

Implementation of Reduced Power Open Core Protocol Compliant Memory System using VHDL

Ramesh Bhakthavatchalu¹, Deepthy G R²

¹ Dept. of ECE
Amrita Vishwa Vidyapeetham,
Amritapuri, Kollam-690525, Kerala, India

² Dept. of ECE
Amrita Vishwa Vidyapeetham,
Amritapuri, Kollam-690525, Kerala, India

Abstract

The design of a large scale System on Chip (SoC) is becoming challenging not only due to the complexity but also due to the use of a large amount of Intellectual Properties (IP). An interface standard for IP cores is becoming important for a successful SoC design. In a SoC the different IP cores are interfaced through different protocols. It increases the complexity of the design. Open Core Protocol (OCP) is an openly licensed core centric protocol intended to meet contemporary system level integration challenges. OCP promotes IP core reusability and reduces design time, design risk and manufacturing costs for SoC designs. OCP defines a highly configurable interface including data flow, control, verification and test signals required to describe an IP core's communication. This paper focuses on the design and implementation of a reconfigurable OCP compliant Master Slave interface for a memory system with burst support. The power reduction using Multivoltage design is the important feature of the paper. The proposed design was implemented in VHDL and the Synthesis is done using Synopsys ASIC synthesis tool Design Compiler.

Keywords: *Memory Controller, Memory, Master, Slave, OCP compliant, Interface, Wrapper, Power Analysis, Burst Transfer, Power reduction, Multi voltage Design.*

1. Introduction

Open Core Protocol is a signal exchange protocol over a family of on-chip core interfaces. OCP data transfer models range from simple request-grant handshaking through pipelined request-response to complex out-of-order operations.

The OCP defines a point-to-point interface between two communicating entities, such as IP cores and bus interface modules (bus wrappers) [1]. Given the wide range of IP core functionality, performance and interface requirements, a fixed definition interface protocol cannot address the full

spectrum of system interface requirements. The need to support verification and test requirements adds an even higher level of complexity to the interface. To address this spectrum of interface definitions, the OCP defines a highly configurable interface. The OCP's structured methodology includes all of the signals required to describe an IP core's communications including data flow, control, and verification and test signals [2]. Since OCP is a core-specific, peer-to-peer protocol, OCP compliance IP cores can be verified independently with a Universal OCP monitor with OCP compliance assertions attached to OCP interface. In fact, to develop OCP compliance assertions, all possible aspects of features of the OCP protocol have to be integrated. OCP provides a master/slave connection between two cores. One core, the OCP initiator core has an OCP master interface. A master interface enables a core to generate OCP requests such as READ or WRITE and receive the READ responses. The other core, called the OCP target core, has an OCP slave interface which allows it to receive and respond to requests.

To simplify timing analysis, physical design, and general comprehension, the OCP is composed of uni-directional signals driven with respect to, and sampled by, the rising edge of the OCP clock. The OCP is fully synchronous and contains no multi-cycle timing paths with respect to the OCP clock. All signals other than the clock signal are strictly point-to-point. The OCP supports a configurable data width to allow multiple bytes to be transferred simultaneously. The OCP refers to the chosen data field width as the word size of the OCP.

2. Open Core Protocol Compliant System

The availability of a common interface platform provided by OCP has inspired system designers to use them as replacements for other interface protocols. OCP compliance is obtained by creating a wrapper around the original designs to meet the OCP specification. A wrapper is a design which satisfies all the specifications given by the Open Core Protocol. Then the designs are interfaced. For Compliance the core must include at least one OCP interface. The core and OCP interfaces must be described using an RTL configuration file. Each OCP interface on the core must comply with all aspects of the OCP interface specification. There are three types of OCP Profiles (i) High Performance (HP) (ii) Generic Profile (GP) (iii) Peripheral Profile (PP) [2]. The PP only implements the simple read/write transfer without other OCP extensions. GP extends the PP with additional data handshake phase and burst extensions for generic device with block data transfer. HP extends the GP with OCP tag extensions to support out-of-order response. OCP specifies 3 major types of interfaces. (i) Bus Bridge Interface (ii) Processor Interface (iii) Memory Interface [3]. The Bus bridge interface includes an external bus like USB or AXI and the internal bus will be OCP. In the Processor interface the interface is between processors which include only the OCP master. The memory interface is for DRAM, SRAM etc. OCP has been adopted by the industry with good results [4]. There is a large number of IPs with OCP interfaces at the top level. These OCP interfaces are different in protocol features or signals to optimize the needs of IP cores. However, all of them follow the same OCP timing and validation rules, which simplify the cost in verification and implementation [2].

