International
Journal
Of Advanced Research in Engineering & Management (IJAREM)
ISSN: 2456-2033 || PP. 07-13

Design and Coverage Driven Verification of AXI2OCP Bridge for Industrial SoC Designs

Limbanna. P. Rathod*, Dr. Jayanthi. K. Murthy†, Mr. Shaik Chand Basha§

*Post Graduate student, † Associate Professor

*-†Dept. of ECE, BMS College of Engineering, Bengaluru, India

§ DV Engineer, SION Semiconductors, Pvt. Ltd, Bangaluru, India

Abstract: The most popular communication architectures for SoC designs are advanced peripheral bus (APB) and the advanced high-performance bus (AHB) of AMBA from ARM. But in recent years some advanced communication protocols such as Advanced eXtensible Interface (AXI) and open core protocol (OCP) are introduced to facilitate parallel communication. These protocols works with different bus widths, protocols and frequencies. Currently semiconductor industries commonly use AXI and OCP in their SoC to support wide range of features. Any mismatch in communication speed and protocol leads to incorrect interaction between the cores and IPs which eventually results in unexpected behavior of the SoC Design. For the synchronized communication between AXI and OCP protocols, it requires advanced AXI2OCP Bridge which utilizes advanced technique for synchronization purpose between them. In order to design and verify the expected behavior of the bridge design, functional verification is a necessary process and complex too. In order to check the complete functionality of AXI2OCP Bridge it is necessary to build a test environment and generate the stimulus for many test-cases to examine the various features of AXI and OCP protocols. This paper mainly deals with the design of such advanced bridge for effective communication between such protocols and implementation of functional coverage and assertions to prove its accurate functionality using Mentor Graphics Questa-SIM tool. A measure of the bus utilization of AXI 3.0 protocol is also presented.

Keywords: AXI 3.0 Protocol, AXI20CP Bridge, System Verilog, Functional verification and Bus utilization.

I. Introduction

The increasing complexity of modern System-on-Chip (SoC) designs leads to more number of Intellectual Property (IP) blocks to be integrated into a chip. In the present market, bus protocols have much importance in the field of SoC design. There are various types of standard protocols available and are used in SoC which requires a bridge to pass the information from one type of protocol to other type of protocol safely and without any data loss. In SoC verification, it is an essential task to check the primary characteristics of standard protocols and IPs, due to increasing complications in the SoC designs.

To verify SoC there are different verification methodologies present in industries. The verification of a design is developed in the form of IPs called as Verification Intellectual Property (VIP).

In SoCs many processors, peripherals and memories are connected as illustrated. In Fig. 1 an Advanced Microcontroller Bus Architecture (AMBA) based SoC design is shown.

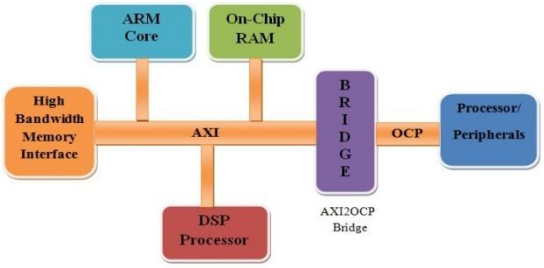

Fig. 1. Typical AMBA based SoC design

Here the processors and memories are connected inside SoC to high performance, high bandwidth interfaces that are connected to the processors or peripherals through the bridge. In SoCs the bridge is a term which can be seen inside a SoC, where we cannot connect two protocols directly, because formats or specifications changes from one protocols to another protocols. To synchronize between both protocols there is need of a bridge. This paper proposes to synchronize both protocols by using a bridge called as AXI2OCP Bridge. This paper is organized as follows: section II discusses on AXI and OCP protocols; section III briefly discusses on implementation de- tails; section IV and V presents experimental results obtained and conclusion.

II. Related Work

A. Advanced eXtensible Interface (AXI) Bus

The AMBA is used as on-chip bus in SoC designs. The AMBA based AXI protocol (AMBA 3.0) is targeted for high- performance and high-frequency system designs it consists of number of features, to make it easy for a high-speed sub- micron interconnects, such as

- It has distinct address phase and data phase.
- Separate Read and Write data channels.
- Supports unaligned data transfers using byte strobes.
- Burst-based transactions with only start address issued.
- Supports out of order transaction completion.
- Supports overlapping transfers.

