IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Design a Bus Bridge between OCP and AHB

[1]Ashish Prajapati, [2]Dr. Alpesh H. Dafda

[1]PG student, [2]Assistant Professor,
[1]Signal Processing and VLSI Technology,
[1]Vishwakarma Govt. Engineering College, Chandkheda, Ahmadabad, India.

Abstract: This work proposes design of bridge for effective communication between protocols open OCP and AHB. There are various types of protocols are available which requires a bridge to pass the information from one type of protocol to other type of Protocol safely and without any data loss. Basically Bus Bridge should convert command and data of open core protocol (OCP) formats to acceptable Advance High performance Bus (AHB) formats. This conversion does not ensure proper communication unless the timings of each protocol were met. Hence the interconnecting Bus Bridge wrapper between OCP and AHB designed with proper timing delay. The functional verification and code coverage will be obtained. The design of bridge architecture done using verilog Hardware Descriptive Language (HDL). This design has been simulated and synthesized. In verification we have generated different test scenarios for Communication Bridge between protocols. We had done code coverage using code driven verification (CDV). The design will be simulated and synthesized using verilog HDL in xilinx.

Index Terms **- AHB, OCP, Bus Bridge, VLSI**

I. INTRODUCTION

Protocols are generally used today to connect IP blocks on structured SOCs. In present market, various types of protocols are available and are used in many applications, which require a bridge to pass data from one protocol to another one safely and without any loss. Basically, SOC is a system which is considered as a set of components and interconnects of them. In recent days, the development of SOC chips and the reusable IP cores [10] are given higher priority because of its less cost and reduction in the period of time to market [8]. The communication between the different IP cores should have a lossless dataflow and should be flexible to designer too. Hence to resolve this issue, the standard protocol buses are used in order to interface two IP cores. Most of IP cores use open core protocol due to its flexibility and functionalities to communicate between them [11]. Similarly ARM uses the AMBA (Advance Microcontroller Bus architecture) which has AHB (Advance High-performance Bus). This bus also has its own advantaged and flexibilities. Now, the data or signal flow between these protocols cannot be done easily because each protocol will be having its own properties and advantages. So in order to interface them, a bus bridge must be created which should have the properties to convert the signal of one form to another which is acceptable by receiving end protocol [5].

The Open Core Protocol (OCP) is a core centric protocol which defines a high-performance, bus-independent interface between IP cores that reduces design time, design risk, and manufacturing costs for SOC designs. The OCP defines a point-to-point interface between two communicating entities such as IP cores and bus interface modules. One entity acts as the master of the OCP instance, and the other as the slave. Only the master can present commands and is the controlling entity. The slave responds to commands presented to it, either by accepting data from the master, or presenting data to the master. The main reason of choosing OCP is that we can configure it as per our requirement. Also it is open to the public. It supports some advance features like configurable sideband control signaling and test harness signals. OCP provides independence from the bus protocols without having sacrifice high performance access to on chip interconnects.

The AHB (Advanced High-performance Bus) is a high-performance bus in AMBA (Advanced Microcontroller Bus Architecture) family. This AHB can be used in high clock frequency system modules. The AHB acts as the high-performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro cell functions.

Basically AHB comprises of four components- Arbiter, Master, Slave, and Decoder. The arbitration mechanism is used to ensure that only one master has access to the bus at any one time. The arbiter performs this function by observing a number of different requests to use the bus and deciding which is currently the highest priority master requesting the bus. A bus master is able to initiate read and write information by providing address and control information. Only one bus master can use the bus at the same time No provision is made within the AHB specification for a bus master to cancel a transfer once it has commenced. After a master has started a transfer, the slave then determines how the transfer should progress. The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer.

