

High Performance AXI4 to AHB Bridge (Using a local interface)

Dhwani Bhatt

E-VLSI & Embedded System Design
Gujarat Technological University
Gandhinagar, Gujarat, India.
Email: Dhwani.bhatt19@gmail.com

Mr. Santosh Jagtap

Wipro Ltd.,
Pune, India.
Email: santosh_s_jagtap@yahoo.co.in

Abstract- In this paper, we have presented the design of a high performance AXI to AHB Bridge. The AMBA (Advanced Micro controller Bus Architecture) developed by ARM is a widely used interconnection standard for System on Chip (SOC). The AMBA 4.0 supports a rich set of bus signals, hence supporting high speed pipelined data transfers. The goal of this project is to synthesize and simulate an interface bridge for Advanced High performance Bus (AHB) to support for both high bandwidth data transfer using a single AXI4.0 transaction. The bridge provides interface between the high performance AXI 4.0 and high bandwidth peripherals of AHB domain. It has a slave interface which receives the AXI4 master transactions and converts them to AHB master transactions and initiates them on the AHB bus. Here we will also be using a local interface with the system. and one of the main advantages of which is that in future while replacing any of the protocols on the either side, the entire system does not need to be programmed, but only the protocol that has been replaced.

I. INTRODUCTION

A. AXI 4

The AMBA3.0 specifications that were developed by ARM introduced the AXI 3 protocol, which was later modified and was introduced as AXI 4 protocol in AMBA 4.0 specifications. The AXI protocol is mainly connected to the high frequency like processor, controller, DSP etc.

The AXI4 consists of five different channels, which are stated as:

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

The write address channel and read address channel generally provide us with the address and control information. The write data channel is used by master in order to write any7 given to the slave, on the other hand, the read data channel is used by master to read any data from the slave. The write response channel is directed from a slave to master and it carries the information indicating the success or the failure of a particular transfer. Thus both the read and write operations can continue simultaneously due the existence of different channels.

B. AHB-Lite

The AHB protocol was introduced in the AMBA 2.0 specifications that were introduced by ARM. Later the AHB-Lite was introduced in the AMBA 3.0 protocol. The AHB-Lite has the same functioning as AHB itself, but it incorporates only some features of AHB and not all.

The AHB Lite is used with high bandwidth devices such as

high bandwidth on chip RAM, high bandwidth memory interface etc.

II. SYSTEM BLOCK DIAGRAM

The figure below shows the block diagram of the proposed design.

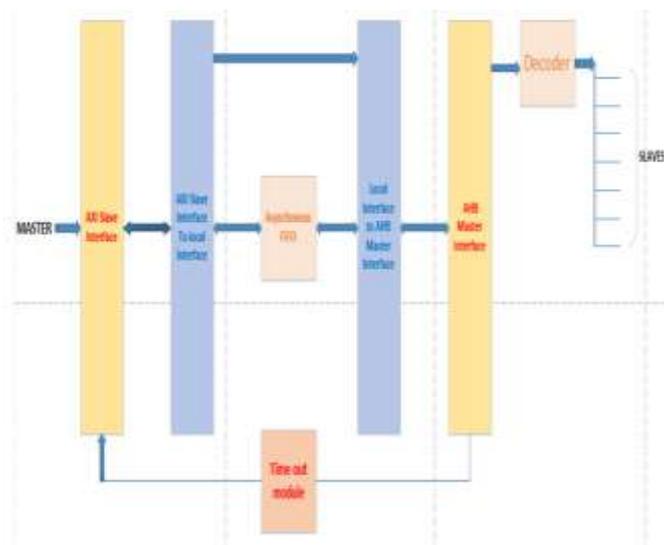


Fig. 1 System Block Diagram

The proposed design of the system is as shown in the figure. The master will first of all send the address and control information to the slave interface (in our case AXI4), which will in turn be sent to the local interface and consequently to the slave device. Once the slave is ready for communicating

and generates an OK signal, the data transfer starts. The data is sent to the FIFO, if the FIFO has sufficient space, then it will accept the data, otherwise it will send the data back and the transfer will be terminated and due to an error and thus it needs to start all over again. The time out module is used by the slave to generate an error signal, when the transfer is taking more than is desired or allotted to it.

