



# Journal of Artificial Intelligence

ISSN 1994-5450

**science**  
alert

**ANSI***net*  
an open access publisher  
<http://ansinet.com>

## **Design of Multiple Master ASIC Interconnect**

**G. Suganya and R. Muthaiah**

School of Computing, SASTRA University, Thanjavur-613402, Tamil Nadu, India

*Corresponding Author: G. Suganya, School of Computing, SASTRA University, Thanjavur-613402, Tamil Nadu, India  
Tel: +91 9880433272*

### **ABSTRACT**

System on Chip is composed of many Intellectual Property (IP) blocks which communicate with each other using different bus protocols. Each IP may have different native protocol. So establishing communication between these bus protocols which vary from one IP to another is of much significance. The protocols used may be either proprietary or non-proprietary. AXI 1.0 and OCP 2.0 protocols and are used to build our ASIC interconnect system. Here we establish communication between these two protocols using an AXI to OCP converter. This converter acts as a bridge between the two protocols clearing the incompatibilities for communication. We also use a clock bridge to enhance communication between IP blocks that operate at different frequencies. The numbers of masters, slaves, address and data widths are also configurable depending on the requirement. The AXI master, AXI to OCP converter, Clock Bridge, OCP master, OCP slaves, arbiter and address decoder are designed using VHDL and synthesized.

**Key words:** OCP protocol, AXI protocol, IP Core, SoC design, VHDL language

### **INTRODUCTION**

Intellectual properties are of much significance in complex SoC designs as they tend to have a huge impact on the complexity, design time and cost. Interfacing these IP cores are of much concern in these days as they play a vital role in achieving lossless and efficient communication. In order to achieve lossless communication two protocols are used. First one is Advanced Extensible Interface protocol (AXI). It has five independent channels (write and read address, write and read data, write response channels). All channels use certain signals for transaction. Since, there are two channels for write and read, the transactions are processed in parallel (ARM, 2011). As a result of this there is an additional area overhead. The second one is Open Core Protocol-International Partnership (OCP-IP) (<http://www.ocpip.org/>). OCP aims in bringing standardization to simplify the problems of integration. It achieves this by the use of a set of interface signals that are independent of the underlying architecture (Karandikar, 2010). Hence designers can focus on the functional aspects of the design, bus designers on the architecture of bus and system integrators on the whole system architecture (Devadas and Latha, 2011). Hence, system integration becomes more efficient.

### **AXI PROTOCOL**

The Advanced Extensible Interface (AXI) protocol has some unique features. It has separate read and write channels thereby providing quicker operations for low latency slaves and also enables Direct Memory Access. Also it supports separate control, address and data states to make transactions more specific. This protocol is burst-based. All transactions have address and

corresponding control command on the address channel that specifies the nature of transaction that has to take place. The data is transferred from master to slave through write channel and slave to master through read channel. There is an additional write response channel from the slave to master to specify it has completed the intended transaction.

### **OPEN CORE PROTOCOL INTERNATIONAL PARTNERSHIP (OCP-IP)**

The OCP establishes a peer to peer interface between two IP blocks that intend to communicate with each other (Wang *et al.*, 2008). Here one core acts as the master and the other as slave. The master is the controlling entity and issues command and data. The slave responds to the commands presented to it, either by accepting data or presenting data to the master (Zhang *et al.*, 2009). In interconnect; we performed read, write and write non-post operations. In addition, the proposed interconnect is configurable such that we can connect many masters and slaves depending upon the requirement. In addition, parameters like address width and data width are configurable.

### **HARDWARE DESIGN OF THE ASIC INTERCONNECT**

The blocks used in the interconnect are AXI master, AXI-OCP converter, OCP master, OCP arbiter, OCP address decoder and OCP slave.

**AXI master:** AXI protocol specifies AXI master as a component which controls the operations such as, read and writes from and to slave. For the Master, the Slave is a location where it can write or reads data. The master is designed as a FSM with read and writes operations. The FSM has two sub FSMs with read and write happening independently due to separate read and write channels. Each operation is initiated with the asserting of the address signal to the slave with its validity. The master waits to receive the response from the slave to proceed further. After getting the response from the slave the master sends the data to be written to the specific address in the slave or reads data from the specific address in the slave (Weber, 2000). The read data channel conveys the data read and response back to the master. The ASIC interconnect we designed is shown in Fig. 1.

**AXI-OCP converter:** The converter involves mapping the inputs and outputs of AXI Master to OCP specific protocols. All OCP signals are sampled by the positive edge of the clock. Except for clock, all other signals are unidirectional. The commands from the AXI master are input to the AXI-OCP converter which is converted into OCP protocol and forwarded to slave. The signals coming out from OCP converter are OCP protocol specific which can be directly connected to OCP slave interface (Bhakthavatchalu and Deepthy, 2011). The response signals coming from OCP slave interface is input to the AXI-OCP converter which is returned to AXI master in AXI standards.

