IJRRECS/August 2013/Volume-1/Issue-4/316-321 **ISSN 2321-5461**

INTERNATIONAL JOURNAL OF REVIEWS ON RECENT ELECTRONICS AND COMPUTER SCIENCE

AMBA-AXI COMPLIANT MEMORY CONTROLLER

Santhoshi Yadav Pulicharla[1], V. Koteswara Rao[2]

[1]M.Tech Student, Dept of ECE, QIS Institute of Technology, Ongole, A.P, India

[2]Associate Professor, Dept of ECE, QIS Institute of Technology, Ongole, A.P, India

ABSTRACT:

The performance of the processors is increasing day by day as the technology is advancing. Yet these new processors need to work in interface with both the low speed peripherals as well as high speed peripherals. The performance of the peripherals like memory did not increase that much. Hence there is the need to match the high performance with the low and different speed peripherals. The AMBA-AHB memory controller has been developed to meet these demands. But the AMBA-AHB memory controller has been designed to meet these needs to interface the AMBA-AHB bus with the different peripherals of different speeds both high and low. The AMB-AHB memory controller can be used for interfacing with only one master and of half duplex operation. This paper presents the implementation of memory controller that can interface with Advanced Extensible Interface (AMBA-AXI) bus which is full duplex and also two masters can communicate at the same time. This design can be used for frequencies above 400 MHz which can be used with the latest high end processors. The AMBA-AXI does not require any bridge like AMBA-AHB. This AMBA-AXI compliant memory controller can be used with the latest ARM processors as well. This memory controller is mainly designed for interfacing the memories like SRAM and ROM and matches well with the memory latencies. Here the Memory controller is designed to improve the system performance. As the Memory Controller is On-chip, the memory accessing time is decreased so that performance is increased. The controller is implemented in Verilog HDL in XILIX ISE 12.1 for porting into the target device Spartan 6 FPGA.

Keywords: ARM, AMBA-AXI, Memory controller, Microprocessor.

1. INTRODUCTION

Authors [1] implemented an AMBA-AHB compliant memory controller which works only with AMBA-AHB bus architecture. AMBA has the disadvantage that it can be used with only one master and half duplex in operation. To make it full duplex and to work with two or more masters, the AXI system bus specification is developed. Hence in our present work a memory controller that is compliant with the AMBA-AXI BUS has been developed.

AXI-MC Features:

The key features of the AXI protocol are:

• It is Full-duplex that is can read and write at a time to two channels

• It provides ID specification

• Uses 3 types of burst transactions like- Incrementing, Wrapping, Fixed

• Low-cost *Direct Memory Access* (DMA) can be provided to separate read and write channels

• Support for issuing multiple outstanding addresses

• Support for out-of-order transaction completion

• Cache can be used

1. ARCHITECTURE OF AXI-MC

The AMBA-AXI memory controller is mainly divided into three parts like: AXI Slave interface, FIFO, External Memory Interface. Figure 1 shows the architecture of AXI-MC.

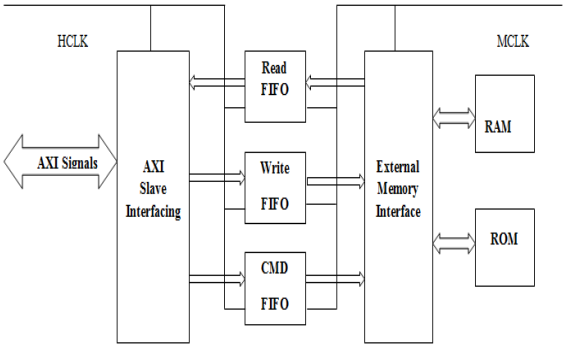

FIGURE: BLOCK DIAGRAM OF AXI COMPLIANT MEMORY CONTROLLER

Figure 1: Architecture of AXI- MC

(A).AXI Slave Interface

The AXI Slave interface plays a vital role in AXI Memory controller. It is used for converting the incoming AXI transfers to the protocol used by the AXI-MC. In AHB, there are no control signals like valid, Awready etc. In AXI Slave Interface there are various control signals like Awready, Awid, and Awresponse etc as shown in figure 2.

