IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

RTL Implementation of AXI Interface for Memory Controller

Dr. K.C. Mahajan[1], Mukesh K Yadav[2]
Professor, Technocrats Institute of Technology (Excellence), Bhopal Madhya Pradesh[1]
PhD Scholar (Electronics) Mewar University Chittorgarh, Rajasthan[2]

Abstract –**This paper proposes an implementation of AXI 2.0 protocol which removes the limitation of communication architecture, which would otherwise reduce the speed of data transfer in System on chip. We have also implemented DDR3 controller which was then interface with AXI 2.0 protocol. Proposed protocol was synthesized on Xilinx 13.1 and simulated using Modelsim 6.5e.**

Keywords – **AXI, DDR3, Modelsim, Xilinx.**

I. INTRODUCTION

With the need of application, chip with a single processor can't meet the need of more and more complex computational task. We are able to integrate multiple processors on a chip thanks to the development of integrated circuit manufacturing technology Now as there are multiprocessing units and processors is getting faster, so compatibility with slow communication architectures a bit difficult furthermore this slow and conventional communication architecture limits the throughput.
To improve the performance we have to develop such efficient on chip Architecture which will be much faster system on chip solution which removes the limitation of communication architecture one of the solution is "AHB bus" but it can't give perfect parallelism as it can allow only one master to communicate at one slave only. While in our design there are five independent transfer channels which make multiple masters access multiple slaves at the same time and gain a perfect parallelism performance in MPSOC design.

As we have seen in AXI 1.0 protocol to achieve high speed communication between processor for on chip communication while we have developed AXI 2.0 Protocol. To improve the performance we have to develop such efficient on chip architecture which will be much faster system on chip solution which removes the limitation of communication architecture. One of the solution is "AHB bus" but it can't give perfect parallelism as it can allow only one master to communicate at one slave only. While in our design there are five independent transfer channels which make multiple masters access multiple slaves at the same time and gain a perfect parallelism performance in MPSOC design.
In this research work, AXI 2.0 protocol is implemented which removes the limitation of communication architecture, which would otherwise reduce the speed of data transfer in System on chip have also implemented DDR3 controller which was then interface with DDR3.

II. PROPOSED METHOD

The AMBA AXI protocol is targeted at high-performance, high-frequency system designs and includes a number of features that make it suitable for a high-speed submicron interconnects [1].

The objectives of the latest generation AMBA interface are to be suitable for high-bandwidth and low-latency designs Enable High Frequency operation using complex bridges meet the interface requirements of a wide range of components be suitable for memory controllers with high initial access latency provide flexibility in the implementation of interconnect architectures be backward-compatible with existing AHB and APB interfaces. The key features of the AXI protocol are:

- Separate address/control and data phases.
- Support for unaligned data transfers using byte strobes.
- Burst-based transactions with only start address issued.
- Separate read and write data channels to enable low-cost Direct Memory Access (DMA).
- Ability to issue multiple outstanding addresses.
- Out-of-order transaction completion.
- Easy addition of register stages to provide timing closure.

Flow Diagram for DDR3 Controller

In this block diagram 1, we are showing that how our interconnect communicate with DDR3 RAM suppose master wants to send some data to the DDR3 RAM then first req is transferred to master interface which made valid signal high to pass req of processor to transfer the req to the interconnect and

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

finely to the DDR3 controller. This required information of valid signal may be respond positively at slave itself by asserting the ready signal high. Note that, according to protocol the communication between master and slave may be established once both ready and valid signal are made high once when both the signal high then the communication between interfaces established and data transfer will be carried out. Here in this diagram the decoder has purpose of selecting the DDR3 controller as one of slaves.