AXI consists of five different channels to read and write such as, read address channel, write address channel, read data channel, write data channel and write response channel as illustrated in Fig. 2. Each AXI transaction contains address and control information signals and two-way READY and VALID handshake mechanism to exchange information between sender and receiver. By using write data channel data is transferred to slave and through the read data channel data is get back to master. For read request transactions read address channel and read data channels are used. For write request transactions write address channel, write data channel and additional write response channels are used.

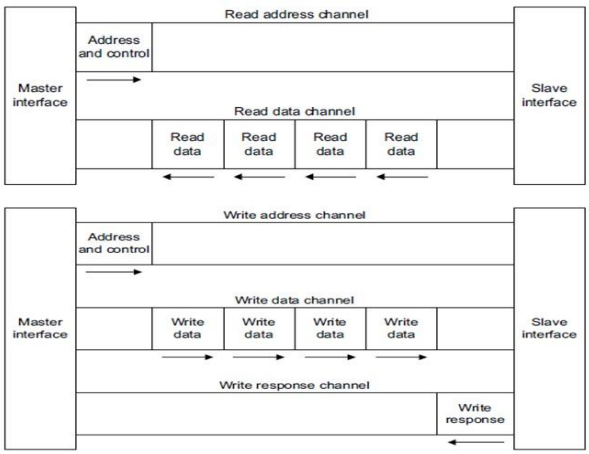

Fig. 2. AXI channel architecture of Read & Write

B. Open Core Protocol (OCP) Bus

The OCP protocol is openly licensed, core-centric protocol and it allows very high performance information transfers. A large complex SoC communication models supports thread identifiers to control out-of-order completion sequences of many parallel transfers. The OCP protocol defines a point- to-point interface between two communicating entities such as IP cores and bus interface modules. One entity acts as master of the OCP instance (IP core) and another one is acts as slave illustrated in Fig. 3.

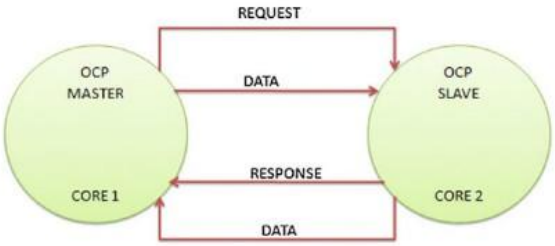

Fig. 3. OCP master and slave communication

In OCP the master will provides commands and acts as controlling part while the slave will responds according to commands taken from master. OCP acts like peer-to-peer communication between master and slave, where the first entity can be a master or slave which depends upon condition of transfer or commands provided by designer. OCP has some of the features;

- Point-to-Point Synchronous Interface.
- Pipelining & Bus Independence.
- Separation between requests and responses.
- Support of Bursts through annotation of transfers with burst information.
- Support for transmission of in-band information.
- Out-of-order request and response delivery using mul- tiple threads and tags.

III. AXI2OCP Bridge Implementation

The AXI2OCP Bridge design called as design under test (DUT) is implemented by using verilog hardware description language. According to the DUT creating verification environment of AXI2OCP Bridge and checking the response from the design is also one of the verification criteria developed in this paper.

Input to the bridge:

- AXI transactions: AXI write/read address, AXI data phase and AXI read data/response phase.

Output from the bridge:

- OCP transactions: OCP write/read address, OCP data phase.

The AXI data and OCP data operates at different rates i.e. incoming from AXI and outgoing from OCP may be at faster or slower rates. To synchronize between these protocols we have to use buffers in designing architecture to store the transactions i.e. address phase, data phase and response phase informations. Firstly, we have to develop AXI2OCP bridge design after completion of design we need to verify it.

To verify the design we need to develop different test-cases inside system verilog environment. These developed test-cases are intern provided to the design (DUT). With the help of these test-cases verification of features corresponding to read, write and write/read phase were done.

A. Proposed verification Environment

AXI2OCP bridge protocol verification environment is con- structed by using SV assertion based functional coverage and code coverage. This proposed implemented environment is able to improve the coverage and time for verification can be reduced. Realistic waveforms are created in Questa-SIM tool by Mentor Graphics.

