International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Design and SV Based Verification of AMBA AXI Protocol for SOC Integration

Rashmi Samanth, Subramanya G. Nayak

Abstract: ***Advanced Microcontroller Bus Architecture Advanced eXtensible Interface (AMBA AXI) provided by the ARM supports the high performance and high frequency system design. The System on Chip (SOC) Integration design needs to meet the low latency and high bandwidth challenges. The complex bridges are necessary when high frequency operations are carried out and there is a need for the interface which meets the requirement for the wide range of the applications, all such requirements without the complex bridges are provided by the AMBA AXI. The verification of such a bus protocol required to make the SOC integration most robust one. System Verilog based verification methodology provides the systematic way of verifying such a SOC to make it as a most reliable one. Also, SV based assertion makes the checking all the protocols specification easier at the verification stage. In the present paper, the general design and verification methodologies for the AXI-Bus and Memory interface for SOC integration is proposed. Verilog based memory and AXI design being done using Verilog HDL and design challenges are discussed. The proposed design implementation supports single and burst based data transfers. The AXI protocol provides the dedicated channels for memory read and write operations. In this work, single master and single slave communication using AXI protocol with 32-bit SARM are designed and implemented. The System Verilog based verification environment is setup and used for the verification IP development. And SV Assertion based verification is being done to thoroughly check the AXI protocol functionality.***

Index Terms: - AMBA AXI, Bus Function Model (BFM), Design Under Test (DUT), Network on Chip (NOC), SOC Integration, SV Assertion, Verification IP.

I. INTRODUCTION

The AMBA AXI's highly advanced features [1] enables its use in almost all highspeed SOC and NOC integration design applications. Since AXI's vast specification makes the design of the system complex and involves numerous challenges [2]. The design of such a complex system leads to developing a generalized predesign called Intellectual property (IP), which can be reused for different applications. System Verilog (SV) is the verification language which provides the various construct to well verify the intended design with all verification aspects [3]. It provides the high-level verification constructs and verification environment based on the object-oriented the object-oriented programming (OOPs). The verification of SOC integration are well implemented using these constructs, the verification IP developed are reusable [4].

In this paper AXI bus and memory interface is designed and verification IP for the same is developed. AXI bus high speed data transfer between the salve(memory) and master is discussed, the AXI bus has multi-channel for different types of data transfer between the slave and master, it could be single read-write or burst based data transfer [5]. The verification of the AXI protocol based on generating Bus Function Modeling (BFM) [6], and SV based verification environment is used [7] to develop a Verification IP for such transaction.

II. DESIGN AND IMPLEMENTATION OF THE AXI-MEMORY INTERFACE

AXI is burst based protocol and has 5 independent dedicated channels for the write and read operation between the master and slave. Read operation is carried out using read address and read data channels whereas write operation is carried out using write address, write data and write response channels. The address channels carry the read and write related control information [5]. AXI protocol supports the issue of the address information before the actual data transfer, also it issues the multiple outstanding and out-of-order completion transactions.

A. AXI Channel definition

A two-way handshake mechanism is provided with READY and VALID signals and set of information signals associated with every independent channel. The valid address, control or data information of the channel are available, then the VALID signal will be shown by the information source. The accept of the information by the destination is shown by the READY signal of the channel [5]. The final data item transaction is indicated by the LAST signal, which is included in both write and read channels. The signal description is provided in the Table I.

Write and Read Address channels: the write and read operation are carried out with address channel dedicated in each of them. The control and required address information for the transactions are carried by the appropriate address channel.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Rashmi Samanth*, Research Scholar, Dept. of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, INDIA.
Subramanya G. Nayak, Dept. of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, INDIA (corresponding