II. RELATED WORK

In recent years, the wrapper is designed in two different ways. From that one is automated synthesis as[4], it proposed an algorithm which used two different protocol's FSM to synthesis the corresponding wrapper. In 2011, Ramesh Bhakthavatchalu, Deepthy G R, Vidhya S, Nisha V proposed low power OCP bridge interface[9], drawback of this is it has increased complexity. In 2010, Chin-Yao Chang, Yi-Jiun Chang, Kuen-Jong Lee, Jen-Chieh Yeh, Shih-Yin Lin and Jui-Liang Ma proposed a on chip bus design with OCP interface[5], drawback of that is it has less reusability. In 2008, Chih-Wea Wang, Chi-Shao Lai, Chi-Feng Wu, Shih-Arn Hwang, and Ying-His Lin proposed on-chip interconnection using OCP[8], again this has also low IP reusability issue. In 2005, Natale Barsotti, Riccardo Mariani, Matteo Martinelli, and Mario Pasquariello proposed dynamic verification of OCP based SOC[6], it inscribe verification of OCP based SOC. Also other work related to Bus Bridge and wrapper based bus implementation techniques for performance improvement of bridge. With some techniques we can improve performance of bridge and get better performance. For design of Bus Bridge between OCP and AHB, we have to study OCP protocol as well as AHB protocol. Also we taken some work related to these protocol in consideration for designing bridge between these two. Also consider its specification for Design Bridge between them and proper working of bridge.

III. LITERATURE WORK

The literature survey is carried out for AHB and OCP protocols individually and then compared with respect to their own functionality. Basic introduction of both protocols are given in introduction part of this paper above.

3.1 Open Core Protocol (OCP)

The block diagram explaining basic operation and characteristics of OCP is shown in figure 3.1. The OCP defines point-to-point interface between two communicating entities such as IP cores and bus interface modules.[1] One entity will work as master of OCP interface and other as slave. Only the master can present commands and is the controlling entity. The slave responds to commands presented to it, either by accepting data from the master, or presenting data to the master.

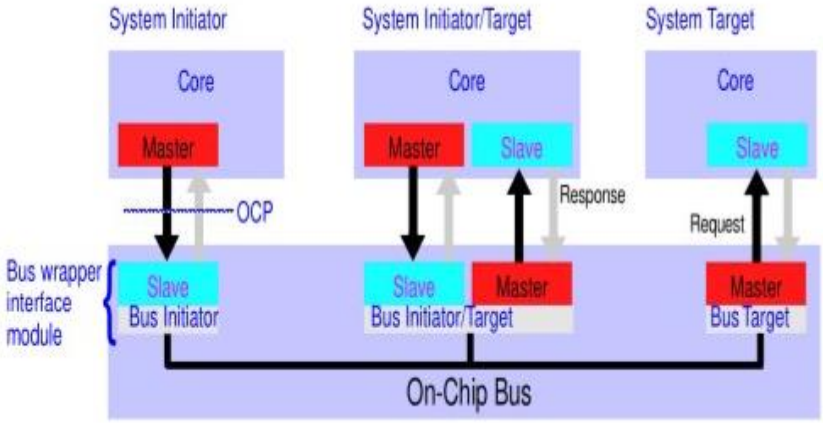

Figure 3.1: Basic block diagram of OCP instance [1]

Above figure shows a simple system containing a wrapped bus and three IP core entities such as one is system target, one is that system initiator, and an entity that is both. The characteristics of the IP core determine whether core needs master, slave or both sides of OCP and the wrapper interface modules must act as the complementary side of the OCP for each connected entity.

A system initiator (as the OCP master) presents command, control, and possibly data to its connected slave (a bus wrapper interface module). The interface module plays the request across the on-chip bus system. The OCP does not specify the embedded bus functionality. Instead, the interface designer converts the OCP request into an embedded bus transfer. The receiving bus wrapper interface module (as the OCP master) converts the embedded bus operation into a legal OCP command. The system target (OCP slave) receives the command and takes the requested action.

3.2 AMBA-AHB protocol

The AHB (Advanced High-performance Bus) is a high-performance bus in AMBA (Advanced Microcontroller Bus Architecture) family. Figure 3.2 shows AHB system design with one AHB master and three AHB slaves. The bus interconnect logic consists of one address decoder and a slave-to-master multiplexor.

