IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 5, Issue 8, March 2017)

Multi-Master Supported Arbiter in AHB System

Sneha Devasthali
snehadevasthali189@gmail.com

Preet Jain
preetjain@gmail.com

Abstract –**Most of the FPGA features communicate with each other through bus means. Each block is designed for a specific bus. The buses related to this work is the AMBA bus and the Wishbone bus. It works with the master / slave architecture. The AMBA bus is extensively utilized in the system on chip solution for interaction with different peripheral. The proposed work includes the regular AMBA specifications of (1) burst transfer, (2) Several bus masters, (3) single clock edge transition and (4) Split transaction. The data transfer is accomplished by a single bus master at a while. Despite the fact that the arbitration protocol is settled, any arbitration algorithm, for example, fair access or highest priority can be actualized relying upon the application necessities. VHDL code is utilized to develop the design and it is synthesized on Virtex-2 series.**

Keywords – **AMBA, AHB, APB, ASB, FPGA, VHDL.**

I. INTRODUCTION

Innovative advancements on the global industrialization have permitted the joining of an ever increasing number of capacities on the same digital integrated circuit. Now a days, various hardware accelerators, microprocessors, communications systems are classified with operating systems. This permits along these lines achieve every one of the capacities important to perform complex computer processing on the similar chip, thus the introduction of the idea of System on Chip, successors of specialized circuits ASIC (Application Specific Integrated Circuit) [1].
The choice of the communication system, in a system On-chip, remains a major problem. This choice depends on components constituting the system and especially the processor used. SoCs design software platforms, proposed by the manufacturers of the FPGA circuits, proposes a well-defined type of communication system [2]. These systems are generally standard buses modeled in one language description of hardware like the VHDL. Recently, with the aim of improving quality and speed communication systems in new systems, networks On-chip (NoC: Network-On-Chip) have been introduced [3]. The principle of NoCs is to ensure parallel between the components of a system on chip.

The AMBA Bus

The Advanced Microcontroller Bus Architecture (AMBA) bus was developed in 1995 by ARM for its own processors and is extensively used as an on-chip bus. There are three categories of AMBA bus: AHB, ASB, and APB.

- The AHB (Advanced High-Performance Bus) bus is characterized by its high speed and high performance. It is further divided into two categories, the master bus (AHB master) and the slave bus (AHB slave). It is the bus that is connected to the masters and fast slaves.
- The Advanced System Bus (ASB) bus is a bus that can replace the AHB bus, but it is less efficient. It is therefore little used, and besides it is not used in this project.
- The Advanced Peripheral Bus (APB) bus is optimized for least power consumption, slave design simplicity and peripheral operation. It is the bus that is connected to slow slaves.

The following figure shows the general view of an AMBA bus circuit:

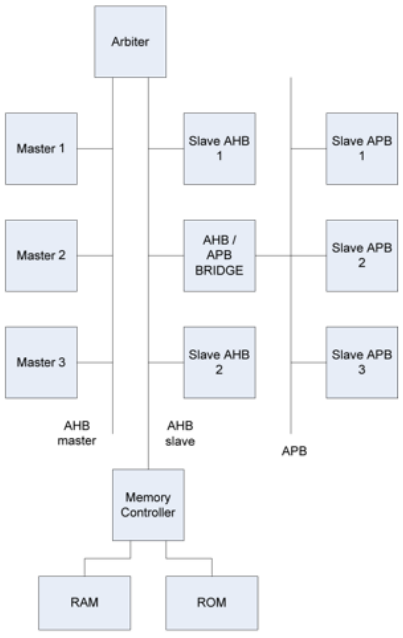

Figure 1: AMBA bus generalized diagram

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 5, Issue 8, March 2017)

Masters are usually processors. They communicate via the arbitrator by specifying the address of the slave with which they wish to exchange data. Each slave can give an answer on his own bus that goes back to the masters. As the name suggests, the referee manages the transfers and priorities of the masters. In addition, it acts as a bridge between the AHB master bus and the AHB slave bus. The memory controller handles RAM and ROM, but it can also take care of IO.