Table I AXI4 Signal Description

Signal	Source: Master /slave	Description
ACLK	Global	Clock signal
ARESETn	Global	System Reset
Write channel		
AWID [3:0]	Master	Write Address ID
AWADDR [31:0]	Master	Write address
AWLEN [7:0]	Master	Write address data length
AWSIZE [2:0]	Master	Write address data size
AWBURST [3:0]	Master	Burst length
AWVALID	Master	Write Address valid signal
WID [3:0]	Master	Write data ID
WDATA [31:0]	Master	Write data
WLAST	Master	Last write data
WVALID	Master	Write data valid
AWREADY	Slave	Write address slave ready
WREADY	Slave	Write address ready
BID [3:0]	Slave	Write response ID
BRESP [1:0]	Slave	Write data response
BVALID	Slave	Write response valid
BREADY	Master	Write response ready
Read channel		
ARID [3:0]	Master	Read address ID
ARADDR [31:0]	Master	Read address
ARLEN [7:0]	Master	Read address length
ARSIZE [2:0]	Master	Read data size
ARVALID	Master	Read address valid
RREADY	Master	Ready read data
ARREADY	Slave	Ready to read address
RID [3:0]	Slave	Read data ID
RDATA [31:0]	Slave	Read data
RRESP [1:0]	Slave	Read response
RLAST	Slave	Read last data
RVALID	Slave	Ready to read data

DATA Read channel: The read data and response information from the slave to the master are carried by this channel and it includes:

• The DATA Bus with 8, 16, 32, 64, 128, 256, 512, 1024 bits of data width

• The read transaction status completion indication response signal.

DATA write channel: This channel carries the writing data information to the slave from the master and includes:

• The DATA Bus with data width of 8, 16, 32, 64, 128, 256, 512, 1024 bits.

• A strobe of byte lane for indicating the valid bytes for every eight data bits.

The write transactions are carried out by the master without the acknowledgement from the slave for the previous write transaction. This feature is possible with treating of the write transaction information as buffered.

Write Response channel: This channel is used by the slave for responding to the write transaction from the master.

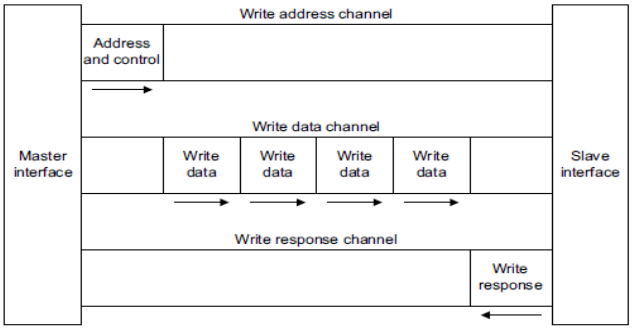

Fig .1: Write channel model

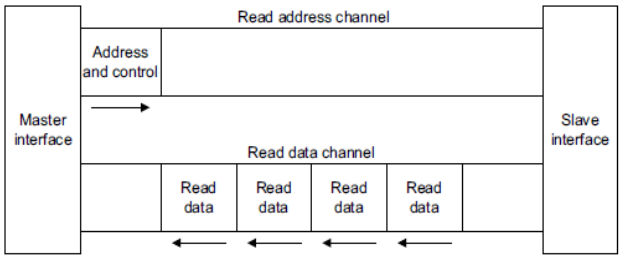

Fig .2: Read channel model

As shown in Figure 1 and 2 for write and read channels, the transaction completion is signaled only the completed ones, not every data transfer in a transaction are singled.

B. The Interface and Interconnect

Multiple master and slave devices connected through some form of interconnect in a typical system [8], this as shown in the Figure 3. The single interface is provided by the AXI protocol for different interfaces such as:

• Interface between an interconnect and the master

• Interface between an interconnect and a slave

• Interface between a slave and a master.

This definition of interface supports variety of implementation for different interconnects.

There are three different interconnect topologies used by the typical system are as follows:

• Shared Bus for Address and Data

• Shared Bus for Address and multiple Data

• Multiple address and data for multilayer

Most of the systems requires the address channel with the bandwidth, which is significantly less than the bandwidth of data channel required. And a good balance the interconnect and the performance of the system are achieved by such system.