Master - A bus master is able to initiate read and write information by providing address and control information. Only one bus master can use the bus at the same time An AHB bus master has the most complex bus interface in an AMBA system. Typically an AMBA system designer would use predesigned bus masters and therefore would not need to be concerned with the detail of the bus master interface. No provision is made within the AHB specification for a bus master to cancel a transfer once it has commenced.

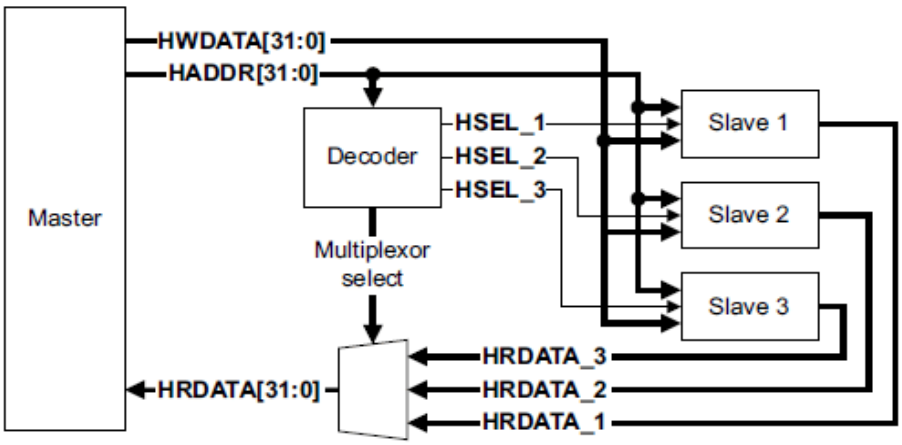

Figure 3.2: AMBA – AHB block diagram [2]

Slave - After a master has started a transfer, the slave then determines how the transfer should progress. Whenever a slave is accessed it must provide a response which indicates the status of the transfer. The HREADYout signal is used to extend the transfer and this works in combination with the response signal Hresp which provide the status of the transfer. The slave can complete the transfer in a number of ways. It can:

- Complete the transfer immediately
- Signal an error to indicate that the transfer has failed
- Delay the completion of the transfer, but allow the master and slave to back off the bus, leaving it available for other transfers.

Decoder - The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. The decoder monitors the address from the master so that the appropriate slave is selected and the multiplexor routes the corresponding slave output data back to the master.

Multiplexer - A slave-to-master multiplexor is required to multiplex the read data bus and response signals from the slaves to the master. The decoder provides control for the multiplexor. A single centralized multiplexor is required in all AHB implementations that use two or more slaves.

The signals involved in designing the AMBA AHB protocol are listed in table 1, which also gives the specification of each signal.

S. No.	NAME	WIDTH	DRIVER	FUNCTION
1	Hclk	1	Clock Source	This clock times all bus transfer at the rising edge of Hclk
2	Haddr	32	Master	The system address bus of width 32-bit
3	Htrans	2	Master	Indicates the type of the current transfer happening
4	Hwrite	1	Master	When HIGH this signal indicates a write transfer and when LOW a read transfer
5	Hsize	3	Master	Indicates the size of the transfer
6	Hburst	3	Master	Indicates if the transfer forms part of a burst.
7	Hwdata	32	Master	The write data bus is used to transfer data from the master to the bus slaves during write operations.
8	Hrdata	32	Slave	The read data bus is used to transfer data from bus slaves to the bus master during read operations.
9	HRADYout	1	Slave	When HIGH the HREADYout signal indicates that a transfer has finished on the bus. This signal may be driven LOW to extend a transfer.
10	Hresp	2	Slave	The transfer response provides additional information on the status of a transfer

TABLE 1: AMBA AHB signal specification

The table also includes the function of each signal and the source from which the each signal is driven. The width of each signal is specified and hence the address has 32 bits and both write and read data also has 32 bits. The operation is performed in a synchronized clock frequency and hence the signals should be changed with respect to the rising edge of the clock.