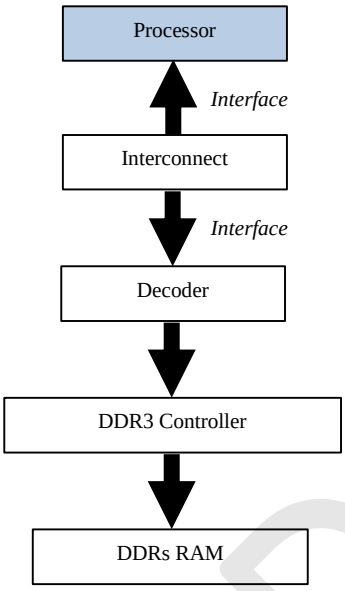

Figure 1: AXI Interface Protocol Interfaced with DDR3 Controller

A typical system consists of a number of master and slave devices connected together through some form of interconnect, as shown in Figure 2.

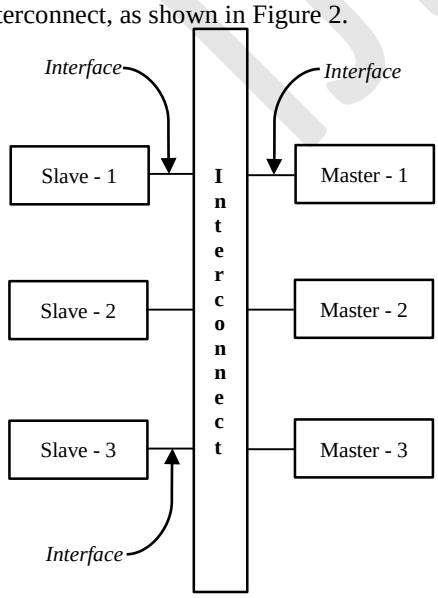

Figure 2: Interconnect

Interconnect is the main heart of this communication bus as it provides all the intelligence. It contents five individual channels for transferring the address and data and address generator. It is also able for transaction reordering mean suppose master1 wants to communicate the slave1 and slave 2 first it sends the address to the slave 1 and some no of bytes require to transfer.

The AXI protocol provides a single interface definition for describing interfaces:

- Between a master and the interconnect
- Between a slave and the interconnect
- Between a master and a slave

The interface definition enables a variety of different interconnect implementations. The Interconnect between devices is equivalent to another device with symmetrical master and slave ports to which real master and slave devices can be connected. Most systems use one of three interconnect approaches:

- Shared address and data buses.
- Shared address buses and multiple data buses.
- Multilayer, with multiple address and data buses.

In most systems, the address channel bandwidth requirement is significantly less than the data channel bandwidth requirement. Such systems can achieve a good balance between systems performance and interconnect complexity by using a shared address bus with multiple data to enable multiple data transfer.

Block Diagram of Write Address Channel

Figure 3 shows the block diagram of Write address channel. Here idappender has responsibility of adding the additional bit to the wid to let the slave known that from which master the particular data or address being transferred then six bit id is transferred to the idstorage block then form id storage, id is transferred to the Multiplexer form. Address storage address is transferred to Multiplexer. Here in the project we have used 4 Masters, those 4 masters are simultaneous to the Slave. Masters are giving their different control signal to the Multiplexer. All Masters are sending the valid signal to Arbiter. Arbiter can select one of the master on fixed priority basis which is known as Round Roubin method. Here in our project, arbiter is selecting the particular Master on the basis of fixed priority, hence arbiter selects the master by issuing the grant to particular master then grant signal is given to the encoder. On the basis of the grant signal, it provides the SEL signal to Multiplexer. Hence Multiplexer decides or selects the Master. Multiplexer provide all signal of selected master to the decoder and finally decoder

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

selects the slave to which address needs to be send. Decoder selects the slave on the basis of the address sent by the slave, hence address is passed from master to slave. The function of write data signal is identical to that of write address channel so only address.