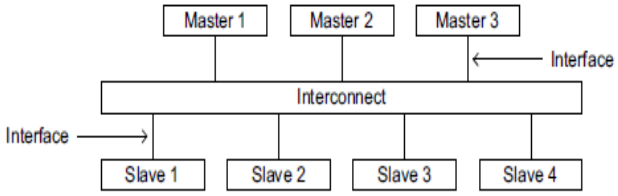

Fig .3: Interface and Interconnect

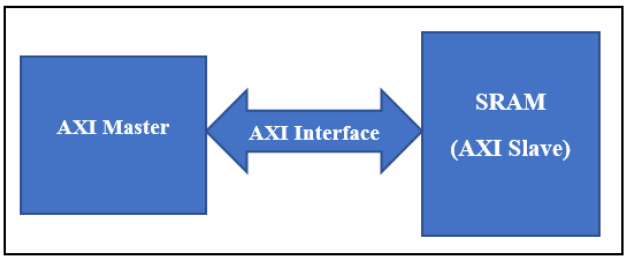

Fig .4: Single Master-Slave Interface

C. The Interface and Interconnect

In this work, single master single transaction is implemented with AXI protocol standard. The 32-bit SRAM is used as a slave, the operations like memory read and write are considered to demonstrate AXI protocol-based transactions. 32-bit address and data line are used to access the memory [10]. The Verilog HDL is used for the design implementation and VTB is used for verifying the SRAM. The proposed overall system implementation as shown in the Fig 4.

III. VERIFICATION METHODOLGY

The system-based verification as shown in the Fig. 5. Is being used in this work [3] [4] [6]. The brief description of each module has follows.

A. Test case

Test List of the test cases are included here. Each test case is different scenarios the transaction like singe write, single read, multiple read and multiple write operation, read after write, write after read etc. the sequence items connected to this test cases will be used to implement the test case scenario.

B. AXI Transaction Generator

Transaction generator is basically a class in SV, it is called a s a sequence item. This includes all the signals are being declared, used can be used in a transaction at any point, which give the status of the signal at that point in simulation run. All the signals are declared are random in this paper scenarios, this helps in generating the random values in the simulation run. These values are being used by the DUT in the later simulation time stamps.

C. AXI Master Transaction

The signals which needs to be driven from the master are included in this instance of the AXI transaction generator. By default, the AXI transaction generator instance values are driven to the DUT. The values are needs to be overridden from the master for test case scenario goes her.

D. AXI Slave Transaction

The signals which needs to be driven from the slave are included in the, generator instance of the AXI transaction. By default, the AXI transaction generator instance values are driven to the DUT. The values are needs to be driven from the slave for test case scenario goes here. Like master transaction.

E. AXI Score board

Basically, this module meant for comparison, this checks the validity of the input and output. The input signal values from the Master and slave are stored here, later comparison is being done using actual and expected results for the driven input values of the signals [11]. The checker will generate the expected results for the inputs used to drive the DUT.

F. DUT (SRAM)

This is the design needs to be verified, driven by the master for read/write operation by the master. In this paper, SRAM is exposed for AXI protocol based read and write operation.

G. AXI master and slave

This block is main block of master and slave, these blocks contains two sub-blocks intended for the Read or Write data and AXI master/slave BFM as shown in Fig.5.

Read/Write data: The objects of classes such as sequencer, driver and monitor are included in this sub block. Assigned sequence are picking and dropping into the driver are done by the sequencer. The driver drives these signals per the protocol. The monitoring of the signal for change as per protocol done by the monitor object [9].

AXI Master and Slave BFMs: The functions related to the buses are included in these classes. Finally, the signals driven from the driver are passed to the DUT.

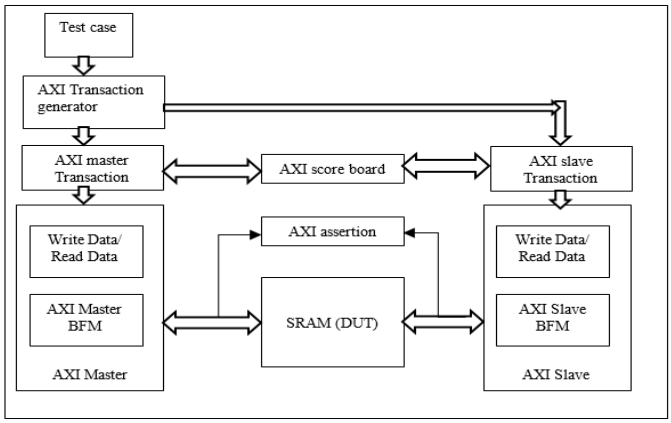

Fig. 5. SV based verification environment

IV. RESULTS AND CONCLUSION

The simulation is carried out based cadence NC-simulator, the results obtained are shown as follows. Fig 5 and 6 shows the design schematic and summary hierarchy and instances obtained from the cadence NC-simulator tool.