**Clock bridge:** The Bus masters and slaves can operate at different clock frequencies. Clock bridges are added to achieve synchronization.

**OCP master:** The OCP master is designed as a FSM with simple read and writes operations. OCP Master simultaneously generates MCmd, MAddr and MData unlike AXI Master which waits for response from slave before it issues MData (Chang *et al.*, 2010).

**OCP arbiter:** Arbiter receives request signals from all the masters. It is implemented using a ring counter to select the masters (Shin *et al.*, 2002). The ring counter design is generic such that it is

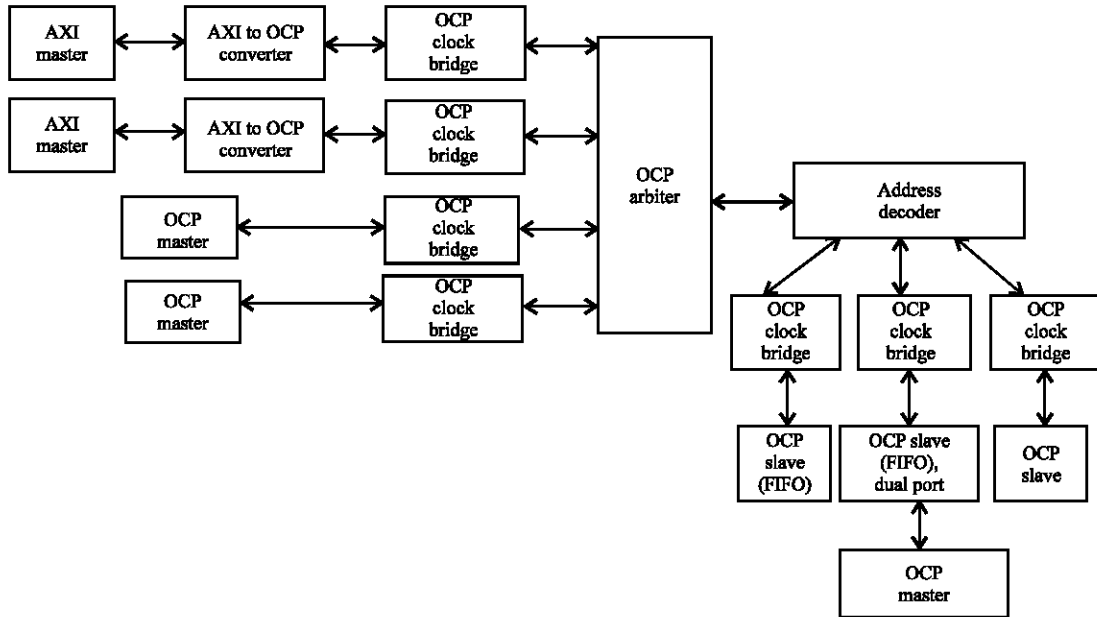


Fig. 1: ASIC interconnect design

possible to connect a configurable number of masters. If the selected master is at idle state it goes to next master else it completes read, write or write non-post operations and then goes to next master. This is done sequentially so each master has to wait for its turn.

**OCP address decoder:** The output of the Arbiter is given to an address decoder. This is used for selecting one slave from a group of slaves. This address decoder design is also generic such that it is possible to connect a configurable number of slaves. The Master that is chosen by the arbiter can communicate with the slave to which it intends. The MAddr signal contains a Slave ID (SID). The first few MSB bits are reserved for the selection of slave. The ID's are stored in a memory. The Address decoder checks for the SID with which the master intends to communicate and establishes a connection between that master and the slave.

**OCP slaves:** Three OCP slaves are designed. First one is a simple memory where master can perform read, write operations. Second slave is a synchronous FIFO. Masters can perform read writes to a fixed address. The FIFO is implemented using a memory and write pointers and read pointers which auto increment on a write and read, respectively. Third slave is a dual port FIFO. It is possible for writes and reads to be issued simultaneously on the two ports. In case writes are issued simultaneously on both the slave ports, the slave performs arbitration between the two masters. Similarly, if in case reads are issued simultaneously on both the slave ports, the slave performs arbitration between the two masters.

