



---

## VHDL Design and Implementation of High Speed Double Data Rate 3 Memory Controller with AXI 2.0 compliant

Sharad Singh Parihar<sup>1</sup>, Prof. Sandip Nemade<sup>2</sup>

<sup>1</sup>M.Tech. Student (VLSI Design) Department of Electronics and Communication Engineering,  
Technocrats Institute of Technology, Bhopal

<sup>2</sup>Associate Professor, Department of Electronics and Communication Engineering,  
Technocrats Institute of Technology, Bhopal

**Abstract-** *The AXI protocol is burst-based. Every transaction has addressed 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. In write transactions, in which all the data flows from the master to the slave, the AXI protocol has an additional write response channel to allow the slave to signal to the master the completion of the write transaction. The AXI protocol enables Address information to be issued ahead of the actual data transfer support for multiple outstanding transactions Support for out-of-order completion of transactions. Here we are implemented AXI 2.0 protocol our goal was to high speed data transfer we have tested our results with Xilinx 8.1 version our synthesis results shows that we achieved the target of increasing the speed from previous method our maximum speed is 180-190 MHz.*

**Keywords-** AMBA, Multi-Processor System, AXI, Direct Memory Access, MPSOC, flexibility.

### Introduction

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 “Advanced High Performance Bus (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 Multi-Processor System On Chip 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, Advanced Peripheral Bus (APB) interfaces. In our method we have made our protocol for only for 4 master to communicate 4 slave. Now as we are using 4 arbitor, 4 multiplexer to support the pipelining, so all master can communicate at a time to different slave so achieving the highest performance at a system level requires that all of the AXI masters and slaves are designed to a high performance target.

In our method we have made our protocol for only for 4 master to communicate 4 slave. Now as we are using 4 arbitor, 4 multiplexer to support the pipelining, so all master can communicate at a time to different slave so



achieving the highest performance at a system level requires that all of the AXI masters and slaves are designed to a high performance target.

In AXI interconnect write operation needs to be responded, this response not given for single transfer. Response operation is started when completion signal occurs for a burst.

The completion signal occurs only after completion of single burst not for single transfer that means after completion of every burst completion a signal is generated this response is given through response channel. Here in this block diagram 1, 2,3and4 are respectively bvalid, bid1, bvalid and bready and 5, 6, 7 and 8 are bvalid1, bid1, bvalid and bready1. In this protocol,4 type of responses are as like OK Exclusive OK,SLERR, DECERR. The operation starts as completion signal is generated, which leads to generation of b valid signal by slave. In response to the assertion of bvalid signal, bready signal also gets high. Slave 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 de-multiplexer there are 4 signals is coming which on the basis of sel signal chooses master. In de-multiplexer 4 bit Id is coming from id decoder/separator which tell that for which transaction response has been given.

## II. Literature Review

Previous method [1] describes the implementation of AXI compliant Dual Drive RAM (DDR3) memory controller. It discusses the overall architecture of the DDR3 controller along with the detailed design and operation of its individual sub blocks, the pipelining implemented in the design to increase the design throughput. It has discusses the advantage of DDR3 memories over DDR2 memories and the AXI protocol operation it has also included combining and reordering the “read/write” commands. For attaining the maximum throughput from the memory, it operates all the memory banks in parallel and minimizes the effect of pre charge/refresh and other DDR3 internal operation [1] before these paper describes the SOC platform and the bus encoding architecture [1].

In [3] the entire protocol described as data sheet of AXI 2.0 protocol which tells that each of the five independent channels consists of a set of information signals and uses a two-way VALID and READY handshake mechanism. The information source uses the VALID signal to show when valid data or control information is available on the channel. The destination uses the READY signal to show when it can accept the data. Both the read data channel and the write data channel also include a LAST signal to indicate when the transfer of the final data item within a transaction takes place. Read and write transactions each have their own address channel. The appropriate address channel carries all of the required address and control information for a transaction. The AXI protocol supports the following mechanisms like variable-length bursts, from 1 to 16 data transfers per burst, bursts with a transfer size of 8-1024 bits, wrapping, incrementing, and non-incrementing bursts.