3.3 Bus Bridge

Basically a bus can be defined as a tool designed to interconnect the functional blocks in a systematic manner. It provides standardization in communication protocols between board-level devices. The Master and Slave AMBA AHB bus wrappers are used to connect bus independent Intellectual Properties (IPs) to an AMBA AHB bus controller. The bus bridge is the one which bridges any two protocols such as AHB and OCP in order to process the data from one block to the other through interconnection.

Based on the basic block diagram and the specifications of the OCP and AHB, designing a bus bridge between these two protocols are

made clear. In the bus bridge design, totally two versions of the wrapper are available, one for bridge level and another for top level.

Signal flow diagram: The Signal flow diagram of OCP-AHB Bridge & Top Level Wrapper is shown clearly in the Figure 3.3. From this figure, we can understand designing and overall operation of Bus Bridge. The OCP-AHB bus bridge design with a synchronous clock. On-chip Buses are the generally synchronous and in a synchronous bus all the operations are synchronized to a global bus clock.

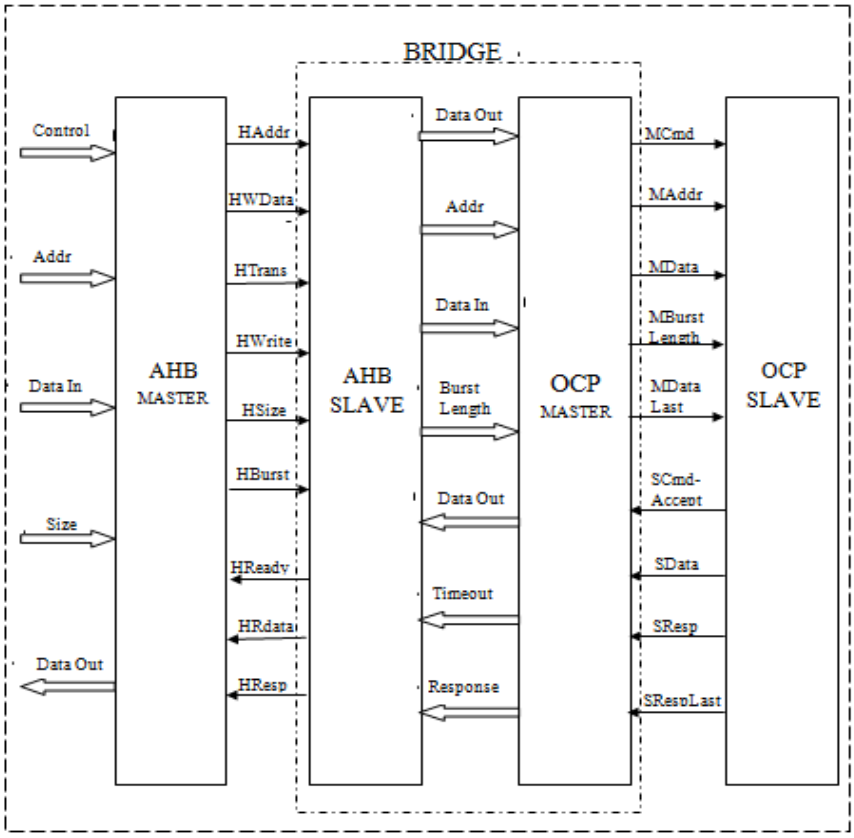

Figure 3.3: Signal flow diagram of OCP-AHB Bridge

Bridge – Basically the bridge contains the AHB slave and OCP master at the synchronized clock. So the AHB slave is fed by the AHB master and the output of the AHB slave is managed in order to feed those signals to the OCP master. The output signals of AHB slave are port mapped in such a way that the OCP master accepts it. Inside the bridge, the AHB slave and OCP master is synchronized by taking care of delay of each signals and the timing of control, address and data.

IV. WORKING METHODOLOGY

Bus Bridge between AHB and OCP has been design using verilog. We had used Xilinx tool for the purpose of designing of bridge using verilog. Here, we have design and developed bridge in Xilinx using RTL (Register Transfer Level). First of all we have design the bridge using RTL and then we had simulate that design in simulator. For simulate the result we have use ISim simulator.