The block diagram of SV environment as shown in Fig. 4 consists of some of the main components

such as AXI master, AXI interface, AXI2OCP Bridge design (DUT), OCP interface and OCP slave. Each module of AXI2OCP Bridge for SV environment is designed with respect to their functionalities. Here DUT acts as bridge, whatever transactions generated by master will be interfaced to DUT through AXI interface and transactions generated by the DUT are interfaced to slave through OCP interface.

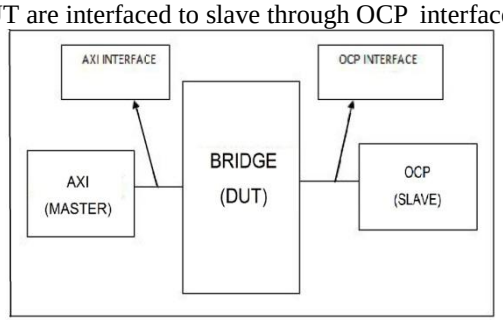

Fig. 4. The block diagram of AXI2OCP Bridge Environment

The individual modules present in AXI2OCP Bridge design are to be instantiated in the top module. Here, we need to maintain the proper synchronization between AXI master and OCP slave. When the data is synchronized between AXI and OCP then we will get the exact handshake signals as well as correct response from the OCP slave. Both AXI and OCP are on-chip communication protocols resides inside SoC design. The AXI and OCP have different set of signals, and there is a need to map those signals by using AXI2OCP Bridge. AXI2OCP Bridge performs various functions like,

- It maps AXI address or command signal format to OCP address or command signal format.
- It maps AXI data signals to OCP data signals.
- It converts AXI responses to OCP responses.

The mapped signals of AXI and OCP are as shown in table I.

TABLE I. COMPARISON OF SIGNALS

AXI	OCP SIGNALS
AWADDR	OCP MADDR
AWID	OCP_MTAGID
AWLEN	OCP_MBURST
WDATA	OCP MDATA
WLAST	OCP_MDATA
BRESP	OCP
ARADDR	OCP ARADDR

The internal architecture of SV verification environment for AXI and OCP is as shown in Fig. 5. This consists of different components such as AXI generator, AXI bus functional model (BFM), AXI monitor, AXI coverage, AXI2OCP (DUV), OCP monitor, reference model, checker, AXI and OCP assertions. Here examining the response from DUT/DUV is used as the verification measure.

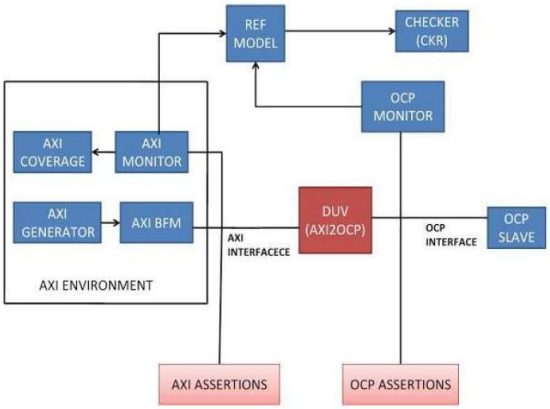

Fig. 5. DUV interfacing with test-bench

The DUV test-bench environment as shown in Fig. 5, needs to design a generator in such a way that each time it should generate different test-cases at AXI master. This will be possible only in SV constraint randomization process. After generating test-cases in the generator, these test-cases need to pass to the DUT. This can be achieved by mailbox, which is connected between the generator and the BFM.

The mailbox behaves as synchronizer i.e. the transactions generated by generator synchronizes with the BFM. Here, BFM plays an important part in verification, where the BFM will receive the transactions generated from generator and drives them to the AXI interface shown in Fig. 5. AXI interface is connected between the master and bridge (DUV). Here we need to concentrate on handshaking that will happen during communication. The AXI interface consists of all signals related to transactions and also output that can be taken for master across AXI interface.

The AXI interface in-turn provides AXI monitor to observe whether the AXI master is generating transactions or not. Now the DUT provides AXI transactions to slave through OCP interface. The OCP interface is connected between AXI2OCP Bridge and slave. The OCP interface in-turn provides OCP transactions to reference model through OCP monitor.

The role of Monitor is to keep on checking on valid trans- actions and the handshaking of signals. The monitor is to be designed according to AXI transactions i.e. transactions are concentrated by AXI interface. From the AXI monitor, signals are provided to reference model and coverage block to check the functional coverage. The reference model compares both the transactions of AXI and OCP. The checker provides final result i.e. write/read transactions obtained from reference model. The assertion blocks are provided for assertion based functional verification of design.

