Design and Verification of AXI4 Master Controller

B.BEENA¹, B.NIREESHA²

¹PG Scholar, Dept of VLSI-SD, TKR Engineering College, Hyderabad, Telangana, India, Email: beenakandikatla@gmail.com. ²Assistant Professor, Dept of ECE, TKR Engineering College, Hyderabad, Telangana, India, Email: bandarunireesha@gmail.com.

Abstract: Advanced microcontroller bus architecture (AMBA) Protocol family provides verification of protocol compliance, enabling comprehensive testing of interface intellectual property (IP) blocks and system-on-chip (SoC) designs. The AMBA advanced extensible interface 4 (AXI4) update to AMBA AXI3 includes the following: support for burst lengths up to 256 bytes, updated write response requirements, removal of locked transactions and AXI4 also includes information on the interoperability of components. AMBA AXI4 protocol system supports 16 masters and 16 slaves interfacing. This paper presents a work aimed to design the AMBA AXI4 protocol modelled in Verilog hardware description language (HDL) and simulation results for read and write operation of data and address are shown in Verilog compiler simulator (VCS) tool. The operating frequency is set to 250MHz. Two test cases are run to perform multiple read and multiple write operations.

Keywords: Intellectual property (IP), AMBA, AXI, VCS, Verilog.

I. INTRODUCTION

Integrated circuits have entered the era of System-on-Chip (SoC) [1], which refers to integrating all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio-frequency functions all on a single chip substrate. In other words, a SoC is an IC that implements most or all the functions of a complete electronic system. In general, the performance of the SoC design heavily depends upon the efficiency of its bus structure. The balance of computation and communication in any application or task is known as a fundamental determinant of delivered performance. There are many companies that develop core IPs for SoC products. With the increasing design size, Intellectual Property (IP) is an inevitable choice for SoC design and the widespread use of all kinds of IPs has changed the nature of the design flow, making On-Chip Buses (OCB) essential to the design. Of all the on-chip buses existing in the market, the AMBA (Advanced Microcontroller Bus Architecture) bus system is widely used as the de facto standard SoC bus. As the de facto Objective of project: ARM introduced the Advanced Microcontroller Bus Architecture (AMBA) 4.0 specifications in March 2010, which includes Advanced eXtensible Interface (AXI) 4.0. Based on AMBA 4.0 bus, this design is an Intellectual Property (IP) core of AXI master. In this project, normal transactions and outstanding transactions of AXI4 with INCR and WRAP burst types are designed. Standard SoC bus, AMBA bus [2] is widely used in the high-performance SoC designs. The AMBA specification defines an on-chip communication standard for designing high-performance embedded micro-controllers.

A. AMBA AXI architecture

AMBA AXI [4] supports data transfers up to 256 bytes and unaligned data transfers using byte strobes. In AMBA AXI4 system 16 masters and 16 slaves are interfaced. Each master and slave has their own 4 bit ID tags. AMBA AXI system consists of master, slave and bus. The system consists of five channels namely write address channel, write data channel, read data channel, read address channel, and write response channel. The AXI4 protocol supports the following mechanisms:

- Unaligned data transfers and up-dated write
- Response requirements.
- Variable-length bursts, from 1 to 16 data transfers per burst.
- A burst with a transfer size of 8, 16, 32, 64, 128, 256, 512 or 1024 bits wide is supported.
- Updated AWCACHE and ARCACHE Signaling details.

Each transaction is burst-based which 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. Table [3.1] gives the information of signals used in the complete design of the protocol. The write operation process starts when the master sends an address and control information on the write address channel as shown in fig 1. The master then sends each item of write data over the write data channel. The master keeps the VALID signal low until the write data is available. The master sends the last data item, the WLAST signal goes HIGH.
When the slave has accepted all the data items, it drives a write response signal BRESP[1:0] back to the master to indicate that the write transaction is complete. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. After the read address appears on the address bus, the data transfer occurs on the read data channel as shown in fig. 2. The slave keeps the VALID signal. LOW until the data is available. For the final data transfer of the burst, the slave asserts the RLAST signal to show that the last data item is being transferred. The RRESP[1:0] signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. The protocol supports 16 outstanding transactions, so each read and write transactions have ARID[3:0] and AWID[3:0] tags respectively. Once the read and write operation gets completed the module produces a RID[3:0] and BID[3:0] tags. If both the ID tags match, it indicates that the module has responded to right operation of ID tags. ID tags are needed for any operation because for each transaction concatenated input values are passed to module.