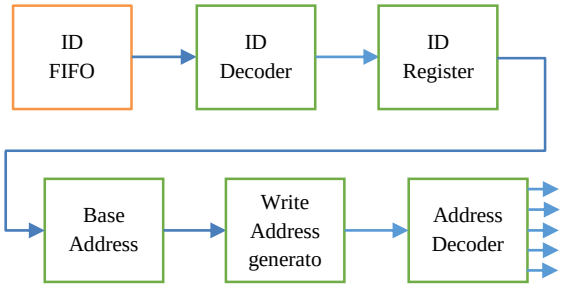

Figure 3: Block diagram of Write Address channel

Block Diagram of Read Address Channel

The function of read data signal is identical to that of write address channel. First of all six bit id from id storage would come on to the port from slave, four signal such as rlast, rready data, rvalid would come on to the slave port. From slave port, six bit id would go to the Id separator and decoder. Id separator/decoder gives four bit Id to the Encoder. Encoder, according to grant Id, will activate select (sel) line. Then four output form slave port will go to demultiplexer. According to selection line, demultiplexer will send the id for particular Master.

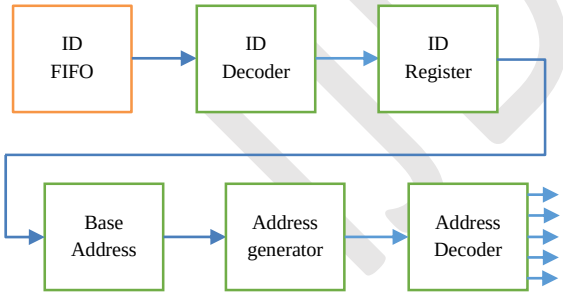

Figure 4: Block diagram of Read Address channel

Block Diagram of Write Response Channel

In AXI interconnect write operation needs to be responded, this response is not given for signal transfer. Response operation is started when completion signal occurs for a burst. The completion signal occurs only after completion of signal burst not for signal transfer that means after completion of every burst, completion signal is generated and this response is given through response channel. here in this block diagram 1, 2, 3 and 4 are respectively bvalid, bid1, bvalid and bready and 5, 6, 7 and 8 are bvalid1, bid1, bvalid and bready1. In this protocol, four types of responses are occurs; OK, Exclusive OK, SLERR and DECERR.

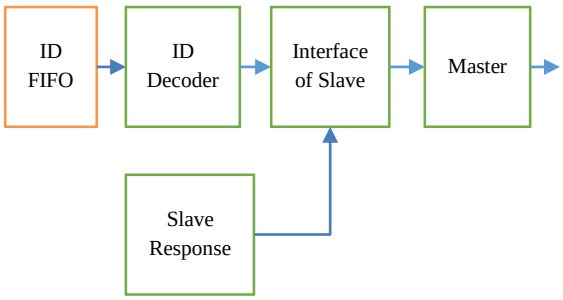

Figure 5: Block diagram of Write Response Address channel

The operation starts as completion signal is generated, which leads to generation of Bvalid signal by slave. In response to the assertion of Bvalid signal, Bready signal also gets high then signal of slave port (Bvalid bready bid bresponse) also gives 6 bit Id signal to Id decoder/separator as result of which Id decoder/separator. According to 6 bit Id, grant signal is provided to the encoder, then encoder issues grant for a particular master to whom response needs to be transferred. In Demultiplexer there are 4 signals is coming which on the basis of sel signal chooses master. In demultiplexer 4 bit Id is coming from Id decoder/separator which tell that for which transaction response has been given.

Block Diagram of Read Data Channel

The read data channel conveys both the read data and any read response information from the slave back to the master. The read data channel includes:

- The data bus that can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits wide.
- A read response indicating the completion status of the read transaction.

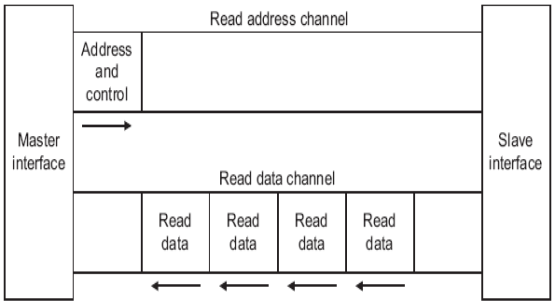

Figure 6: Read address and data channel

III. Simulation And Results

Simulation is carried out on Modelsim 6.5e simulator and syntheses is done using Xilinx 13.1.