## RESULTS

The simulation results are shown for five masters (2 AXI and 3 OCP) and three slaves after integration. The basic transactions are performed by all five masters with respect to slaves. The first slave is a simple memory followed by synchronous FIFO and dual port FIFO. Slave ID's are as follows 000,001 and 010. Table 1 shows MCmd input values and command values. Figure 2 shows

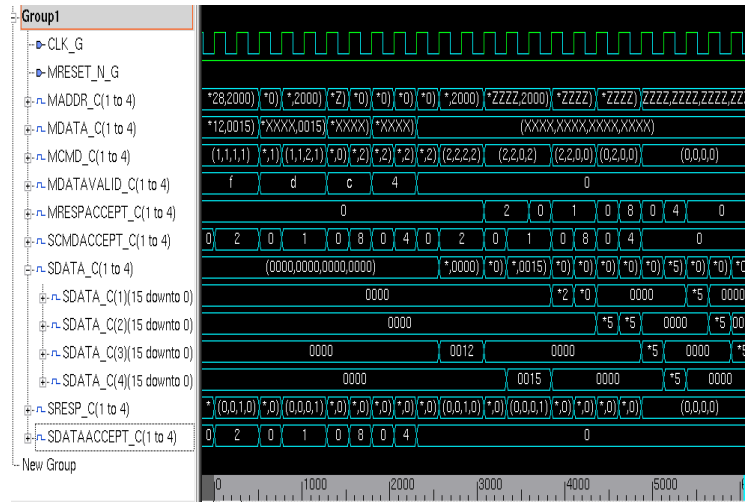


Fig. 2: Single read and write transactions

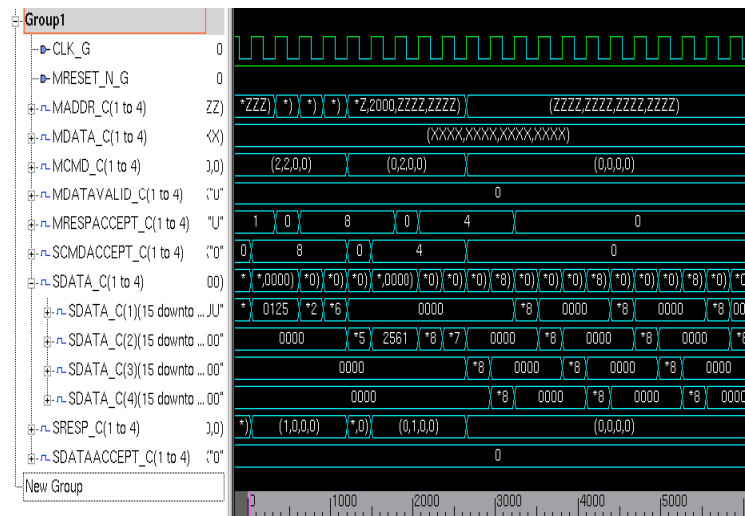


Fig. 3: Back to back read and write transactions

Table 1: MCcmd input value and its command

MCcmd	Command
000	Idle
001	Write
010	Read
101	Write non post

single read and write operation performed by all masters on the slaves. Initially master 1 is selected. If MCcmd is other than 000 the grant remains with the first master and the master performs all intended operations on the slaves.

The selected masters commands are given to the address decoder which selects the particular slave for the transaction depending upon the slave ID, indicated in the MAddr first 3 MSB bits.