### II. RELATED WORK:

In a SoC, it houses many components and electronic modules; to interconnect these, a bus is necessary. There are many buses introduced in the due course some of them being AMBA [2] developed by ARM, CORE CONNECT [4] developed by IBM, WISHBONE [5] developed by Silicore Corporation, etc. Different buses have their own properties the designer selects the bus best suited for his application. The AMBA bus was introduced by ARM Ltd in 1996 which is a registered trademark of ARM Ltd. Later advanced system bus (ASB) and advanced peripheral bus (APB) were released in 1995, AHB in 1999, and AXI in 2003[6].AMBA bus finds application in wide area. AMBA AXI bus is used to reduce the precharge time using dynamic SDRAM access scheduler (DSAS) [7]. Here the memory controller is capable of predicting future operations thus throughput is improved. Efficient Bus Interface (EBI) [8] is designed for mobile systems to reduce the required memory to be transferred to the IP, through AMBA3 AXI. The advantages of introducing Network-on-chip (NoC) within SoC such as quality of signal, dynamic routing, and communication links was discussed in [9].To verify on-chip communication properties rule based synthesizable AMBA AXI protocol checker [10] is used.

### III. PROPOSED WORK

#### A. Block diagram of implementation of AXI

The AXI master is designed in Verilog HDL and simulated using VCS tool. It is verified using a slave model. An interface [8] is designed between the master and slave in order to ease the use of signals in master and slave. The master asserts the AWVALID or ARVALID when it has the valid address and control information and transmits the AWADDR or ARADDR respectively on to the bus. The address is accepted when the slave asserts the AWREADY or ARREADY signals. A total of 3 FIFO’s are used in the project. They are write FIFO (wfifo), write response FIFO (bfifo) and read FIFO (rfifo). Write data

<table>
<thead>
<tr>
<th>Signal</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWID</td>
<td>Master</td>
<td>Identification tag for write address group of signals</td>
</tr>
<tr>
<td>AWADDR</td>
<td>Master</td>
<td>Gives the address of first transfer in a write burst transaction.</td>
</tr>
<tr>
<td>AWLEN</td>
<td>Master</td>
<td>Gives the number of transfers in a write burst</td>
</tr>
<tr>
<td>AWSIZE</td>
<td>Master</td>
<td>Gives the size of each transfer in a burst</td>
</tr>
<tr>
<td>AWBURST</td>
<td>Master</td>
<td>Determines how address for each transfer within a burst is calculated</td>
</tr>
<tr>
<td>AWLOCK</td>
<td>Master</td>
<td>Provides additional information about the atomic characteristics of transfer</td>
</tr>
<tr>
<td>ARCACHE</td>
<td>Master</td>
<td>Indicates how transactions are required to progress through a system</td>
</tr>
<tr>
<td>AWPROM</td>
<td>Master</td>
<td>Indicates privileged and security level of a transaction and whether it is data access or an information access.</td>
</tr>
<tr>
<td>AWQOS</td>
<td>Master</td>
<td>Used as a priority indicator for the associated write or read transaction. Implemented only in AXI4.</td>
</tr>
<tr>
<td>AWREGION</td>
<td>Master</td>
<td>Permits a single physical interface on a slave to be used for multiple logical interfaces. Implemented only in AXI4.</td>
</tr>
<tr>
<td>AWUSER</td>
<td>Master</td>
<td>Optional user defined signal in the write address channel. Supported only in AXI4.</td>
</tr>
<tr>
<td>AWVALID</td>
<td>Master</td>
<td>Indicates that the channel is signaling valid write</td>
</tr>
<tr>
<td>AWREADY</td>
<td>Slave</td>
<td>Indicates that the slave is ready to accept an address and associated control signals.</td>
</tr>
</tbody>
</table>
valid signals, write address valid signals, write address and data signals, IDs, burst length, last signals and strobe signals are stored in wfifo. When the last data item is received, the access is moved into the write response FIFO. In a write response FIFO, response valid, error and response ID signals are stored. In a read FIFO(rfifo), read data and address valid signals ,read data and address signals, ID signals, read data, read last and read response signals are stored.