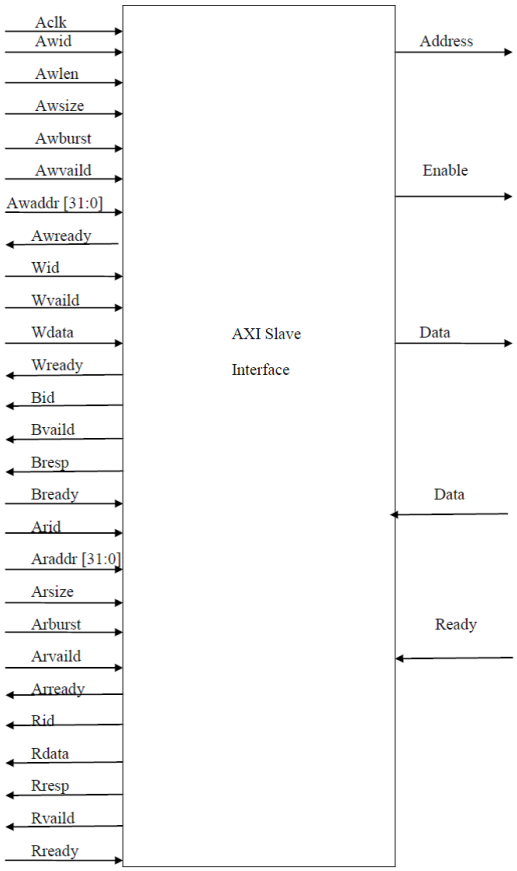

Figure 2:AXI slave interface

As AXI is full duplex, read and write can be performed at a time that is due to channels like

1. Write Channel-Address, data, Response
2. Read channel-Address, Response and Data

Here 2 clocks are used Hclk and Mclk. As AXI slave interface operates at high frequency clock and external memory interface operates at low frequency clock. To get the synchronization between two clocks FIFO is used.

[B].External Memory Interface

It is used for selection for the memory read/write. RAM & ROM are selected based upon the MSB bit .If MSB bit is high, then RAM is selected and if MSB bit is low ROM is selected and then depending upon 30^{th} and 29^{th} bit selection of banks is done.

[C].FIFO

In FIFO internal address will be specified. In FIFO clocks will be synchronized. In FIFO[2]full and empty conditions will be occurring.in FIFO there are three types of FIFO. They are as

Follows:

1. Read FIFO

2. Write FIFO

3. Command FIFO

Command FIFO**:**

When Master i.e. processor wants to send or receive the data then first address is to be located .So the address that which is sent by the processor through address bus will be send to the Command FIFO and from there to the External Memory interface there chip selection will be done.

Write FIFO:

If data is to be written from AXI to External Memory interface then Write FIFO is used

Read FIFO:

If data is to be read from the RAM & ROM to the Processor then the Read FIFO is used.

2. METHODOLOGY

As System frequency is improving it becomes harder for address decoding in one clock cycle, so wait states are being inserted into data cycle for enough time being assured for address decoding .A Burst Method is used for solving the problem in this paper . In AXI, it uses up to 256 transactions of wrapping type but incrementing transactions only 16 like AHB. By using this burst transaction power is consumed.

In ARM, the instruction is 32-bit while RAM and ROM are 8-bit instructions. Due to this the two addresses which are lower of ROM and SRAM are not connected so bank selection is done as shown in the below figure3.

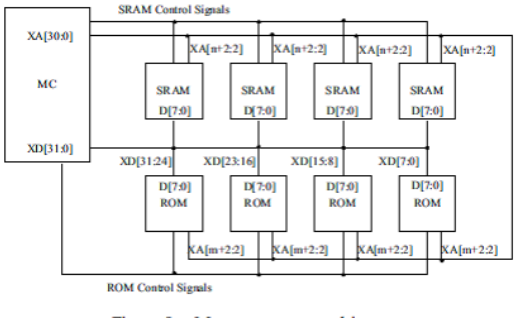

Figure 3: Memory System Architecture

Here the implementation of the present design oriented method is shown in the figure 4 which shows the Read and Write states of RAM and ROM .Here the present method completely overcome the drawbacks of the several previous methods in terms of the degraded performance based strategy followed by the improvement in the system with respect to the present method in a well oriented fashion respectively.