Figure 4.1 showing the Xilinx tool, which we have use for design RTL of bridge. We have developed RTL design of the Bus Bridge. After that we have simulate that design using Isim simulator. In that simulator we can observe data read and write in form of waveforms.

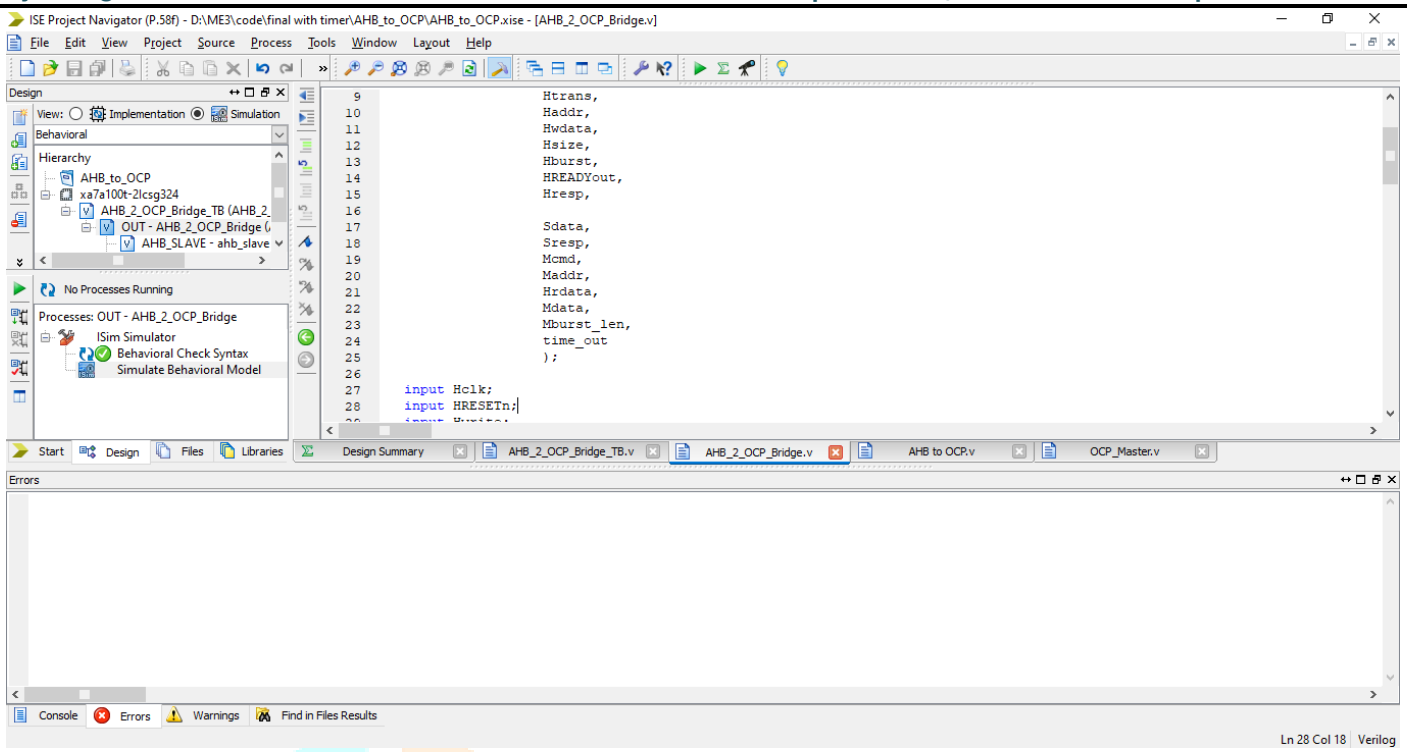

Figure 4.1: Coding for design of Bridge in Xilinx tool

V. SIMULATION RESULTS

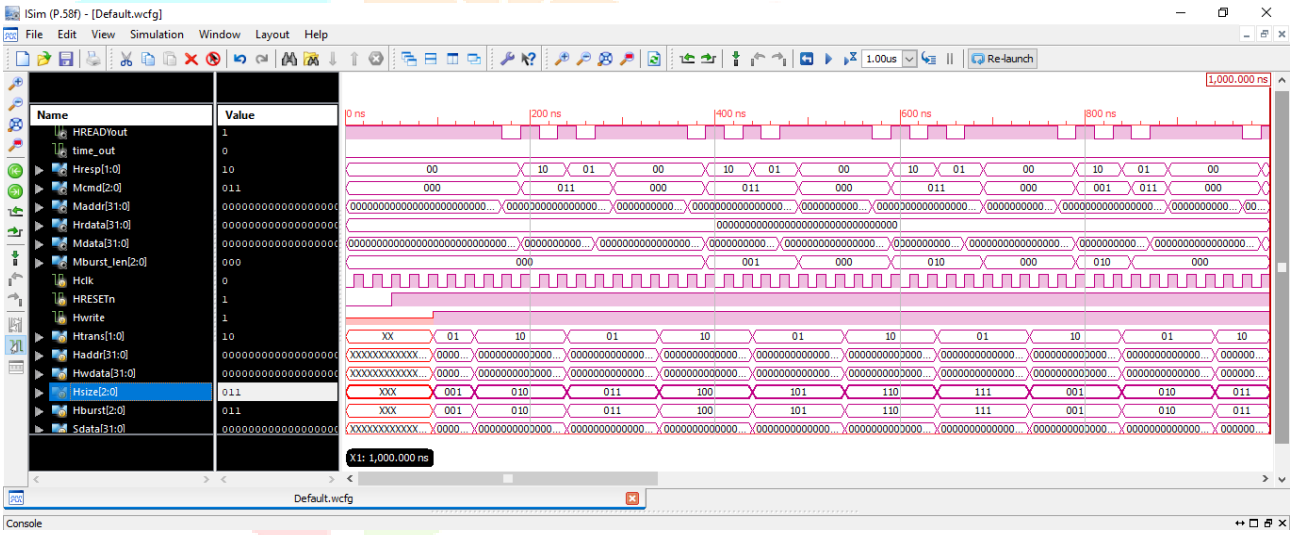

Figure 5.1: AHB – write operation