In this work a number of distinct types of test-cases are implemented and verified.

1) Read phase involves two channels those are read address, read data, and response channels need to examine.
2) Write phase involves three channels those are write address, write data and write response channel. The write phase is splits into three response channel. These are necessary to examine if an write phase handshaking of signals is happening perfectly or not in all the three channels.
3) Write/read phase involves whole write and read chan- nels. Here, in this test-case all five channels are examined.

IV. Experimental Results

This design has been successfully implemented and verified on Questa-SIM tool for various test-cases in an industrial design. The simulation result of AXI2OCP Bridge for different test-cases like write, read and write-read transactions were observed.

A. Simulation Waveforms

As shown in Fig. 6 simulation waveform of AXI for randomly generating data and addresses, and generated data written on to OCP slave is shown. The master writes data on dif- ferent address locations with write address size of 32-bit

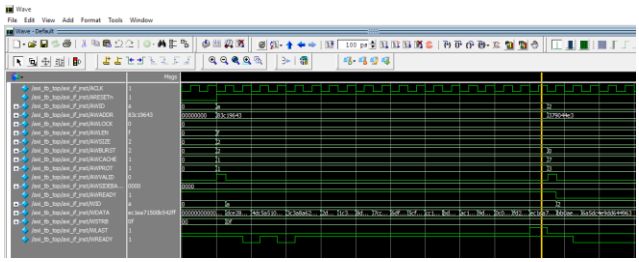

Fig. 6. Simulation waveform for write transaction

(AWADDR [31:0]) and data size of 64- bit (WDATA [63:0]) is shown. At OCP side same data appears through the signal MData [63:0].

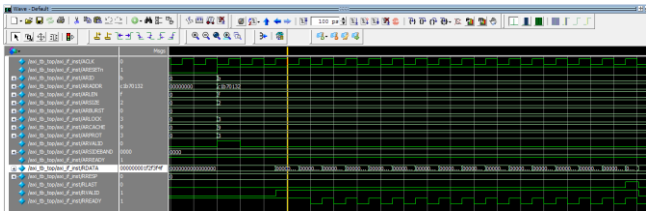

Fig. 7. Simulation waveform for read transaction

The read transaction simulation waveforms are as shown in Fig. 7. In read operation the data is being read from the OCP port with 64-bit of data and 32-bit of address.

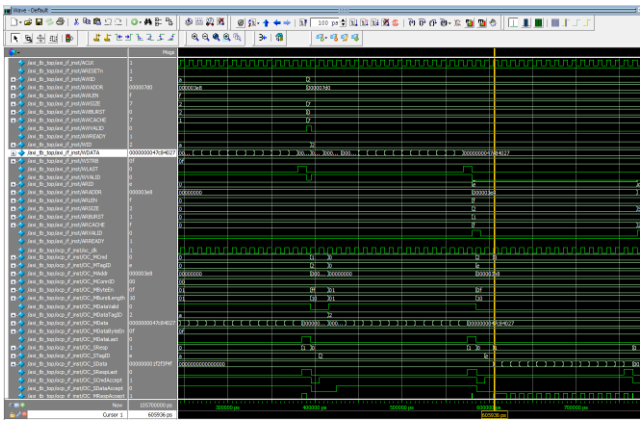

Fig. 8. Simulation waveform for write-read transaction

The write-read transaction simulation waveforms are as shown in Fig. 8. The write-read operation occurs simultaneously, where the AXI master is writing data into OCP master with same data and same address location and AXI master reading the same data from OCP slave with same address locations. Write-read transaction operation is using same 64-bit of data and 32-bit of address locations

B. Bus Utilization:

Bus utilization is the percentage of bus cycles used for transferring transactions. The write transaction shown in Fig. 6 took totally 19 clock cycles. The read transaction shown in Fig. 7 took totally 17 clock cycles. The bus utilization for read and write in percentage are shown in table. II. Read operation acquired 94% of bus usage.

TABLE II. BUS UTILIZATION FOR WRITE AND READ TRANSACTIONS

Test	Data	Total	Bus Utilization
Write	16	19	84.21
Read	16	17	94.11

C. Coverage:

The coverage report of AXI2OCP Bridge is obtained on Questa-SIM tool. Fig. 9 shows 27 test-cases have passed to the design and obtained overall weighted average of 93.98 percent. Functional coverage obtained 100 percent with 140 covergroups, 3 directive with 18 out of 20 assertions hits for write and read transactions. The code coverage for DUT is obtained nearly 97 percent.

