



ISSN 2395-650X

International Journal of
Life Sciences Biotechnology Pharma Sciences

IJLBPS

www.ijlbps.org

E-mail: editorijlbps@gmail.com editor@ijlbps.org

VERILOG BASED AMBA AXI PROTO COL

KARANAMJAYASREE¹,DR.P.AMMIREDDY².

¹ M.TECH(VLSI &ES)IN VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY,
VVITCOLLEGERD,NAMBURU, ANDHRA PRADESH 522508.

²PROFESSORINDEPTOFE.C.EINVASIREDDYVENKATADRIINSTITUTE OF TECHNOLOGY,VVITCOLLEGERD,
NAMBURU,ANDHRA PRADESH522508.

Abstract- This paper mainly focuses on verifying the important features of advanced extensible interface (AXI). Verifying the memory transactions of AXI includes the verification of all the five channels: write address, write data, write response, read address, and read data. In this work, a Verification Intellectual Property cores (VIP) based methodology is used to carry out the verification process. In the VIP design, the entire test environment is modeled using system verilog, and the read, write transactions from the same and different memory locations have been verified with the quantitative values of Busy Count, Valid Count, and its Bus Utilization. Verifying the system connectivity during write and read cycles is also one of the fundamental features verified in this paper.

KEYWORDS: WRITE AND READ TRANSACTIONS, AXI PROTOCOL, VERIFICATION IP, BUS UTILIZATION, AND COVERAGE MODE ANALYSIS.

INTRODUCTION

The rapid growth in CMOS technology and Computer Aided Design leads to usage of a large number of IP (Intellectual Property Cores) in the complex digital designs [1]. With the help of these IP cores integration, nowadays the SOC (System on Chip) design becomes more popular and intensively used for many applications [2]. Also all the SOC designs use bus protocols for making the data communication and synchronization [4-6]. Hence the reusability of a large number of IP cores in the complex design makes the functional verification process so crucial (i.e. as it involves 70% percent of the time span as compared to the 30% of the time span for design process). In order to overcome this difficulty and large time span, the verification engineers proposed a methodology for verifying the functionality of the chip using an inbuilt verification environment called as Verification IP (VIP) [7-10]. Normally the bus protocols used in the modern SOC are

APB (Advanced Peripheral Bus), AHB (Advanced high performance Bus) and AXI (Advanced Extensible Interface). On comparing these three bus protocols, the AXI bus protocols give better performance and consume moderate power. Hence all the SOC design involves the usage of AMBA AXI bus in the internal architecture. Normally in the functional verification, the failure mode analysis and fault robustness is performed for the design. For overcoming the lag in the functional verification process, a verification environment based methodology is proposed using System Verilog in coverage enabled mode. In this verification mode environment, the two operations are carried out as i) Proper test-cases are defined for the DUV (Device under Verification) & i) Determine the number of test-cases to analyze and cover all the functionalities of the design. During the verification process, the coverage enabled mode easily

identifies the bugs in the design by identifying the coverage of the code and validating the functional behavior of the design for all test scenarios.

EXISTING METHOD

AMBA-AXI BUS ARCHITECTURE

AMBA (Advanced Microcontroller bus architecture) is an on-chip bus protocol from

ARM. It tells about the on-chip interconnect specifications for establishing the connection between the processor, functional blocks and peripherals. The AXI bus protocol architecture is the most suitable and usable in modern SOC and FPGA. This is due to the characteristics of large bandwidth capability, interfacing with complex bridges, large memory access and high backward compatibility with peripherals. Also the AXI bus architecture is having separate read and write data phases with individual data channels, unaligned data transactions and burst mode access. This section explains the bus architecture of AXI protocol used for the data communication and synchronization operations. In this bus architecture the data transactions are established in two phases as a transaction is pictorially represented in the fig.1.

i) Read Transaction phase and ii) Write Transaction phase in the bus. The corresponding phases of operation are explained in the subsections with the master-slave configurations.