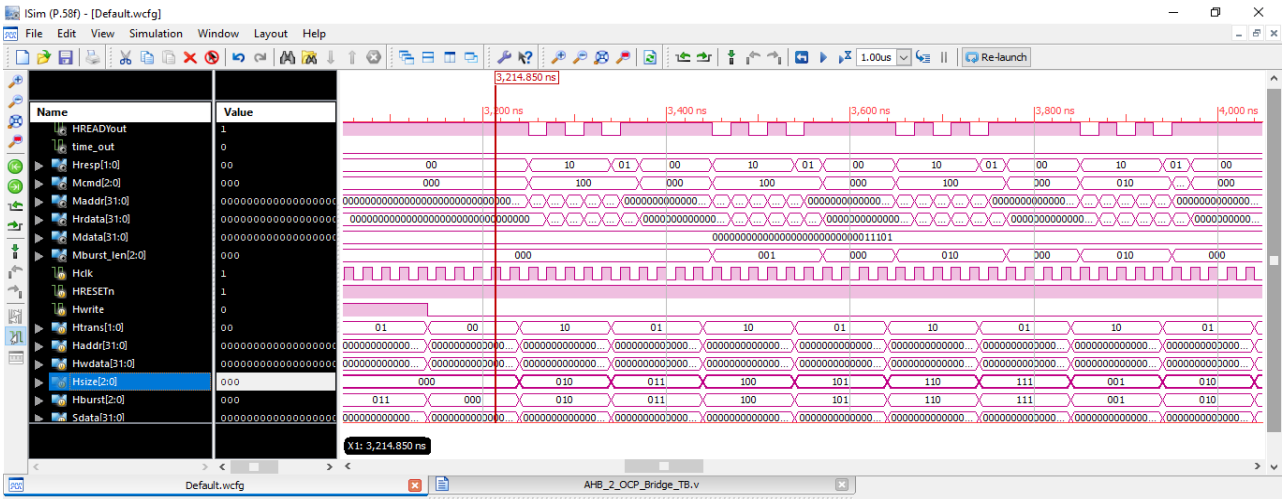

Figure 5.2: AHB – read operation

Here, we can see the simulation results of the AHB-OCP Bridge. Figure 5.1 showing AHB – write operation and figure 5.2 showing AHB – read operation. When Hwrite signal is high, AHB slave leads to write state which will provide address and data to OCP master. With this inputs OCP master will issue the Mcmd, Maddr and Mdata to OCP slave. After that OCP slave store the data in memory as per data given.