Figure 2 shows an example circuit that shows the operation of the AHB master and AHB slave buses. On the master side, the arbiter has as its output a single bus (containing the data and control bits) which will be the master input (ahbmi) and has as input all the output busses (ahbmo) of each master. The same principle is found on the Slavonic side, but with different signals.
In general, the master makes a query on the ahbmo bus to read or write on a slave. The referee takes care of sending it to the good slave via the bus ahbsi. The slave answers on the ahbso by specifying the master number. Finally, the master receives the answer through the referee via the ahbmi bus.
There is therefore an "In" and "Out" part for each AHB bus.
In this project, the AHB master bus is not required to be studied in depth, but the slave AHB bus must be well detailed. The following table shows the signals of the ahbsi bus:

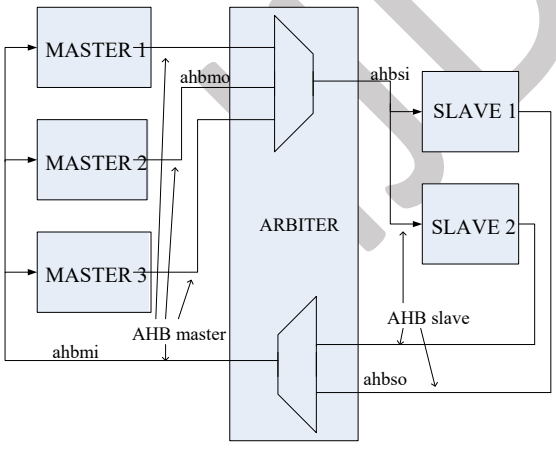

Figure 2: AMBA arbiter

In this research work an industry-standard arbiter is implemented for AHB (AMBA). The specification describes the bus attributes, the protocol definition, and types of transactions, bus management, and the programming interface required to design.

II. Proposed Methodology

Arbitration to pick the subsequent bus master utilized a round robin arbitration scheme. This guarantees no master gets empty. At the point when a master has sealed the bus, the round robin arbitration is abrogated and the master with the seal holds most noteworthy need to the bus.
The sixteen AMBA Bus masters are Master 0 through Master 15. Slave 0 through Slave 15 are the sixteen AMBA Bus slaves. The AMBA AHB Bus Arbiter/Decoder contains a default master-Master0, and a default slave- Slave 0.
AMBA AHB Bus Arbiter features are summarized:

- AMBA AHB Bus arbiter function
- Round robin arbitration
- Default master- Master 0
- Default slave- Slave 0

The Arbiter block monitors the AMBA Bus for requests and chooses the master with highest priority request as the next AMBA bus transaction master. If there are no requests, the Default Master is chosen as the master to drive the next AMBA Bus transaction.

Testing the IP

The method of testing the IP was kept simple since it was more important to concentrate on the functionality of the IP.

The Subsystem Specifications

The AHB ARBITER IP can be broken into two subsystems. The two major components of the system under design are the controller and data path.
Controller: The controller is essentially a Mealy state machine. It keeps track of the different sections of an arbiter transaction. The first state of the controller is the start state. The next state to check the grant if grant is there then it will make the necessary signal high which will further control the counter, and counter interface block.
Data Path: The Data path are further divided into several sub system blocks, few of them are controlled by controller.

Overall Description of Design

The followings are the major steps involves in the Arbiter designs form architectural or functional point of view:
The bus_request of different masters has to pass through the bus_req block which is responsible to pass the request to other logical blocks. This block is depends upon the enable pin which is coming from priority storage block.
Bus_request further pass through interface block and goes to priority logical block. The interface block is

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 5, Issue 8, March 2017)

giving the enable signal to priority logical block and interface block is responsible for monitor the data transaction through data_done signal, it can assert and deassert the enable pin depends upon the data_done.

This bus_req goes to the priority logic block. this block further decide that which master request will get the highest priority depending upon the priority this block is generate the Grant signal .

This grant signal goes to priority storage block, encoder block and as output port to interact with Master. After getting the Grant signal Master will send address, burst to indicate the type of transfer, and slave will also send Hready, Hresp, Hsplit.

At the same time when master samples the signals to the arbiter grant signals which are the output of the arbiter pass through the mux, inside the arbiter, for this mux bus_master no is select line, which indicates that which master is accessing the bus.

The output mux then passes to the controller block which will generate the necessary signals for counter, i.e. the controller will control the operation of counter.

The grant output from the priority logic block is or and then sent to the priority storage block which will store the priority and pass the enable signal to the next priority depending upon the grant value. And the whole operation is repeated depends upon the transaction mode.