3. Basic OCP Interface

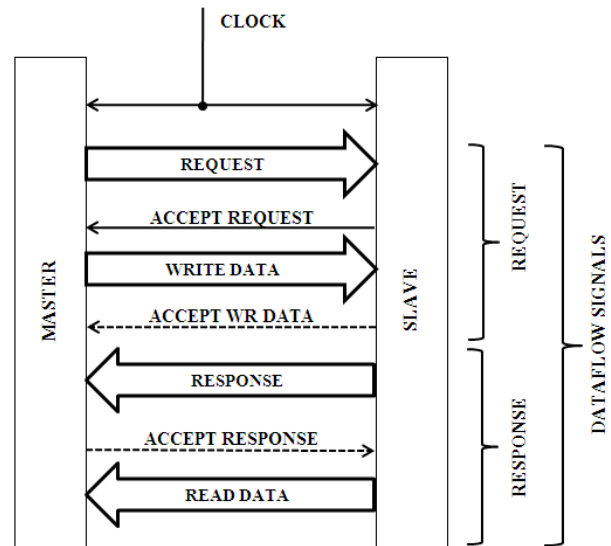


Fig. 1 Basic OCP Interface

Figure shows the basic OCP interface which consists of a Request and Response phase. The Master gives a Request to the slave, here a Write request and the Write data. The Slave accepts that request and gives a request accept signal back to the master. The slave then responds to the request and sends a response signal to the Master. Then according to the Master's request the data is read from the slave and is given to the master. The signals shown in dotted lines are optional. The basic OCP interface consists of dataflow signals only.

4. Memory Interface System

The scope of the research is that for understanding the OCP compliant system a model is needed since no such works exists in this field with experimental results. Hence a memory system with a Memory Controller as the OCP Master and Memory as OCP Slave is designed. First their performance is analyzed as the system itself is and then with the OCP wrapper.

The system includes an OCP compliant Memory Controller and a Memory where the memory controller acts as the OCP master and the Memory acts as the OCP Slave. This paper discusses the Peripheral OCP profile with Simple Write and Read transfer and Generic OCP profile with data handshaking and burst transfer [2].

The Master gives Requests and accepts responses. The slave receives and responds to the Requests provided by the master. Handshake signals are provided for both Master and Slave which indicates acknowledgements. The Memory designed can act as both program memory and

data memory and can be used as a memory system for any current day SoC design.

5. Implemented system

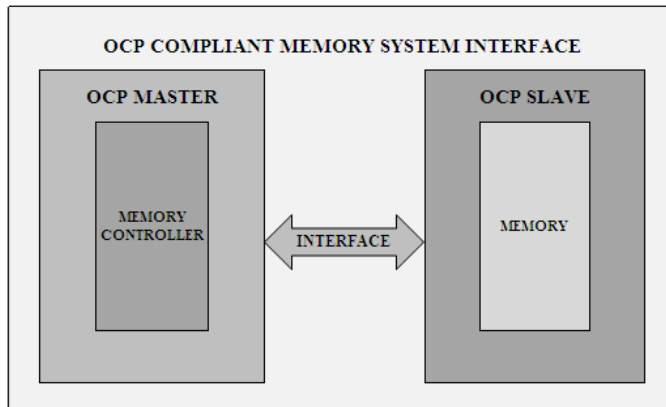


Fig. 2 Block Schematic of the Design Implemented

The proposed system is tested under 32-bit data bus and 8-bit address bus. The major control signals are Memwrite and Memread each of 1-bit length. Memory Controller gives Write and read requests as well as Address to the Memory. The Memory will respond to it by writing data into the memory address and reading data from the specified address and is given to the Memory controller. The slave responds with the response signals. During write operation the master starts a request phase by switching its command field to write and presents a valid data and address. The slave accepts the command and captures data and address and a write is performed according to the design. The master starts a Read request by switching its command field to Read. It presents a valid address and slave accepts the command. The slave captures data from the specified the address and is driven to the Master. The response is also given to master to indicate that the data is valid.

The inputs of the system are Address and Data which are in the form of Binary values. Another input is the 7-bit instruction which is given as the input of Memory Controller from which 2-bit opcode is extracted to determine the Memory operation (last 2 bits of the instruction).