**B. INCR Burst**

The flowcharts for write transaction and read transaction using INCR burst burst type are shown in Fig 3 and Fig 4 respectively. If AWBURST is 1, then it represents an INCR burst type for a write transaction. The address and control signals are valid when AWVALID and AWREADY are high. The data is transferred when WVALID and WREADY signals are high. The numbers of transfers are counted using the variable x. If ARBURST is 1, and then it represents an INCR burst type for a read transaction. The address and control signals are valid when ARVALID and ARREADY are high. The data is transferred when RVALID and RREADY signals are high. The numbers of transfers are counted using the variable x. When x is equal to the burst length (AWLEN + 1), RLAST is asserted. Otherwise, the data transfer continues.

**C. Wrap burst:**

The flowcharts for write transaction and read transaction using WRAP burst type are shown Fig 5. The WRAP burst type is similar to INCR burst but the address wraps around to a lower address if an upper limit is reached.
If AWBURST is 2, then it represents a WRAP burst type for a write transaction. After each transfer, the address increments in the same way as for an INCR burst. For the first transfer, the address is the start address itself. For subsequent transfers, the address is calculated depending on AWSIZE and AWLEN. If this incremented address is the upper address limit ((wrap boundary) + (total size of data to be transferred)), then the address wraps round to the wrap boundary. Here the variables x and y are used to calculate the number of transfers before and after the upper address limit. In the above flowchart, NB is the Number of Bytes transferred ($2^{AWSIZE}$), LBL1 and UBL1 are the lower and upper byte lanes for the first transfer and LBL and UBL are the lower and upper byte lanes for the subsequent transfers.

IV. SIMULATION RESULTS

The design is implemented in Verilog HDL and verified using System Verilog. The simulation is done using the VCS tool. The simulation output signals generated are as follows from input side the validating signals AWVALID/ARVALID signals are generated by interconnect which gives the information about valid address and ID tags. For write operations BRESP[1:0] response signal generated from slave indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLERR, and DECERR. For read operations RLAST signal is raised by slave for every transaction which indicates the completion of operation. Simple transactions include the basic AXI read and write transactions. In the below waveforms Fig 6 to 11, only one transaction is performed where the start address is given as 32’h0. Also AWBURST = 1, AWLEN = 0 and AWSIZE = 2 which implies it is an INCR burst with only one transfer per burst and transfers 4bytes of data. In this case, one transaction is equal to one transfer.

For the same address and control signals, a read transaction is performed as shown in the waveform below.

![Flow chart for write WRAP burst type](image)

**Fig6. Flow chart for write WRAP burst type.**

**Fig7. Waveform for simple write transfer.**

**Fig8. Waveform for simple read transfer.**

**Fig9. Multiple write transactions with INCR burst.**
Design and Verification of AXI4 Master Controller

V. CONCLUSION

AMBA AXI4 is a plug and play IP protocol released by ARM, defines both bus specification and a technology independent methodology for designing, implementing and testing customized high-integration embedded interfaces. The data to be read or written to the slave is assumed to be given by the master and is read or written to a particular address location of slave. In this work, slave was modeled in Verilog with operating frequency of 250MHz and simulation results were shown in VCS tool. To perform single read operation it consumed 160ns and for single write operation 565ns.

Future Scope: The AMBA AXI4 has limitations with respect to the burst data and beats of information to be transferred. The burst must not cross the 4k boundary. Bursts longer than16 beats are only supported for the INCR burst type. Both WRAP and FIXED burst types remain constrained to a maximum burst length of 16 beats. These are the drawbacks of AMBA AXI system which need to be overcome.

VI. REFERENCES

[5] Silicore Corporation, Wishbone system-on-chip (soc) interconnection architecture for portable ip cores,

Author’s Profile

Mrs. B. Beena Pursuing M.Tech in VLSI Systems Design from TKR Engineering College. Her area of interest is VLSI design and Embedded systems.

Mrs. B. Nireesha Asst. Professor, in TKR Engineering college. Her area of interest is VLSI systems and communication theory.