III. Simulation And Results

Synthesis has been carried out on vertex-2 board using Xilinx13.1 ISE.

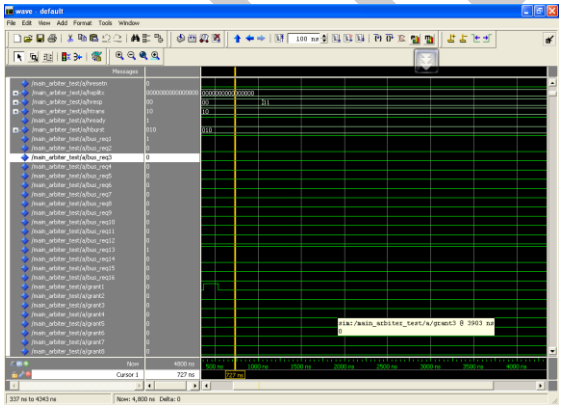

Figure 3: Simulation waveform of main arbiter: top entity of arbiter

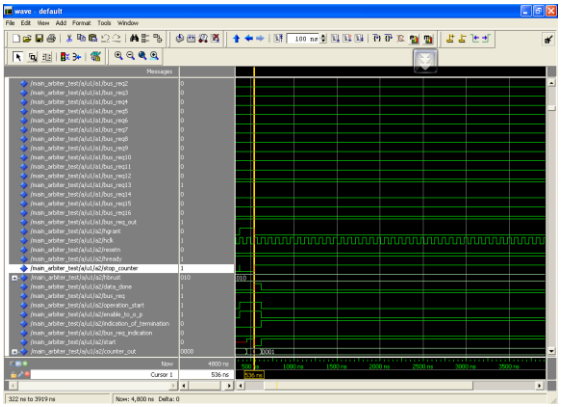

Figure 4: Simulation waveform of counter interface

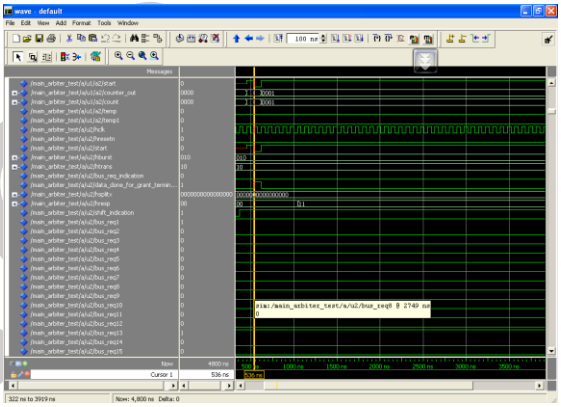

Figure 5: Simulation waveform of priority logic

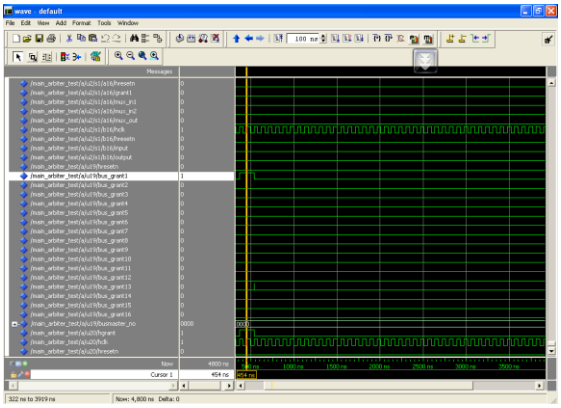

Figure 6: Simulation waveform of Encoder

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 5, Issue 8, March 2017)