There is no change in the designs of Memory Controller and Memory when OCP wrapper is introduced. The wrapper covers the existing designs to make them OCP compliant.

The proposed system is parameterizable for both address and data. However for the experiment the parameters are set as shown below.

Table 1: Design Parameters

Design Parameter	Size(in bits)
------------------	---------------

Address	8
Data	32
Instruction	8
Burst Length width	4

5.1 Memory Controller without OCP wrapper

Memory Controller is the Master of the interface which controls the operations of the Memory. A simple Controller with Memwrite as the write control signal and Memread as the read control signals is designed as the Master [5], [6].

5.2 Memory without OCP wrapper

A simple SRAM is designed as the slave in the interface. It performs either write or read operation in response to the control signals from the master [5], [6].

5.3 OCP Compliant memory Controller

Memory Controller is the master who gives control signals to the memory which is the slave. The memory controller is reconfigurable. The control signals are for controlling the write and read operation of the memory. This memory controller is wrapped with an OCP wrapper which contains the basic OCP signals and will serve as an OCP master. OCP Master command is for Transfer command and this 3-bit signal indicates the type of OCP transfer the master is requesting. Each non-idle command is either a read or write-type request, depending on the direction of data flow. According to the Master command the slave will be either written into or read from [1].

5.4 OCP Compliant Memory

Memory is the slave which responds to the transfer requests provided by the master. Data can be written into the memory and read from the memory. This memory is enclosed within a wrapper which contains the basic OCP signals and will acts as an OCP slave. The response signals will be sending back to the master [1].

5.5 OCP Compliant Memory Interface

The entire system acts as a memory system interface which is a suitable alternative for a memory interface for any SoC design [3]. The design covers peripheral profile with simple read and write and Generic profile with data hand shaking and basic burst transfer [2].

5.6 OCP Compliant Memory Interface with Burst data transfer

A Memory System is not complete without Burst Transfer. Burst is a set of transfers that are linked together

into a transaction having a defined address sequence and number of transfers. There are three general categories of bursts. In Imprecise bursts, Request information is given for each transfer. Length information may change during the burst. In Precise bursts, Request information is given for each transfer, but length information is constant throughout the burst. Single request / multiple data bursts (also known as packets) is also a precise burst, but request information is given only once for the entire burst. To express bursts on the OCP interface, at least the address sequence and length of the burst must be communicated. The implemented design for burst transfer is having a word size of 32 and address width of 8. The address is incremented by 4 on each transfer and the Burst length is fixed as 4 throughout the entire transfer. In the research the Burst address sequence is selected as INCR, which is incrementing Burst. Three modes of Burst data transfer are discussed here. Burst write, Single Request Multiple Read Burst transfer and Burst Write with combined Request and data. In single request Multiple Read, the request is given only once and multiple data is read. In Burst Write with combined Request and data, the data is written and read as a burst.

6. Experimental Observations and Results

6.1 Design Setup

Table2: Design Setup

Design method	VHDL based behavioral
Verification	Modelsim 6.3b
Synthesis platform	Xilinx ISE 10.1 , Synopsys Design Compiler(DC)
Hardware Platform	Xilinx Vertex 5
Power Analysis Platform	Synopsys Design Compiler(DC)

6.2 Simulation Results

Simulation output is shown for Memory Interface System with burst support. For the given burst transfer the simulation is done for 1500 ns.

This simple proof of concept design was used for verification of the propounded OCP compliant design.