READ TRANSACTION PHASE

In the read transaction phase, the read address (AR) and read data response (R) signals are involved in the data communication and synchronization process. In the read transaction the first the valid address (AR) and Read channel access is

communicated by master to initialize the transactions and then the slave responds with a signal AR ready (i.e. making the address ready for the communication). After the reception of AR ready the master checks the address and then receives the RVALID and read channel data signals. Following to that the master establishes the channel communication with control signal with a valid RR ready and establishes the read transactions. The entire architecture/operation of the read

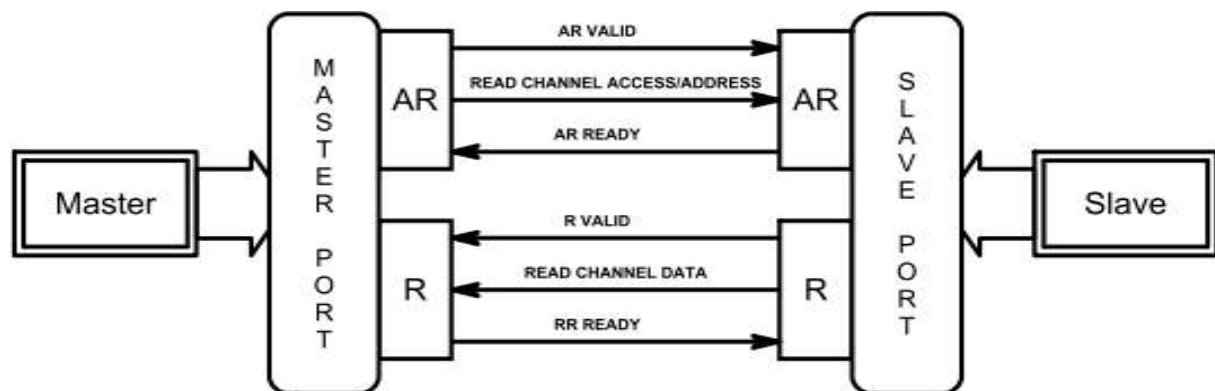


Figure 1: AXI Read Transactions

Write Transaction Phase

The write transactions are happening in the following three modes as i) Write address channel, ii) Write data channel and iii) Write response channel. In the write address channel mode, the master starts the communication with the control signals AW

VALID and Write channel access/address to the slave (i.e. it sends the valid address and channel control and address). Immediately the slave responds with a valid AWREADY. In the same manner during the Write data channel mode, the master initializes the transactions with a WVALID and Write channel data, following to that the slave acknowledges by a valid WREADY signal. Similar to the data and address mode, the Write response channel mode involves

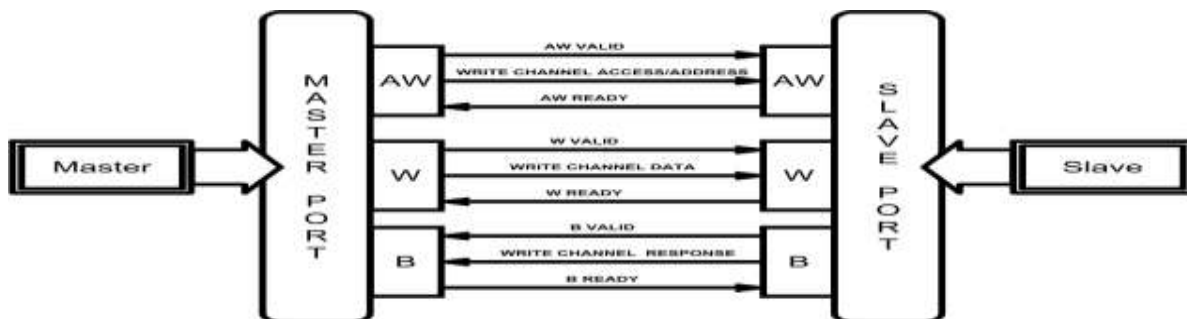


Figure 2: AXI Write transactions

the transaction started by master with the control signals BVALID and write response channel. Upon receiving the control signal, the slave establishes the communication by a valid BREADY signal. The operation of the write transactions are shown in the fig.2.