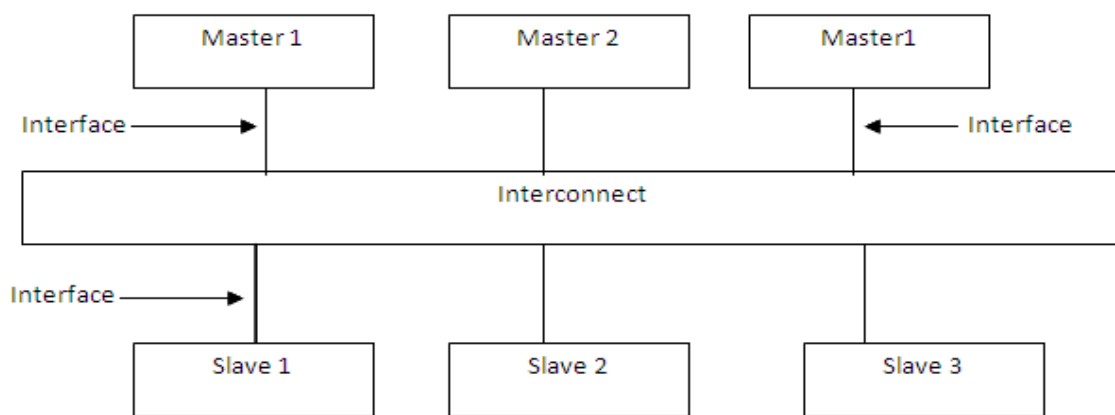
[3] SOC designs are increasingly becoming more complex. Efficient on-chip communication architectures are critical for achieving desired performance in these systems. System designers typically use Bus Cycle Accurate (BCA) models written in high level languages such as C/C++ to explore the communication design space. These models capture all of the bus signals and strictly maintain cycle accuracy, which is useful for reliable performance exploration but results in slow simulation speeds for complex designs, even when they are modeled using high level languages. Recently there have been several efforts to use the Transaction Level Modeling (TLM) paradigm for improving simulation performance in BCA models. However these BCA models capture a lot of details that can be eliminated when exploring communication architectures. In this paper we extend the TLM approach and propose a new and faster transaction-based modeling abstraction level



(CCATB) to explore the communication design space. Our abstraction level bridges the gap between the TLM and BCA levels, and yields an average performance speedup of 55% over BCA models. We demonstrate how fast and accurate exploration of tradeoffs is possible for high-performance shared bus architectures such as AMBA 2.0 and AMBA 3.0 (AXI) in industrial strength designs at the proposed abstraction level.

### III. Methodology Used

A typical system consists of a number of master and slave devices connected together through some form of interconnect, as shown in Figure 1.



**Fig. 1:** Interconnect.

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

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

In AXI interconnect write operation needs to be responded, this response not given for single transfer. Response operation is started when completion signal occurs for a burst.

The completion signal occurs only after completion of single burst not for single transfer that means after completion of every burst completion a signal is generated this response is given through response channel. Here in this block diagram 1,2,3and4 are respectively bvalid,bid1,bvalid and bready and 5,6,7 and 8 are bvalid1, bid1, bvalid and bready1. In this protocol,4 type of responses are as like OK Exclusive OK,SLERR, DECERR .