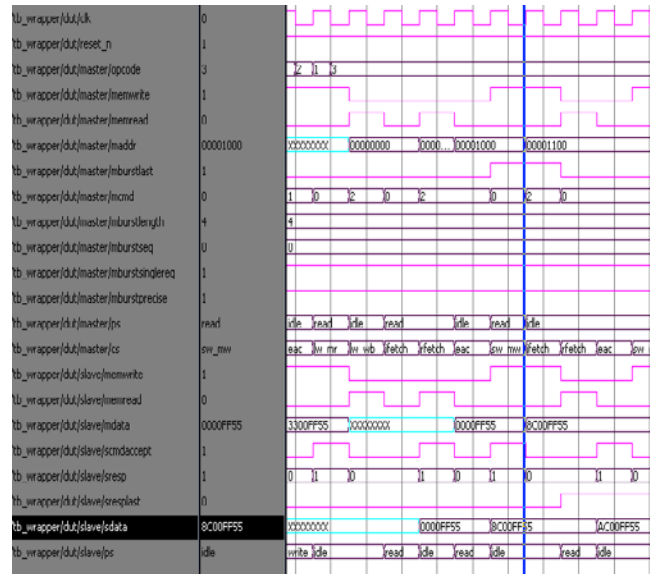


Fig. 3 Simulation Results with simple Burst transfer

Simulation results show that the signals are sampled at the rising edge of the clock signal (clk). OCP defines active low resets (reset_n). When the MCcmd signal (Master Command) signal is 000 it is the idle state. When MCcmd is 001 data is written into the memory and when MCcmd is 010 the data is read from the memory. For precise Burst data transfer, the address is incremented by four and for 32 word size, MBurstLength is held constant as 4. When the last address comes MReqLast will be high. When the last data is read SRespLast will be high. Only the most important signals and responses are shown here.

6.3 Synthesis

Synthesis was done both in FPGA and ASIC platforms. The performance analysis is based mainly on power consumption and speed. The results obtained are as shown in the tables below.

Table3: Frequency Analysis

Design	Frequency Of Operation
With wrapper	676.590MHz
With wrapper, with burst	558.566MHz
Without wrapper	633.309MHz

Table shows that the speed of operation increases with the use of OCP wrapper. But addition of burst transfer decreases the speed of the system. It is obvious since introduction of bursts will increase the minimum period of operation.

Another major analysis is on the power consumption.

Table3: Power Analysis (Xilinx Xpower Results)

<i>Design</i>	<i>Power in watts</i>	
	Memory Sytem Without Wrapper	Quiescent power
Dynamic power		0.007
Total power		0.041
Memory System With Wrapper	Quiescent power	0.034
	Dynamic power	0.013
	Total power	0.047
Memory System With Burst Transfer	Quiescent Power	0.303
	Dynamic Power	0.042
	Total power	0.345

Table shows the increase in power with the OCP wrapper is not too high when compared with the original designs. Hence the proposed system can be used as an alternative to ay memory systems. The burst transfer consumes high power since the frequency of operation is high as shown in the frequency analysis.

Further research is done with Synopsys tool design compiler which is based on 13 micrometer technology library. When the design is synthesized, each module is mapped to the gates and modules available in the specific technology Library.

In the analysis of power different modes of burst transfers are considered because the systems incorporating bursts are seemed to consume large power. The results are given in the table below.

Table5: Power Analysis with different burst transfers

<i>Design</i>	<i>Dynamic Power</i>
Memory Sytem Without Wrapper	90.0266 uW
Memory Sytem With Wrapper	100.7338 uW
Memory System With Simple Burst Transfer	114.9890 uW
Memory System With Single Request Multiple Burst Transfer	117.7919 uW
Memory System With Burst With Combined Request And Data	117.8017 uW

The results show that different Burst transfers have almost same power and hence any of them can be used efficiently according to the need.

5.4 Power Reduction:-Multi voltage Design

<i>Supply Voltage for Memory Controller</i>	<i>Design</i>	<i>Power In Millie Watts</i>	<i>% Power Saving</i>
At 5V	Slave	448.449	
	Master	947.184	
	Wrapper	1.40e+03	
At 4.75V	Slave	442.647	2.95%
	Master	910.250	
	Wrapper	1.35e+03	
At 4.5V	Slave	180.327	7.11%
	Master	851.923	
	Wrapper	1.03e+03	

Energy efficiency has become a very important issue to be addressed in today’s system-on-a-chip (SoC) designs. Multi supply voltage (MSV) is thus introduced to provide flexibility in controlling the power and performance tradeoff. One of the most effective ways is by lowering the voltage supply and has become the latest technique for power optimization. Multi voltage design provide “just enough” power to support different functional operations [11]. For dynamic power, a minor adjustment to the voltage level can result in a significant reduction in power consumption, which is proportional to the square of the voltage.

