXI includes separate address/control and data phases. It has Separate read and write data channel, support for unaligned data transfers using byte strobes, Burst – based transaction with only start address issued, ability to the issue of multiple outstanding addresses, Out-of-order transaction completion, and for the Support of data interleaving, Easy addition of register stages to provide timing closure

The AXI protocol supports the following mechanisms: Variable length bursts, from 1-16 data transfers per burst, Bursts with a transfer size 8-1024 bits. Wrapping, incrementing, non -incrementing bursts. The AXI protocol provides a single interface definition for describing interfaces between a master and interconnect between a slave and interconnect and between a master and a slave.

AMBA 3 AXI protocol is a Powerful Evolution has many benefits of AMBA 2 standard by greatly extending performance and flexibility of systems based on AMBA technology. One of the key goals of AMBA 3 AXI protocol is the interoperability with existing AMBA technology. This is a well supported protocol interface with the very powerful features like : Out - of order data transactions, Data interleaving , Multiple outstanding transactions.

II.BLOCK DIAGRAM:

The block diagram of AXI Generic Master Controller in the fig 1.1 shows:

- The order in which AXI channel signals are described
- The master and slave interface conventions for AXI components.

The main components are:

- AXI Master: AXI master initiates transfer on the bus. User configurable knobs to support various features of master:

- AXI Slave: AXI slave responds to the transaction initiated by the master. Fully configured wait states and slave error response. Storing ability as internal memory and supplying data on the demand.
- AXI Interconnect/Arbiter: AXI arbiter can take multiple masters at a time and arbitrates for the bus using configurable priority scheme.

The AMBA AXI protocol is targeted at high - performance, high - frequency system designs and it includes a number of features that make it suitable for a high-speed submicron interconnect. As well as the data transfer protocol, the AXI protocol includes optional extensions that cover signaling for low-power operation.

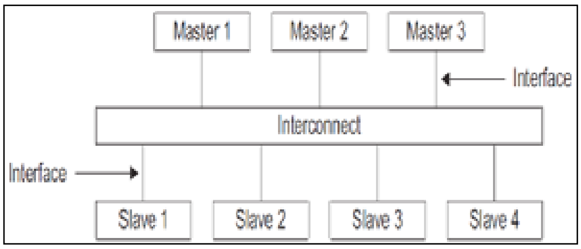

Figure1: ARCHITECTURE FOR READ AND WRITE CHANNEL

The AXI protocol is burst-based. Every transaction has address and control information on the address channel that describes the nature of the data to be transferred. The data is transferred between master and slave using a write data channel to the slave or a read data channel to the master shown in the fig 2.1. In write transactions shown in the fig 2.2, in which all the data flows from the master to the slave, the AXI protocol has an additional write response channel to allow the slave to signal to the master the completion of the write transaction.

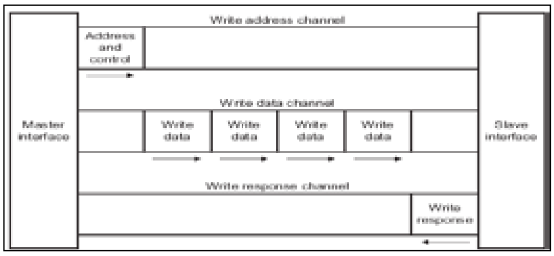

Fig 2 Channel architecture of writes

The AXI protocol enables:

- Address information to be issued ahead of the actual data transfer
- Support for multiple outstanding transactions
- Support for out-of-order completion of transactions.

HANDSHAKE PROCESS

All five channels use the same VALID/READY handshake to transfer data and control information. This two-way flow control mechanism enables both the master and slave to control the rate at which the data and control information moves. The source generates the VALID signal to indicate when the data or control information is available. The destination generates the READY signal to indicate that it accepts the data or control information. Transfer occurs only when both the VALID and READY signals are HIGH.

There must be no combinatorial paths between input and output signals on both master and slave interfaces. Fig 2.4 to Fig 2.6 shows examples of the handshake sequence. In Fig 2.4, the source presents the data or control information and drives the VALID signal HIGH. The data or control information from the source remains stable until the destination drives the READY signal HIGH, indicating that it accepts the data or control information. The arrow shows when the transfer occurs

DEPENDENCIES OF CHANNEL HANDSHAKE SIGNALS

To prevent a deadlock situation, we must observe the dependencies that exist between the handshake signals. In any transaction:

- The VALID signal of one AXI component must not be dependent on the READY signal of the other component in the transaction
- The READY signal can wait for assertion of the VALID signal.