III. INTERNAL BLOCKS

A. AXI4 slave interface:

The FSM for the read and the write cycles of the AXI4 interface are as under:

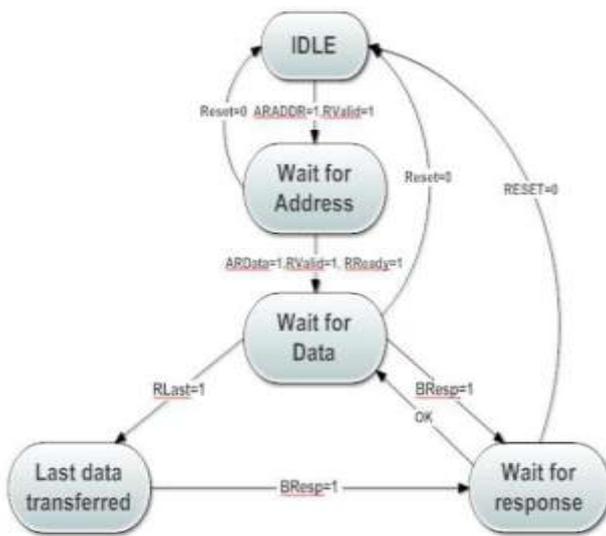


Fig. 2. Read FSM

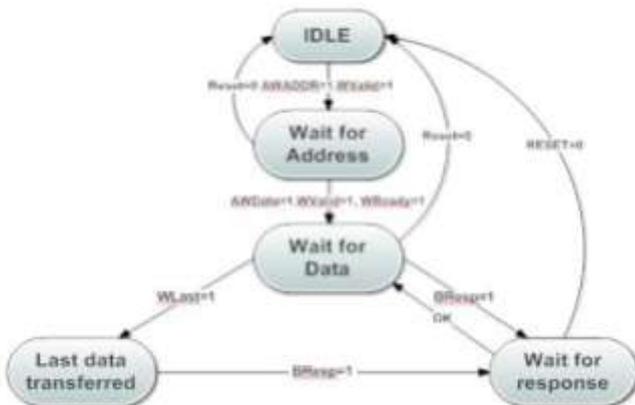


Fig.3 Write FSM

The state machines in the above figures show the read and the write state machines of the AXI4 slave interface. Whenever the Reset signal is active low, then the state machine will be in the idle state. Now whenever the address signal is available then the wait for address state will be executed, followed by the wait for data on availability of the data. Whenever the last data is

transferred, then the FSM will switch to the last data transferred state and then depending on the response received from the slave side, the wait for response state will be executed. In between if at any point the reset signal goes low then the state machine will jump to idle state. The working of both the FSMs is same, the only difference is the channels on which these signals are present.

B. AHB-Lite Master Interface

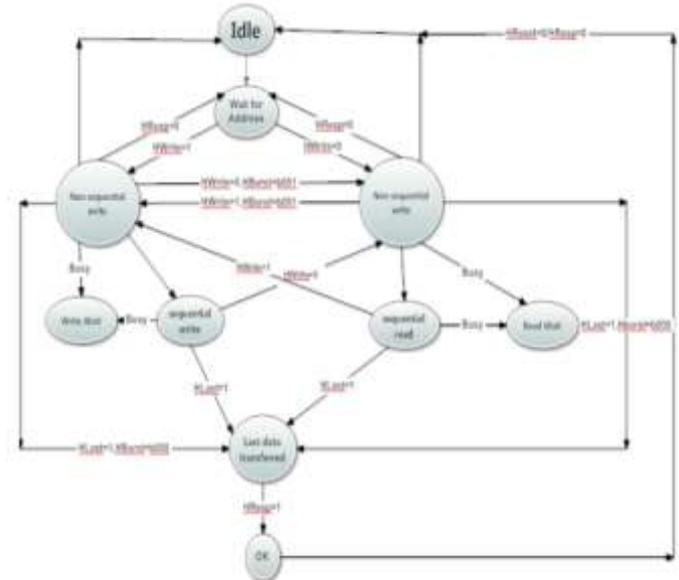


Fig.4 AUB Master FSM

The FSM for the AHB-Lite master interface is as shown in the figure above. Whenever the reset signal is low, then it will be in the idle state. Now as soon as the address is received, it will go to the write address state. Now depending upon the HWRITE signal status as high or low, it will execute the non sequential write or non sequential read signal response. Now depending on the burst, type of transfer and the status of write signal, the further states will be executed as shown in the FSM. Finally when the HLAST signal is high, then the machine will jump to the last data transferred state. and depending on the slave response, the response state is executed.