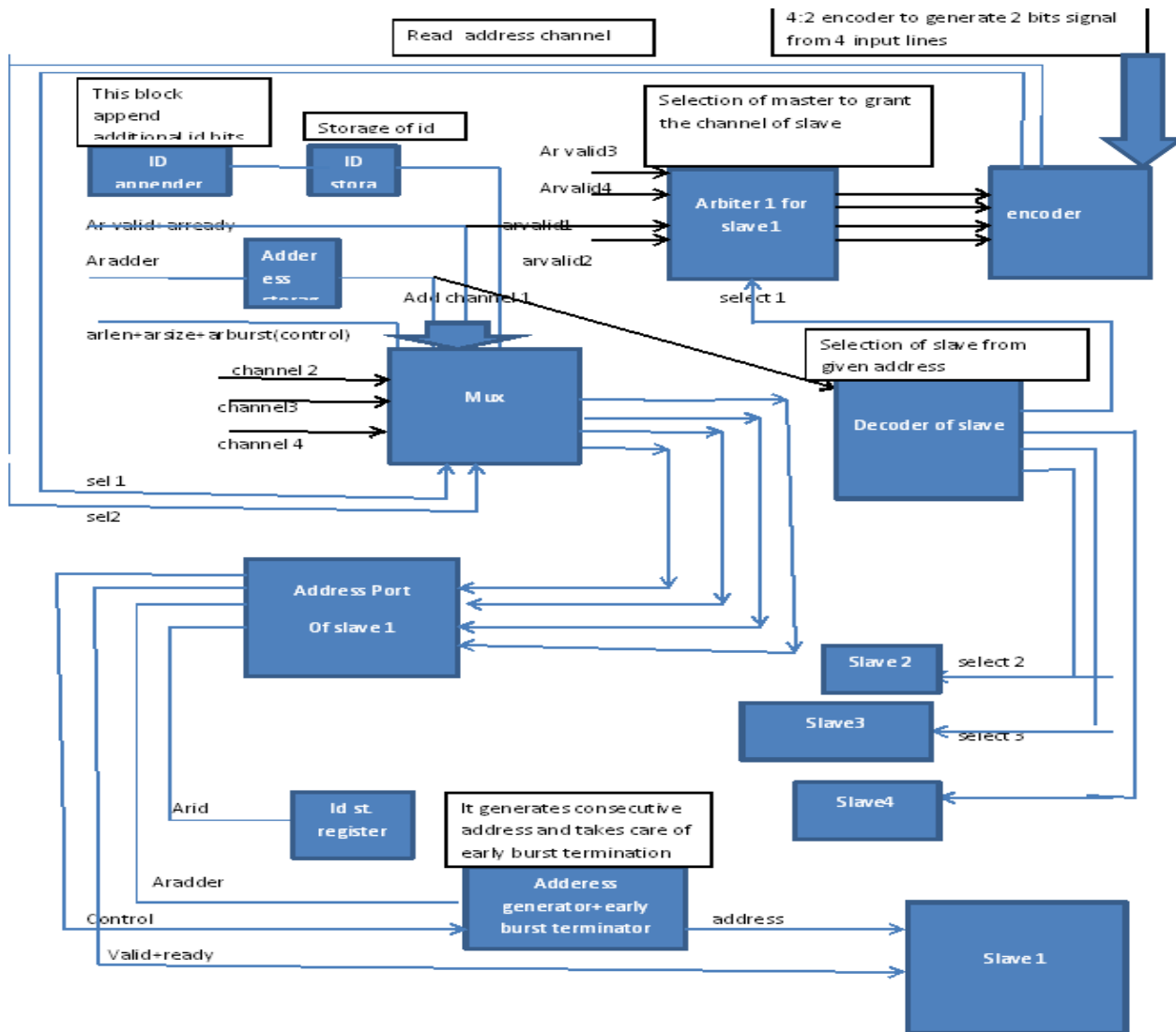


Fig. 2: Block Diagram Read Address Channel.