While it is acceptable to wait for to be asserted before asserting READY, it is also acceptable to assert READY,VALID by default prior to the assertion of VALID and this can result in a more efficient design.

The single-headed arrows point to signals that can be asserted before or after the previous signal is asserted. Double-headed arrows point to signals that must be asserted only after assertion of the previous signal.

III. IMPLEMENTATION

Generally BRESP occurs during write transaction. If the data is written in slave, then it acknowledges. For example, if the signals WVALID and WREADY are high then WDATA signal goes high. Thus BVALID and BREADY occurs which further makes BRESP signal high. The timing diagram for BRESP Signal

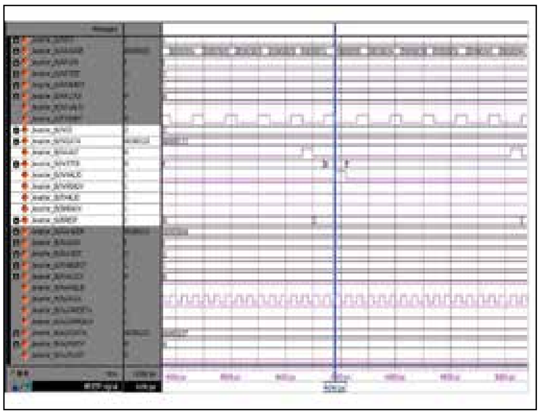

Figure 3: Waveforms

EXTENSIONS TO BASE PAPER

The concept of the AMBA AXI can be extended to multiple masters and multiple slaves as shown in the below block diagram fig 4.1. The crosswire router connects AXI masters and AXI slaves through their respective ports as described pictorially.

AMBA AXI4 (Advanced eXtensible Interface 4) is the fourth generation of the AMBA interface specification from ARM. Xilinx ISE Design Suite extends the Xilinx platform design methodology with the semiconductor industry's first AXI4 Compliant Plug-and-Play IP. For customers relying on IP to meet their Time-to-Market requirements for 7-Series, Virtex-6 and Spartan-6 based designs, the AXI4 Plug-and-Play IP offers a single standard interface to make IP integration easier. IDS 13.1 offers a broad set of AXI4 based IP with a single open standard interface across the Embedded, DSP, and Logic domains.

IV. CONCLUSIONS

- Hence, all the functionalities for AMBA-AXI protocol have been designed and verified using Advanced EDA tool.
- Transaction ordering improves system performance because it reduces the effect of transaction latency

FUTURE SCOPE

1) Future improvements in the AXI 3.0 interface block is AXI-Lite which will be a subset of it and can be designed with more features like address wrapping mode and write response signal. For ex: Burst type (b10, b11).

- In the wrapping burst the address wraps around to a lower address when a wrap boundary is reached.
- The write response signal other then OKAY, EXOKAY are SLVERR and DECERR can be implemented.

2) Instead of Verilog Test bench, if we use System Verilog verifi-

cation methodology, we can reduce the verification span of time and NRE cost.

ACKNOWLEDGMENT

I would like to articulate my profound gratitude and indebtedness to Dr. T Srinivasulu, Principal, GNIT., and Prof. B. Kedarnadh, Head of the Department, GNIT, for guiding and encouraging me in all aspects.

I wish to extend my sincere thanks to all the faculty members of Guru Nanak Institute of Technology.

REFERENCE

1]ARM, “AMBA Specification Overview”, available at http://www.arm.com/. | [2]ARM, “AMBA Specification (Rev 2.0)”, available at http://www.arm.com. | [3]ARM, “AMBA AXI Protocol Specification”, available at http://www.arm.com. | [4]Benini and D. Bertozzi, “Network-on-chip architectures and design methods,” IEEE Proc.-Comput. Digit. Tech., Vol. 152, No. 2, March 2005 | [5]Chien-Hung Chen, Jiun-Cheng Ju, and Ing-Jer Huang, “A Synthesizable AXI Protocol Checker for SoC Integration,”IEEE,,ISOCC2010. | [6]Ing. Dries Driessens and. Tom Tierens, “Overview Embedded Buses,” 2003 by Patrick Pelgrims.