PROPOSED VERIFICATION ENVIRONMENT

The verification environment is a standard method of verifying the DUV in code coverage mode. The following components

are present in the proposed verification environment i) Test-suite, ii) Virtual sequencer, iii) Sequencer, iv) Driver, v) AXI-Monitor, vi) Protocol checker, vii) Bus Monitor, viii) Coverage Collector and ix) Scoreboard.

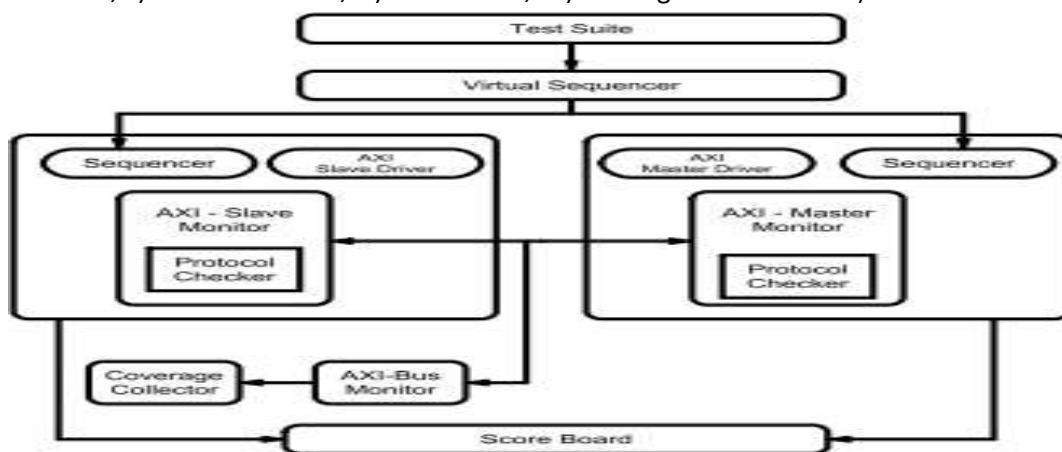


Figure - 3 Verification Environment

Test-Suite: Test-suite consists of all the test-cases for checking the specific key features of the DUV and the functionality in a constrained random mode. The test-case is of the form constrained random mode or direct random test-case. The test is regressively performed on the des-

ign and analyzed. It normally contains a set of stimuli for running the design.

Virtual Sequencer: Generally, the sequencer generates random sequences for verifying the key properties of the design. The generated sequence is called a top level

sequence and splitted into corresponding level the virtual sequencer verifies the functionality of each subcomponent by driving/forcing them with the set of sequences.

Sequencer: Sequencer is a component, which generates set of sequences. The generated sequences are of two types active and passive. The active sequence drives the signal whereas the passive does not drive the signal. The sequencer aligns and generates the sequence accordingly to the criteria of functionality checks.

Driver: Driver is the component which interfaces or forces all other components to respond accordingly to the feeded stimulus (i.e. simply it drives the input to the corresponding blocks).

Monitor: The monitor watches all the data transactions of all the components and reports the result to the coverage collector.

It has a TLM port for analysis and virtual interface to handle the points of the DUV signals.

Protocol checker: In this module, the DUV functionality is verified, the interface compliance is also checked with the temporal behavior of the inputs. Also the responses are monitored for all the combinations of the input sequences.

Bus monitor: Bus monitor screens the functional operation of individual components and connects the

master/slave monitor. Finally the transactions are collected in the coverage collector as a database.

Coverage Collector: Coverage collector collects all the code coverage information like branch, statement and FSM in the DUV is covered by the specified test-case or not. Normally the verification engineers aim for 100% of coverage during the functional verification.