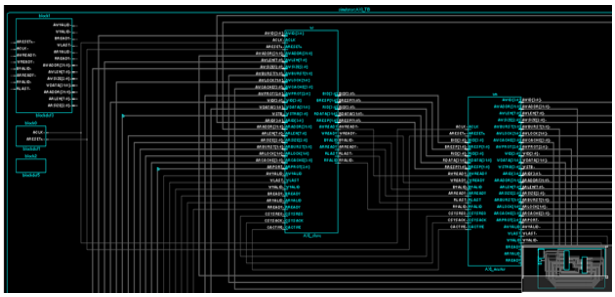

Fig. 6. Design Schematic

	Instances	Unique
Modules:	4	4
Registers:	91	91
Scalar wires:	48	-
Vectored wires:	51	-
Always blocks:	5	5
Initial blocks:	4	4
Cont. assignments:	1	4
Pseudo assignments:	28	28
Simulation timescale:	100ps	

Fig. 7. Design hierarchy and instance report

A. Simulation verification results

The data write AXI transaction is carried out for the single single data of write length for 32 bit data, back to back write transaction is implemented as shown in the Fig. 8, the write trasaction is accordance with the AXI write trasaction discussed in previous trasaction. The write operation at the slave level also captured which as shown in Fig 9. The when ever write enable singnal is enabled then data write happened at the memory.

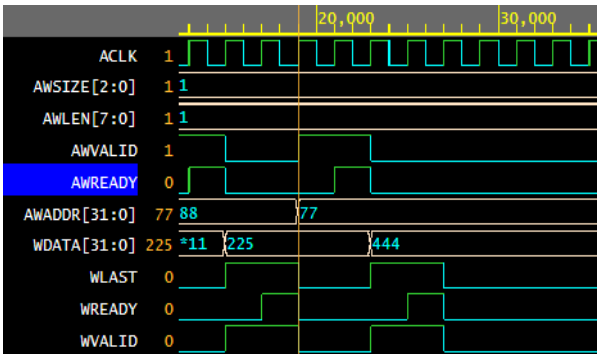

Fig. 8. AXI protocol single back to back write operation simulation results

Fig. 9. Data Write operation at slave i.e. memory level

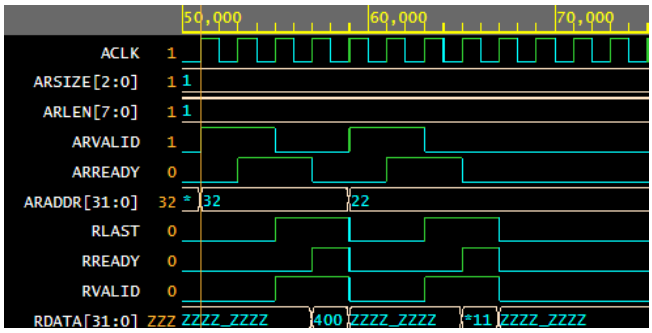

Fig. 10. AXI protocol single back to back read operation simulation results

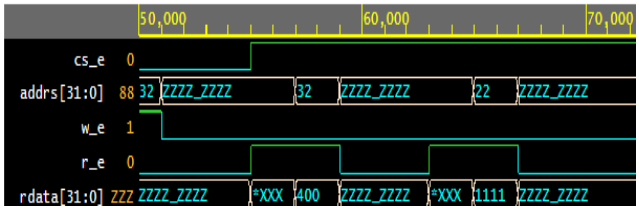

Fig. 11. Data Read operation at slave i.e. memory level

Like the write transaction, read transaction is carried out based on the AXI protocol. the back to back read operation from the different memory location are carried out, which as shown in Fig 10. The data read at the slave level also captured, the as read whenever read signal is enabled, like as shown in Fig 11.