Design Compiler has a provision to set varying voltage for different designs. At the time of synthesis Synopsys uses a default voltage of 5V for 13um technology. This voltage is

<i>Supply Voltage for Memory Controller</i>	<i>Design</i>	<i>Power In Millie Watts</i>	<i>% Power Saving</i>
At 5V	Slave	448.449	
	Master	947.184	
	Wrapper	1.40e+03	
At 4.75V	Slave	442.647	2.95%
	Master	910.250	
	Wrapper	1.35e+03	
At 4.5V	Slave	397.279	7.11%
	Master	816.955	
	Wrapper	1.21e+03	

kept the same for the Memory Block since it contains the critical path. It is evident that the decrease in voltage will increase the delay. The voltage for the Memory Controller is reduced to 4.75 and then to 4.5 V. Percentage saving in power is given as %saving = Total power- (Reduced Master Power+ Original slave Power) / Total power. The power of both slave and master is reduced .But the power reduction analysis is done by keeping the power of slave as same as that at 5V since the aim is to reduce the power of master only since it has less delay. Hence the

power delay product is maintained which is used as the parameter to find the operating voltage of the circuit at which energy dissipation is minimal. The results of Multi voltage designs are shown in the tables below.

Table6: Power Reduction for system without Burst

Table7: Power Reduction for system with Burst

Table8: Power Reduction for system with Burst Single Request Multiple data

Table9: Power Reduction for system with Burst Combined Request and Data

<i>Supply Voltage for Memory Controller</i>	<i>Design</i>	<i>Power In Millie Watts</i>	<i>% Power Saving</i>
At 5V	Slave	139.266	
	Master	923.895	
	Wrapper	1.06e+03	
At 4.75V	Slave	139.020	4%
	Master	878.095	
	Wrapper	1.02e+03	
At 4.5V	Slave	124.771	12.51%
	Master	788.096	
	Wrapper	912.868	

From the tables it is clearly shown that the power can be reduced significantly with Multi voltage design for different Memory Transfers.

6. Conclusions

A parameterizable and reconfigurable OCP compliant memory system specifically targeted to use with high speed applications is discussed here. The primary trigger to the development of such design is the lack of availability of a common interface that can be used with the different IP cores in a SoC design. This paper discusses the use of OCP for a memory system interface and concentrates on enhancing the memory system performance with different modes of Burst data transfer. Power Reduction with Multi voltage design is implemented with good results.

References

[1] "Open Core Protocol Specification 3.0", International Partnership, 2000- 2009 OCP-IP Association, Document Revision 1.0.
 [2] Chih-Wea Wang, Chi-Shao Lai, Chi-Feng Wu, Shih-Arn Hwang, and Ying-Hsi Lin, "On-chip Interconnection Design and SoC Integration with OCP", Proceedings of VLSI-DAT, 2008, pp. 25 – 28, April 2008.
 [3] OCP-IP, "Open core protocol international partnership," <http://www.ocpip.org/>, 2007.

[4] JamesAldis, "Use of OCP in OMAP 2420" <http://www.ocpip.org/>,2005.
 [5] www.mips.com, "Computer Architecture and Engineering", Lecture 8 ,Designing a Multicycle Processor.
 [6] David A. Patterson, John L.Hennessy, "Computer Organization and Design", Third Edition, Morgan Kaufmann Publishers, pp.318-339.
 [7] Shihua Zhang, Asif Iqbal Ahmed and Otmane Ait Mohamed, "A Re-usable verification Framework of Open Core Protocol", Circuits and Systems and TAISA Conference, 2009, pp. 1-4, june

<i>Supply Voltage for Memory Controller</i>	<i>Design</i>	<i>Power In Millie Watts</i>	<i>% Power Saving</i>
At 5V	Slave	985.036	
	Master	323.353	
	Wrapper	1.31e+03	
At 4.75V	Slave	888.996	2.5%
	Master	291.826	
	Wrapper	1.18e+03	
At 4.5V	Slave	797.880	4.81%
	Master	261.916	
	Wrapper	1.06e+03	

28,2009.

[8] W.-D. Weber, "Enabling reuse via an IP core-centric communications Protocol", In Proc. IP 2000 System-on-Chip Conference, pages 217-224, Mar 2000.
 [9] Prashant D. Karandikar, "Open Core Protocol (OCP) An Introduction to Interface Specification", 1st Workshop on SoC Architecture, Accelerators & Workloads Jan 10 2010.
 [10] Chien-Chun (Joe) Chou, Konstantinos Aisopos, David Lau, Yasuhiko Kurosawa and D. N. (Jay) Jayasimha, "Using OCP and Coherence Extensions to Support System-Level Cache Coherence", Technical Paper, pg. nos.10, April 2009.
 [11] Qiang Ma and Evangeline F. Y. Young, Multivoltage Floor plan Design, IEEE transactions on computer-aided design of integrated circuits and systems, vol. 29, no. 4, April 2010 607