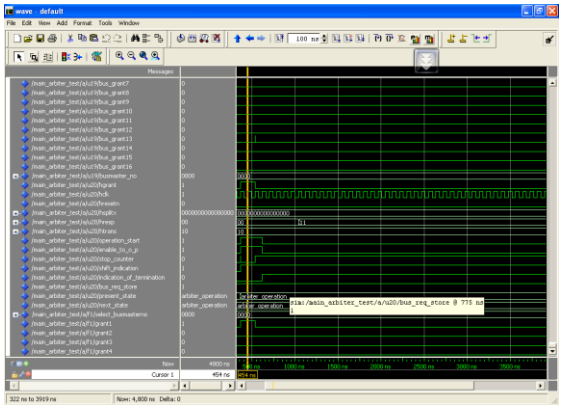

Figure 7: Simulation waveform of Controller

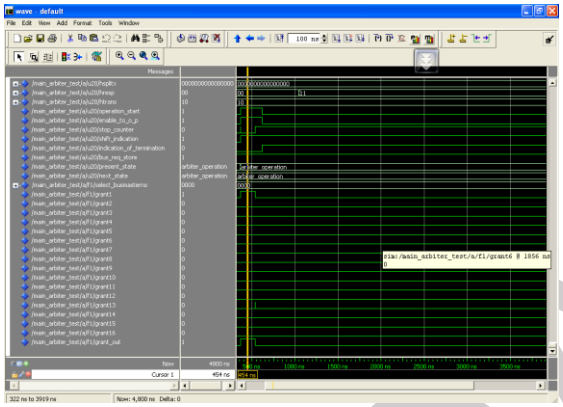

Figure 8: Simulation waveform of Grant Mux

IV. CONCLUSION

The design has been developed using VHDL code and synthesized using Xilinx ISE 14.2 ISE. The design is simulated on Modelsim 6.5 & verified through effective test bench. The advantage of this design is that we have taken care of latch formation, as it is a FPGA implementation hence with less latch & maximum flip-flop have enhanced our area efficiency.

REFERENCE

[1] Kees Goossens, Om Prakash Gangwal, Jens Rover, and A. P. Niranjan, "Interconnectand Memory Organization in SOCs for advanced Set-Top Boxes and TV-Evolution, Analysis, and Trends", In JariNurmi, HannuTenhunen,JouniIsoaho, and Axel Jantsch, editors, Interconnect-Centric Designfor Advanced SoC and NoC, chapter 15, pages 399–423. Kluwer, April 2004.

[2] Poletti, Francesco, Davide Bertozzi, Luca Benini, and Alessandro Bogliolo. "Performance analysis of arbitration policies for SoC communication architectures." Design Automation for Embedded Systems 8, no. 2-3, pp. 189-210, 2003.

[3] Huang, Yu-Jung, Yu-Hung Chen, Chien-Kai Yang, and Shih-Jhe Lin. "Design and implementation of a reconfigurable arbiter", In 7th WSEAS International conference on signal, speech and image processing (SSIP'07), Beijing, China. 2007.

[4] Shanthi, D., and R. Amutha, "Design Approach to Implementation Of Arbitration Algorithm In Shared Bus Architectures (MPSoC)", Computer Engineering and Intelligent Systems 2, no. 4, pp. 185-196, 2011.

[5] E. Raja, K.V. Ramana, "Implementation of Multilayer AHB Bus matrix for ARM", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2011.

[6] Dr. Fazal Noorbasha, B. Srinivas, Venkata Aravind Bezawada, V. Sai Praveen, "Implementation of an Adaptive-Dynamic Arbitration Scheme for the Multilayer AHB Busmatrix", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue 4, pp.825-831, July-August 2012.

[7] Jalle, Javier, Jaume Abella, Eduardo Quinones, Luca Fossati, Marco Zulianello, and Francisco J. Cazorla. "AHRB: A high-performance time-composable AMBA AHB bus." IEEE 20th Symposium in Real-Time and Embedded Technology and Applications (RTAS), , pp. 225-236, 2014.

[8] Shraddha Divekar, Archana Tiwari, "Multichannel AMBA ARB with Multiple Arbitration Technique", IEEE, International Conference on Communication and Signal Processing, April 3-5, pp. 1854-1858, 2014.

[9] Pravin S. Shete, Dr. Shruti Oza, "Design of an AMBA AHB Reconfigurable Arbiter for On-chip Bus Architecture", International Journal of Application or Innovation in Engineering & Management (IJAIEM), ISSN 2319 – 4847, Vol. 3, Issue 5, May 2014.

[10] Shashidhar R., Sujay S. N., Pavan G. S., "Implementation of Bus Arbiter Using Round Robin Scheme", International Journal of Innovative Research in Science, Engineering and Technology, ISSN: 2319-8753, Vol. 3, Issue 7, July 2014.

[11] Mrs. R. Ramya, Ms. Preethi Vadivel, "Slave-Side Arbitration and Implementation For Multilayer - AHB Bus Matrix", International Journal of Engineering Research and Reviews, ISSN 2348-697X, Vol. 2, Issue 3, pp: 56-60, September 2014.