IV. SIMULATION RESULTS

The simulation results of some of the modules are shown as under:



Fig. 5. AHB-Lite Master Interface FSM simulation results

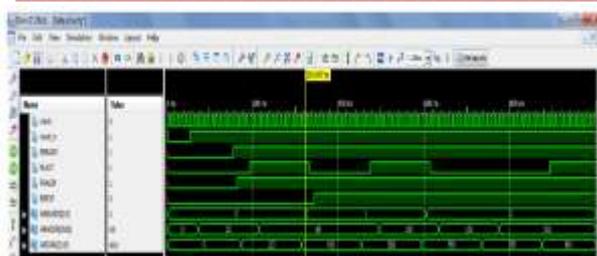


Fig. 6. AXI Slave Interface Read FSM simulation results

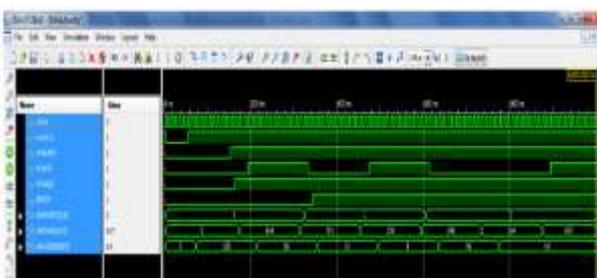


Fig. 7. AXI Slave Interface Write FSM simulation results



Fig. 8. AXI to local interface simulation results

V. CONCLUSION

Thus we conclude that with the help of address decoder, we can select the single slave that is needed with the help of single input given by the master. The multiplexer helps to multiplex the data that is sent by all the slaves and send it to the master. Finally there is the FIFO which can be used for data storage while transferring data between systems with different speeds and clock. It helps to get synchronization in between clocks passing through different domains.

REFERENCES

- [1] AMBA AXI and ACE Protocol Specification ARM, 2011
- [2] A Distributed AXI-based Platform for PostSilicon Validation ,”M. H Neishaburi, Zeljko Zilic McGill University, 3480 University Street, Montreal, Quebec Canada H3A 2A7.Mh.neishabouri@mail.mcgill.ca, zeljko.zilic@mcgill.ca , 2011 29th IEEE VLSI Test Symposium
- [3] Towards a Configurable Many-core Accelerator for FPGA-based Embedded Systems, Marco Ramirez, Masoud Daneshtalab, Pasi Liljeberg, Juha Plosila Department of Information Technology, University of Turku Email: maanrl, masdan, pakrli, juplos@utu.fi t, IEEE 2013
- [4] VLSI Design of Multichannel AMBA AHB, Shraddha Divekar, Archana Tiwari M-Tech, Department Of Electronics, Assistant professor, Department Of Electronics RKNEC Nagpur, RKNEC Nagpur,IEEE-2014
- [5] Implementation of an On-Chip Bus Bridge between Heterogeneous Buses with Different Clock Frequencies, Sangik Choi and Shinwook Kang, Mobile Convergence Team, Digital Media RD Center, Samsung Electronics Co., Ltd., Suwon-city, Kyunggi-do, Republic of Korea, IEEE-2011
- [6] Design and implementation of APB Bridge based on AMBA 4.0, Chenghai Ma, School of Information Science and Engineering, Shandong University, Jinan, China Zhijun Liu, School of Information Science and Engineering, Shandong University, Jinan ,China Xiaoyue Ma, Department of Electronics, Shantou University, Shantou, China, IEEE-2005
- [7] Design of AMBA AXI4 protocol for System-on-Chip communication, Shaila S Math and Manjula R B, Dept. of ECE, REVA Institute of Technology and Management, Bangalore, India. International Journal of Communication Network and Security (IJCNS)-2014
- [8] AMBA 3 AHB-Lite Protocol v1.0 Specifications (www.arm.com)-2006
- [9] AMBA AXI4.0 Protocol Specifications (www.arm.com)-2010
- [10] AXI4 Specifications- Xilinx
- [11] AHB-Lite Specifications - Xilinx