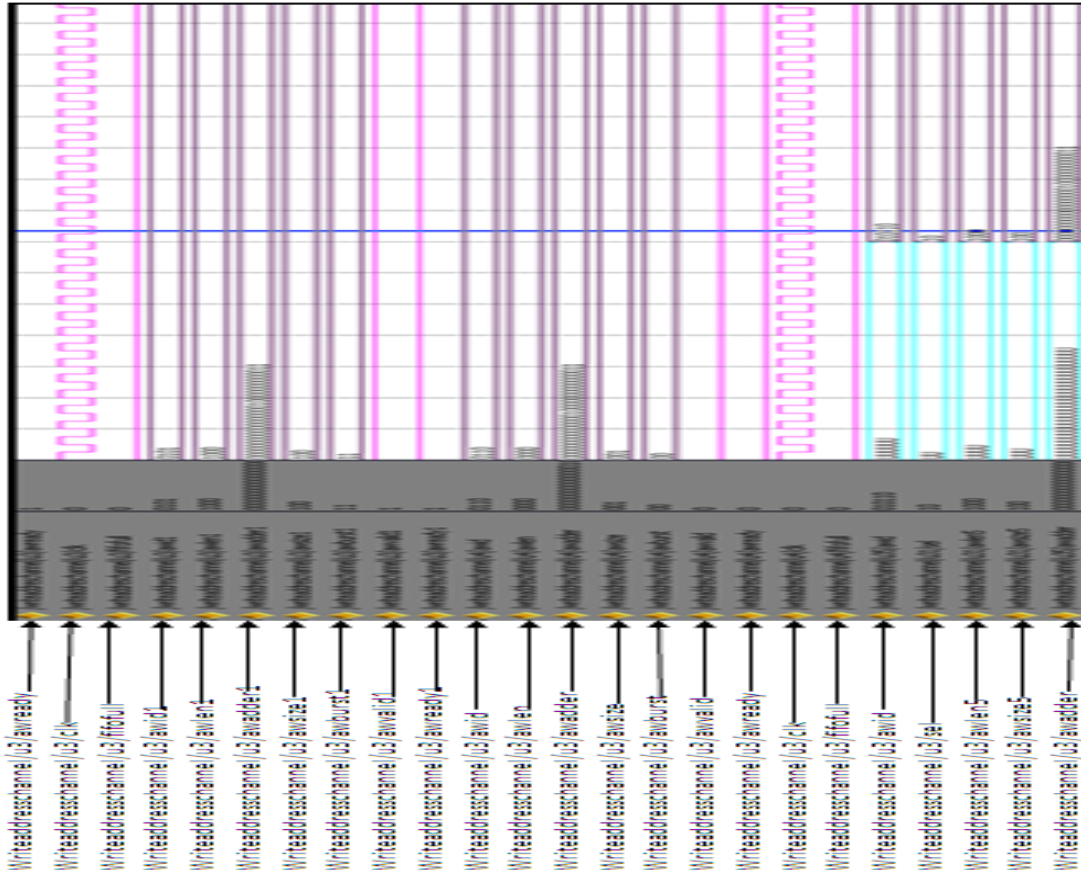
#### IV. Implementation

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 “000000000000000001000000000011” is and awid is “0101”. Both of this awid and aaddress signal is being transferred to U31 component as shown in fig. Hence address,output signal of slave port is receiving the a aaddress signal as “000000000000000001000000000011” and awidas “0101” as shown in Fig6.1 Write Address Channel. If bid signal is ‘0010’ and bresp signal is ‘01’ then on condition of bvalid=1, response signal (bid1, bresp1, bvalid1 and bready1) are transferred to the output.

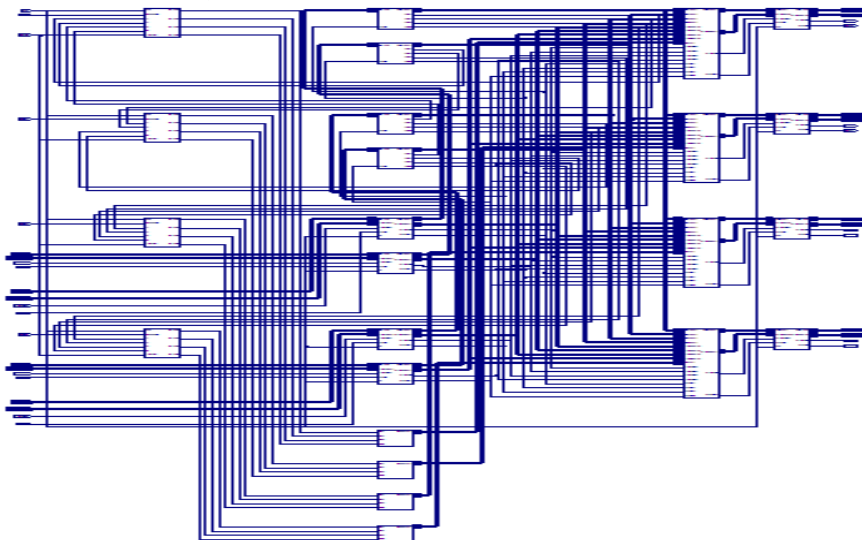
Here the simulation of response encoder has been shown. If “sel1” signal is getting equal to one then the output of encoder, “gun” will be equal to “00” “sel2” signal is getting equal to one then the output of encoder, “gun” will be equal to “01” “sel3” signal is getting equal to one then the output of encoder, “gun” will be