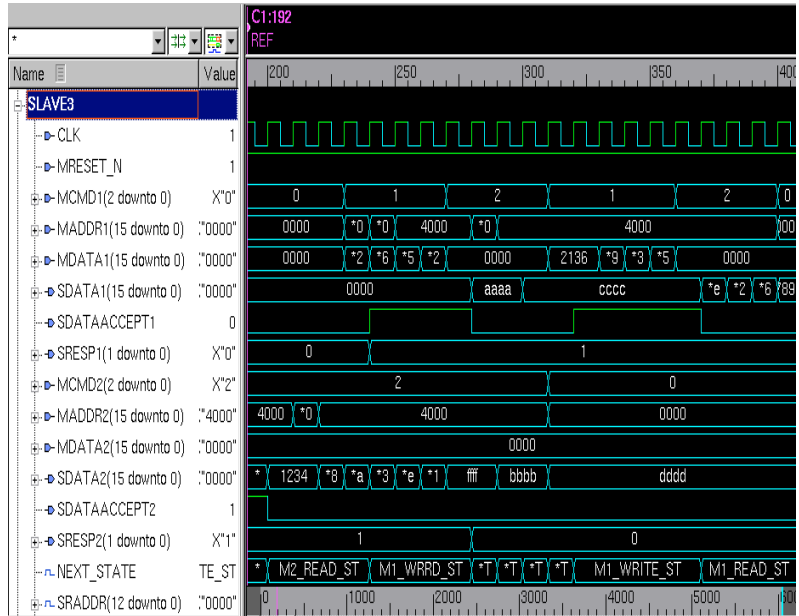


Fig. 4: Arbitration in dual port FIFO

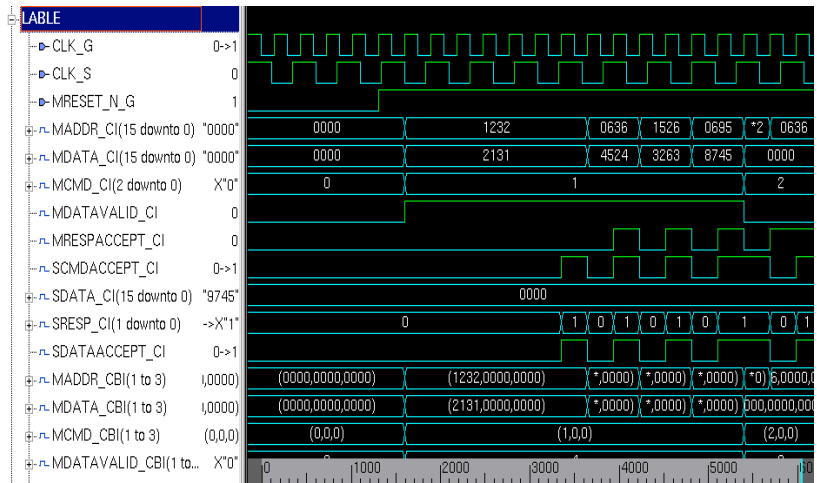


Fig. 5: Synchronization using clock bridge

Once MCmd becomes 000 the grant passes to the next master. In this way, all masters get access to the corresponding slaves. Figure 3 shows back to back write and read operations by all masters. All masters write to three different slaves and written data are read back. There is a fixed OCP master which communicates with dual port FIFO. Figure 4 shows the arbitration in the dual port FIFO. The other input of the FIFO is from the arbitration output of other 4 masters. When both masters wish to read/write simultaneously, the slave performs arbitration between the two masters. Figure 5 shows the synchronization achieved using a clock bridge where all masters and slaves operate at different frequencies.

## **CONCLUSION**

A multi-master interconnect system for ASIC presented in this paper is implemented for transactions like read, write and write non post. The number of masters and slaves in interconnect are configurable and can operate at different clock frequencies. The parameters like address and data width are also configurable. Thus, the multi master ASIC interconnects designed using OCP protocol reduces the time and complexity to design a new interface for different IP cores every time and provides a common platform for interfacing all IP cores that uses AXI and OCP protocols thereby achieving core reusability.

## **REFERENCES**

- ARM, 2011. AMBA AXI protocol v. 1.0 specification. ARM Limited. <http://www.arm.com/>
- Bhakthavatchalu, R. and G.R. Deepthy, 2011. Implementation of re-configurable open core protocol compliant memory system using VHDL. *Int. J. Comput. Sci. Issues*, 8: 416-421.
- Chang, C.Y., Y.J. Chang, K.J. Lee, J.C. Yeh, S.Y. Lin and J.L. Ma, 2010. Design of on-chip bus with OCP interface. *Proceedings of the IEEE International Symposium on VLSI Design Automation and Test*, April 26-29, 2010, Hsinchu, Taiwan, pp: 211-214.
- Devadas, M. and K. Latha, 2011. Design of high performance on-chip bus OCP-IP protocol with advanced bus functionalities. *Proceedings of the International Conference on Advanced Computing, Communication and Networks*, June 2, 2011, Chandigarh, India, pp: 924-929.
- Karandikar, P.D., 2010. Open Core Protocol (OCP): An introduction to interface specification. *Proceedings of the 1st Workshop on SoC Architecture, Accelerators and Workloads*, January 10, 2010, Bangalore, India.
- Shin, E.S., V.J. Mooney III and G.F. Riley, 2002. Round-robin arbiter design and generation. *Proceedings of the IEEE 15th International symposium on System Synthesis*, October 2-4, 2002, Kyoto, Japan, pp: 243-248.
- Wang, C.W., C.S. Lai, C.F. Wu, S.A. Hwang and Y.H. Lin, 2008. On-chip interconnection design and SoC integration with OCP. *Proceedings of the IEEE International Symposium on VLSI Design, Automation and Test*, April 23-25, 2008, Hsinchu, Taiwan, pp: 25-28.
- Weber, W.D., 2000. Enabling reuse via an IP core-centric communications protocol: Open Core Protocol™. *Proceedings of the IP 2000-the System-on-Chip Conference*, March 20-22, 2000, The Santa Clara, CA., USA., pp: 217-224.
- Zhang, S., A.I. Ahmed and O.A. Mohamed, 2009. A re-usable verification framework of Open Core Protocol (OCP). *Proceedings of the Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference*, June 28-July 1, 2009, Toulouse, France, pp: 1-4.