B. Conclusion

The SOC integration based on AMBA AXI BUS protocol involves complexity, which induces numerous challenges design implementation and integration phases. In this paper the verification IP based on the AXI protocol is being developed and AXI based single master-slave read write transaction is carried out.

The verification IP is developed based on the systematic approach based on the system Verilog verification environment. This kind of verification environment provides the plat form for strategizing and developing the fully coverage driven verification IPs for any complex designs. This work can be extended to burst based transactions and make the integrated system works perfectly. So, SV based verification strategies can be adopted for any complex SOC Integration verification.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Dr. Sathish Kumar M, Head of Department, ECE, MIT, Manipal for his continuous and constant guidance. Also, we would like to thank other Professors, lab technicians of ECE department for extending their support and help by providing technical help and ideas for the paper, without which It would not possible to come up with this kind of paper.

REFERENCES

1. H. G., “Verification Of Amba Axi Bus Protocol Implementing Incr And Wrap Burst Using System Verilog,” *International Journal of Research in Engineering and Technology*, vol. 05, no. 03, pp. 201–206, 2016.
2. P. Guoteng, L. Li, O. Guodong, F. Qingchao, and B. Han, “Design and Verification of a MAC Controller Based on AXI Bus,” *2013 Third International Conference on Intelligent System Design and Engineering Applications*, 2013.
3. C. Spear and G. Tumbush, “A Complete SystemVerilog Testbench,” *SystemVerilog for Verification*, pp. 385–414, 2011.
4. N.-C. Chang, Y.-Z. Liao, and T.-S. Chang, “Analysis of shared-link AXI,” *IET Computers & Digital Techniques*, vol. 3, no. 4, p. 373, 2009.
5. P. Cotret, J. Crenne, G. Gogniat, and J.-P. Diguet, “Bus-based MPSoC Security through Communication Protection: A Latency-efficient Alternative,” *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, 2012.
6. G. Mahesh and S. Sm, “Functional verification of the Axi2OCP bridge using system verilog and effective bus utilization calculation for AMBA AXI 3.0 protocol,” *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015.
7. G. M, R. Sebastian, S. R. Mary, and A. Thomas, “A SV-UVM framework for Verification of SGMII IP core with reusable AXI to WB Bridge UVC,” *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2016.
8. P. S. R.., “Design and Verification Environment for Amba Axi Protocol for Soc Integration,” *International Journal of Research in Engineering and Technology*, vol. 03, no. 15, pp. 338–343, 2014.
9. ARM, AMBA AXI protocol specifications, Available at, http://www.arm.com, 2003
10. M SivaPrasad Reddy, B. Babu Rajesh, Tvs Gowtham Prasad, “A Synthesizable Design of AMBA-AXI Protocol for SoC Integration,” in International Journal of Engineering Inventions Volume 1, Issue 3 (September2012) PP: 19-26
11. A. Shrivastav, G. Tomar, and A. K. Singh, “Performance Comparison of AMBA Bus-Based System-On-Chip Communication Protocol,” *2011 International Conference on Communication Systems and Network Technologies*, 2011.

AUTHORS PROFILE

Ms.Rashmi Samanth is presently working as Research scholar in the Department of Electronics & Communication Engineering, Manipal Institute of Technology, Manipal. She received her B.E degree in Electronics & Communication Engineering and the M.Tech degree in Microelectronics from Manipal Institute of Technology, Manipal.

Dr. Subramanya G. Nayak is presently working as Professor in the Department of Electronics & Communication Engineering, Manipal Institute of Technology, Manipal. Dr. Nayak has B.E degree in E&C Engineering, M. Tech in Biomedical Engineering and Ph.D., in Electrical and Electronics Engineering with specialization in Pattern Recognition. His research areas of interest are in the field of Pattern Recognition and Processor Architecture design and applications. Dr. Nayak has published more than 40 papers in National / International conferences and in reputed Journals. His few papers are published as Book Chapters in Springer online. He is member of Editorial Board / Reviewer of many International Journals, DAC committee member of research scholars of MAHE and member of Examination Committees of other Universities. (corresponding author of the paper)