equal to “10” “sel4” signal is getting equal to one then the output of encoder, “gun” will be equal to “11” as shown in Fig 6.5 simulation of Response-Encoder



**Fig.3:** Simulation of Write Address Channel.



**Fig.4:** RTL view of Response Channel.



### V. Conclusion

In our scheme we have implemented with AXI 2.0 protocol is 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. we have tested our project with Xilinx 8.1 on (name of package) our synthesis result shows that our speed, still, is better as compared to previous method which is 190.500MHz.

In our scheme we have implemented with AXI 2.0 protocol is which removes the limitation of communication architecture, which would otherwise reduce the speed of data transfer in SOC we have also implemented DDR3 controller which was then interface with AXI 2.0 protocol. We have tested our project with Xilinx 8.1 on our synthesis result shows that our speed, still, is better as compared to previous method which is 190.500MHz. We have completed our work with keeping this intention in mind that speed should be optimized. In our work we have kept everything distributed to achieve perfect parallelism like we have implemented 4 multiplexers, 4 arbitors, 4 encoders and 4 decoders hence no any master have to wait for any of this devices to be free Hence all master would be able to transfer the data simultaneously. In previous method the latency was 10 to 35 clock we were expecting it reduce and we have reduce up 16 clock and our maximum speed is 190 MHz in our scheme.

Timing Summary:

Minimum period: 5.249ns Maximum Frequency: 190.500MHz

Minimum input arrival time before clock: 4.328ns

Maximum output required time after clock: 3.921ns

Maximum combinational path delay: No path found

### REFERENCES

- [1] Fu-ming Xiao, Dong-sheng Li, Gao-ming Du, Yu-kun Song, Duo-li Zhang, Ming-lunGao, "Design of AXI Bus Based MPSoC on FPGA" Anhui Provincial Natural Science Foundation grant #090412029 and National High Technology Research and Development Program of China grant #2008AA1Z135 Embedded Bus Architecture Report Presented by the Bus Architecture TSC.
- [2] XinpingZhu ,Sharad Malik, "A hierarchical modeling framework for on-chip communication architectures", IEEE/ACM international conference on Computer-aided design, pp.663-671, November 10-14-2012.
- [3] Terry Tao Ye, LUCA BENINI, Giovanni De Micheli, "Packetized On-Chip Interconnect Communication Analysis for MPSoC", Messe Munich, Germany, pp. 344-349, 2013. AMBA AXI Protocol specification. [www.arm.com/armtech/AXI](http://www.arm.com/armtech/AXI).
- [4] Jiang Zhou-liang, Quan Jin-guo, Lin Xiao-kang, "Analysis and Application of New Generation AMBA 3 AXI Protocol."
- [5] N. Genko , D. Atienza , G. De Micheli , J. M. Mendias , R. Hermida , F. Catthoor, "A Complete Network-On-Chip Emulation Framework" Proceedings of the conference on Design, Automation and Test in Europe, p.246-251, March 07-11, 2015.
- [6] W. Zhang et al., "Design of a Hierarchy-Bus Based MPSoC on FPGA" International Conference on Solid-state and Integrated Circuit Technology, ICSICT'06, pp. 1966-1968, 2016.



[7] GengLuo-Feng et al., "Design and Performance Evaluation of a 2D- Mesh Network on Chip Prototype Using FPGA" ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEM, APCCAS2008, pp.1264-1267, 2018.

[8] San jose "StratixII Device Handbook". Altera Copration, 2017,pp. 1-768.