Similarly when Hwrite signal is low, AHB slave go to read state and issue address to OCP master. With this OCP master give Mcmd and Maddr to OCP slave. Now OCP slave go to the read state and fetch the data for the given location. Then OCP slave pass the data to OCP master and OCP master will send the data to AHB slave.

VI. CONCLUSION

The paper work represents the design of Bus Bridge between AHB and OCP, which acts as interface between these two different protocols. The simulations results show that read and write communication between these two protocols through bridge is proper. All commands and data are properly transferred from AHB to OCP by using bridge. We have verified the working of bridge by giving different and multiple data to the bridge and those data are successfully transferred.

REFERENCES

[1] OCP-IP, "Open Core Protocol Specification", Available at http://www.ocpip.org/

[2] ARM, "AMBA-AHB specification", http://www.arm.com/

[3] G. Geethaa Reddy, K. Vanisree, David Solomon Raju, "Implementation of Bus Bridge between AHB and OCP" Global Journal of Advanced Engineering Technologies, Vol1, Issue4, 2012

[4] V. Dapos silva, S. Ramesh and A. Sowmya, "Bridge over troubled wrappers: automated interface synthesis", Proceedings of 17th International Conference on VLSI Design (VLSID '04), pp. 189 – 194, 2004.

[5] Chin-Yao Chang, Yi-Jiun Chang, Kuen-Jong Lee, Jen-Chieh Yeh, Shih-Yin Lin and Jui-Liang Ma "Design of On-Chip Bus with OCP Int erface", IEEE, 2010

[6] Natale Barsotti, Riccardo Mariani, Matteo Martinelli, and Mario Pasquariello, "Dynamic verification of OCP-based SOC", in Proc. IEEE Int'l Conf. on System-on-Chip, Tampere, Finland, Nov. 2005, p. 22.

[7] Kenneth L. Calvert and Simon S. Lam, "Formal Methods for Protocol Conversion", IEEE Journal on Selected Areas in Communications, Vol. 10, pp. 127-142, No. I, Jan 1990.

[8] Chih-Wea Wang, Chi-Shao Lai, Chi-Feng Wu, Shih-Arn Hwang, and Ying-His Lin "On-chip Interconnection Design and SOC Integration with OCP", IEEE, 2008

[9] Ramesh Bhakthavatchalu, Deepthy G R, Vidhya S, Nisha V "Analysis of Low Power Open Core Protocol Bridge Interface Using VHDL.", IEEE, 2011

[10] Kenichiro Anjo, Atsushi Okamura and Masato Motomura, "Wrapper-Based Bus Implementation Techniques for Performance Improvement and Cost Reduction", 04 May 2004, IEEE Journal of Solid-State Circuits (JSSC)

[11] Cheng-Ta Wu, Feng-Xiang Huang, Kuan-Fu Kuo, Ing-Jer Huang "An OCP-AHB Bus Wrapper with Built-in ICE Support for SOC Integration", IEEE, 2012

[12] Chin-Yao Chang, Yi-Jiun Chang, Kuen-Jong Lee, Jen-Chieh Yeh, Shih-Yin Lin and Jui-Liang Ma, "Design of On-Chip Bus with OCP Interface", 28 Jun 2010, VLSI Design Automation and Test (VLSI-DAT)

[13] G. Shiny & S. Hanumanth Rao, "Design of on-chip bus OCP protocol with bus functionalities."International Journal of Power System Operation and Energy Management ISSN (PRINT): 2231 – 4407, Volume-2, Issue-3,4

[14] http://www.google.com/