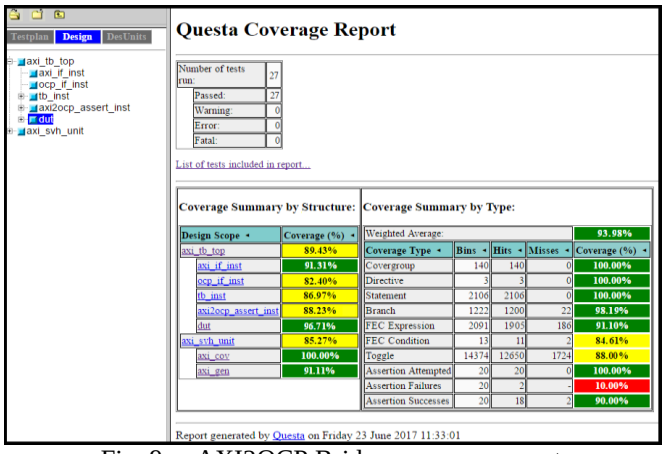

Fig. 9. AXI2OCP Bridge coverage report

V. Conclusion

This paper aims to design and verify various AXI2OCP Bridge features using coverage closure criteria. The assertion based functional verification and code coverage has been obtained. The practical waveforms and coverage reports are generated in Questa-SIM tool without any change in the design behavior and from coverage criteria, verification of the design is improved. In most of the industrial designs verification engineers prefer this coverage technique rather than the test- bench verification to reduce bugs in design. Different test scenarios are generated for AXI2OCP Bridge like write, read transfers, burst based transfers and write-read transfers. The effective bus utilization has been calculated for read and write transfers and obtained bus utilization for read to 94 percent.

Future work is to show some more unique features of AXI like protected and cache type of transactions. Automation can be used for code improvement to set up regression for all test cases as a group.

References

[1]. Golla Mahesh, Sakthivel S.M, "Functional Verification of the Axi2ocp Bridge using System Verilog and effective bus utilization calculation for AMBA AXI 3.0 Protocol", *IEEE Sponsored 2nd International Con- ference on Innovations in Information, Embedded and Communication systems (ICIIECS) (2015).*

[2]. Chin-Yao Chang, YI-Jiun Chang, Kuen-Jong Lee, Jen-Chieh Yeh, Shih- Yin Lin, Jui-Liangma, "Design of On-Chip Bus with OCP Interface", *IEEE conference in Hsin Chu, Taiwan, 2010, 978-1-4244-5271-2/10.*

[3]. G.Mahesh, Sakthivel. SM, "Verification of Memory Transactions in AXI Protocol using System Verilog Approach", in *IEEE ICCSP conference, 2015.*

[4]. Technical Data Sheet, "AHB to APB Bridge (AHB2APB)", *Part Number:T-CS-PR-0005-100 Document Number:I-IPA01-0106-USR, Rev 05 March 2007.*

[5]. Chris Spear, "System Verilog for Verification", *2nd edition, Synopsys, Inc. Marlboro, MA, USA: Springer, 2008.*

[6]. Roopak Sinha, Parthasarathi Roop, Samik Basu,"Correct-by- Construction Approaches by SoC design", *springer, 2014, ISBN:978-1- 4614-7863-8-7864-5.*

[7]. Ashok B. Mehta ,"SystemVerilog Assertions and Functional Coverage" *Guide to Language, Methodology and Applications, Springer 2014.*

[8]. V.N.M.Brahmanandam K, Choragudi Monohar, "Design of Burst Based Transactions in AMBA-AXI Protocol for SoC Integration", *International Journal of Scientific and Engineering Research Volume 3, July-2012, ISSN:2229-5518.*

[9]. Bergeron,Janick, "Writing Testbenches using SystemVerilog", *Synopsys, 2006, ISBN:978-0-387-31275-0*

[10]. Advanced Microcontroller Bus Architecture (AMBA) Specification 3.0, *http://www.arm.com.*

[11]. Open Core Protocol(OCP) Specification, *http://www.accellera.org/downloads/standards/ocp/.*