VERIFICATION PLAN

The verification plan tells the property has to be verified and drives the coverage criteria to be satisfied. A good plan contains measurable metrics and successful test-case to achieve the coverage and functionality check. The plan of verification tells what to be verified, how to verify it. In this AMBA AXI verification the properties to be verified are i) System connectivity during read and write cycle,

base and primitive sequences. Thus ii) Transaction routing and iii) finally the Data integrity. In the system connectivity checking test, the system integration is monitored by the corresponding acknowledgement signals. Then during the routing stage the bus data occupancy structure is verified by the bus utilization and efficiency. Finally the data integrity is noted for every transactions and the functionality check pass is carried out for each of the stage.

In this AXI protocol, the verification plan focuses the coverage of the following scenarios as

❑ **Read phase**

❑ **Write phase**

❑ **Write & Read phase from same address**

❑

Write & Read phase from different address

ess

❑ **Overlapped transactions**

For the corresponding scenarios the test-cases are generated and the simulation is carried out for the functionality verification. Using the following bus performance metrics Valid Count, Busy Count and Bus Utilization factor, the performance of the bus is assessed. If the bus is not verified accurately in the corresponding test-cases, then the test-cases are modified accordingly to reach the specific property checks. Also in the verification, the code coverage based verification mode analysis is carried out for the DUV; if the 100% coverage is not achieved during the analysis the test-case is altered to cover all the code parts of the design.

RESULTS & DISCUSSIONS

Performance evaluations employing quality metrics for data metrics are presented in this part based on AMBA AXI simulation results in five separate test cases. The individual AXI Master/Slave modules are modeled using System Verilog and integrated into the modeled test environment for verification before the simulation analysis can begin. The DUV is integrated with the sequencer, interface, driver, and monitor to evaluate performance in a limited mode in the verification environment. Simulation is

conducted out with the Xilinx ise after the top module integration with DUV. In this example,

test-case is created to cover five various scenarios, including: (i) reading, (ii) writing, (iii) reading and writing from the same address, (iv) reading and writing from a different address, and (v) overlapping transactions.

VERIFICATION OF READ PHASE

In this phase the read data transactions of the AMBA AXI protocol is verified with the Master/Slave configuration mode. During the read cycle transaction the signals ARVALID, ARREADY, RVALID, RREADY, RLAST, RDATA and ARSIZE are monitored and verified. Here the functional ity check primarily involves two modes of operation namely i) Read Address and ii) Read Response channel. For the every positive edge of clock, the read address channel will fetch the address at the high state values of ARVALID and ARREADY. Similarly after a time delay,

the

the response will be instantiated to high mode with the high values of the signal on RVALID and RREADY. The last transaction of the read operation is indicated by the high values of RLAST. The values of ARSIZE and ARLEN are measured and indicate the number of transactions occurred in read mode. For verifying the read phase at test case is generated and simulated to test the functionality of the read operation. The working of the read mode includes two channels as explained above with the measure of the quality metrics for the read address, read channel data with channel response. The entire operation of the read phase is pictorially represented as a waveform in the fig.4.