3. EXPECTED RESULTS

A comparative analysis is made between the present method to that of the several previous methods in a well effective manner and are shown in the below figure 3 in the form of the elaborative fashion respectively .Here in the below figure we can see the read and write with zero wait states of ROM and RAM by using AXI Memory Controller.

By using Asynchronous clocks [4], we can improve the system

performance.Figure 4 shows that the data transferred for the RAM and ROM with zero wait states.

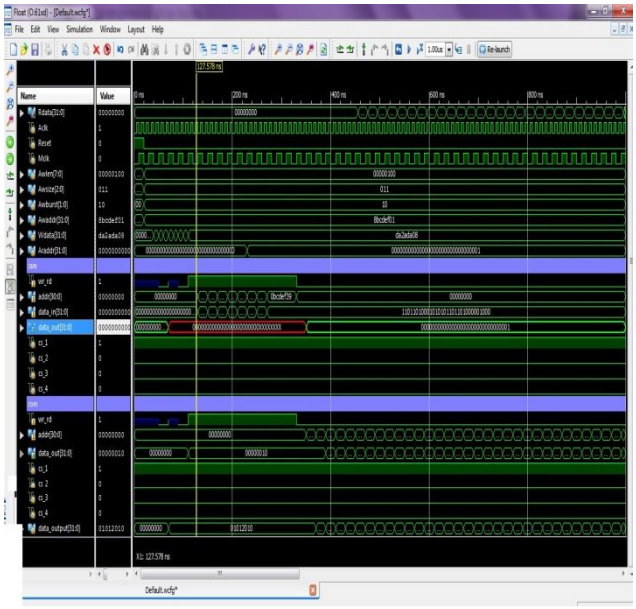

Figure 4: Read and write with zero wait States of ROM and RAM

4. CONCLUSION

In this paper the implementation of a memory controller that is AXI compatible is presented. As the new bus technology specifications are developed, then there is the need to develop a memory controller compatible with the new specifications.

REFERENCES

[1] Hu Yueli, Yang Ben "Building an AMBA AHB compliant Memory Controller" IEEE, 2011 Third International Conference on Measuring Technology and Mechatronics Automation.

[2] AMBA® AXI™ and ACE™ Protocol SpecificationAXI3.

[3]Simulation and Synthesis Techniques for AsynchronousFIFO DesignClifford E. Cummings, Sunburst Design, Inc.

[4].Carter, J.; Hsieh,W., "Impulse: building a smarter memorycontroller", High-Performance Computer Architecture, Dept. ofComputer. Sci., Utah Univ., Salt Lake City, UT.

[5].Clifford E. Cummings, "Synthesis and scripting Techniques forDesigning Multi-Asynchronous Clock Designs", Sunburst Design, Inc661.

AUTHORS:

1. Santhoshi Yadav Pulicharla is a II year M.Tech. (VLSI and Embedded Systems) student of QIS Institute of Technology, Ongole , AP, India affiliated to JNT University , Kakinada.

Sri V. Koteswara Rao received his B.Sc, degree from Andhra University, waltair. He received his AMIETE degree in electronics and communication engineering from IETE, India. He received his M.Tech in Embedded Systems and VLSI design from JNTU, Hyderabad. He is a member of IEEE, USA., Fellow of IETE., India and a Professional member of ACM, USA. He worked for AP state govt., India as Deputy

Executive Information Engineer for several years. He is presently working as Associate professor, in ECE department of QIS Institute of technology, Ongole, AP, india. He is presently also a research scholar in VLSI at Vignan University, Vadlamudi, Guntur Dt., AP., India.