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

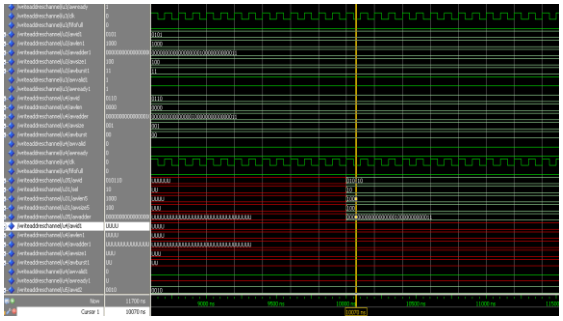

Figure 7: Simulation waveform for Write Address Channel

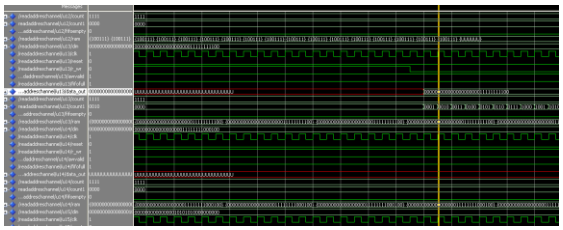

Figure 8: Simulation waveform for Master

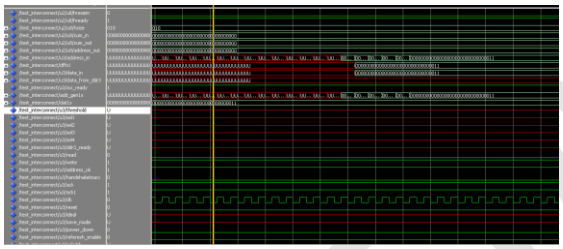

Figure 9: Simulation waveform for write operation with DDR3

Here in the simulation u3 component is master from where address needs to be transferred input signals of so address which was given to the master (u3) is "00000000000000000010000000000011" is and awid is "0101". Both of this awid and awaddress signals are being transferred to u31 component as shown in figure, hence address output signal of slave port is receiving the a awaddress signal as "00000000000000000010000000000011" and awid as "0101".

IV. CONCLUSION

We have implemented AXI 2.0 protocol which removes the limitation of communication architecture, which would otherwise reduce the speed of data transfer in System on chip. We have also implemented DDR3 controller which was then interface with AXI 2.0. Proposed approach was synthesized with Xilinx 13.1.

REFERENCE

[1] "AMBA AXI Protocol specification". www.arm.com/armtech/AXI

[2] Vijaykumar, R K Karunavathi, Vijay Prakash, "Design of Low Power Double Data Rate 3 Memory Controller with AXI compliant", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume 1, Issue 5, June 2012.

[3] Osborne, S., Erdogan, A.T. Arslan, T., Robinson, D., "Bus encoding architecture for low- power implementation of an AMBA-based SoC platform", IEEE Proceedings on Computers and Digital Techniques, Vol. 149, Issue 4, July 2002.

[4] Fu-ming Xiao, Dong-sheng Li, Gao-Ming Du, Yu-kun Song, "Design of AXI bus based MPSoC on FPGA", 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication (ASID), 2009.

[5] Terry Tao Ye, LUCA BENINI, Giovanni De Micheli, "Packetized On-Chip Interconnect Communication Analysis for MPSoC," Proceeding of the DATE, Messe Munich, Germany, pp. 344-349, 2003.

[6] Sudeep Pasricha, Nikil Dutt, "On-Chip Communication Architectures: System on Chip Interconnect", Morgan Kaufmann, 2010.