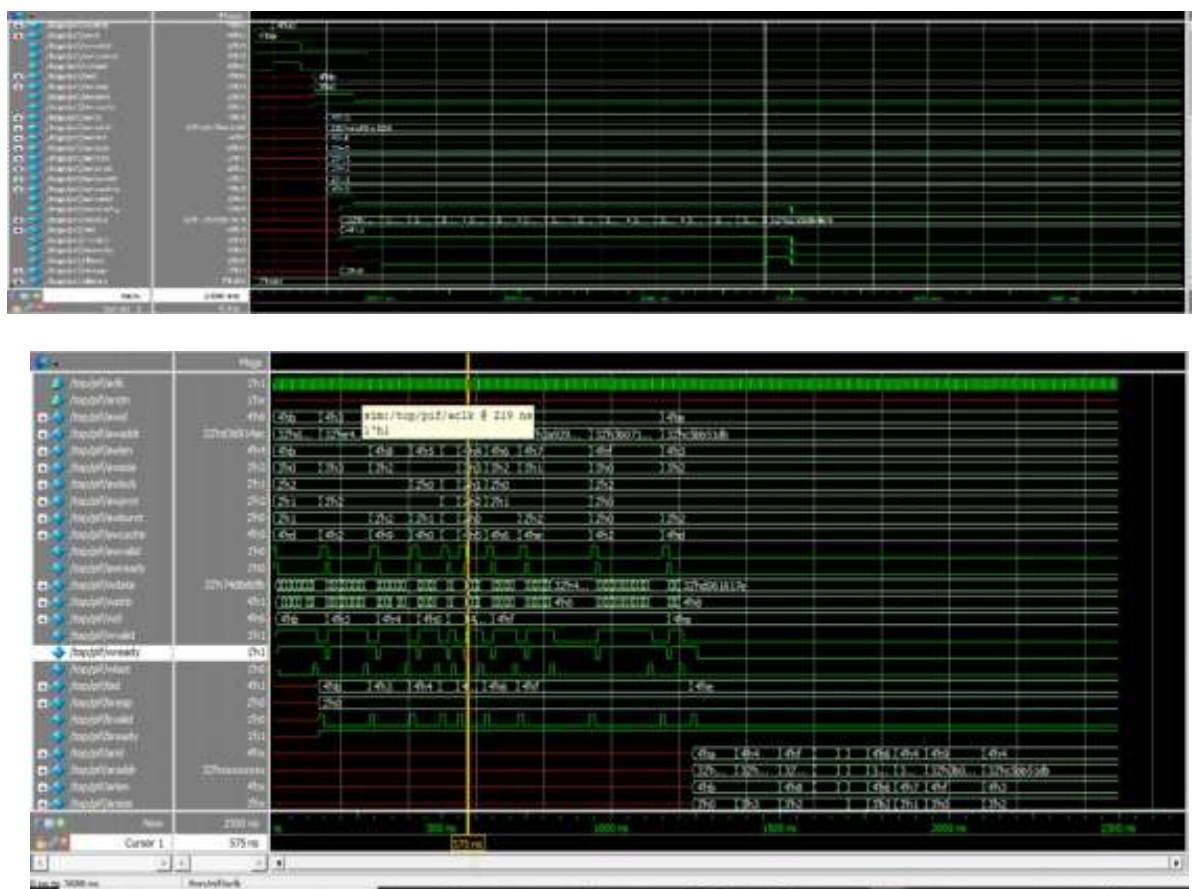
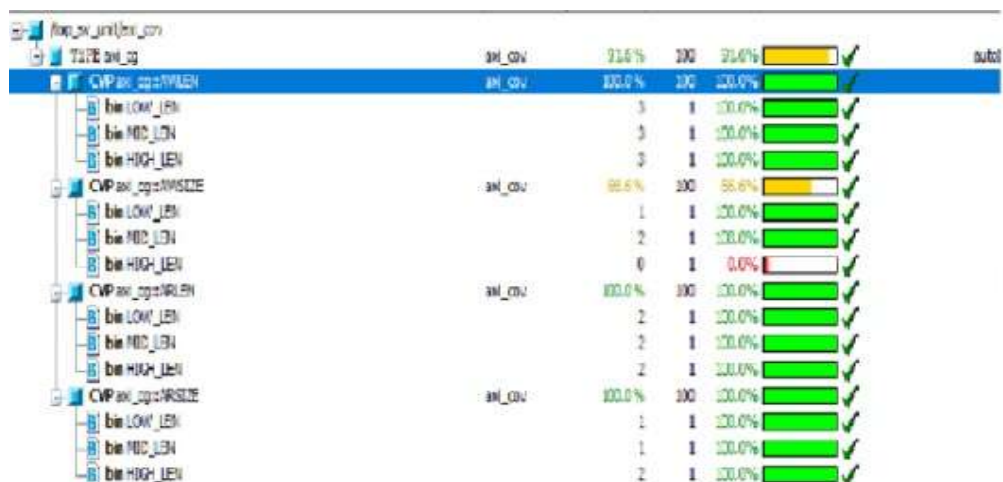


Figure4:ReadPhaseSimulationResponse

VERIFICATION OF WRITE PHASE

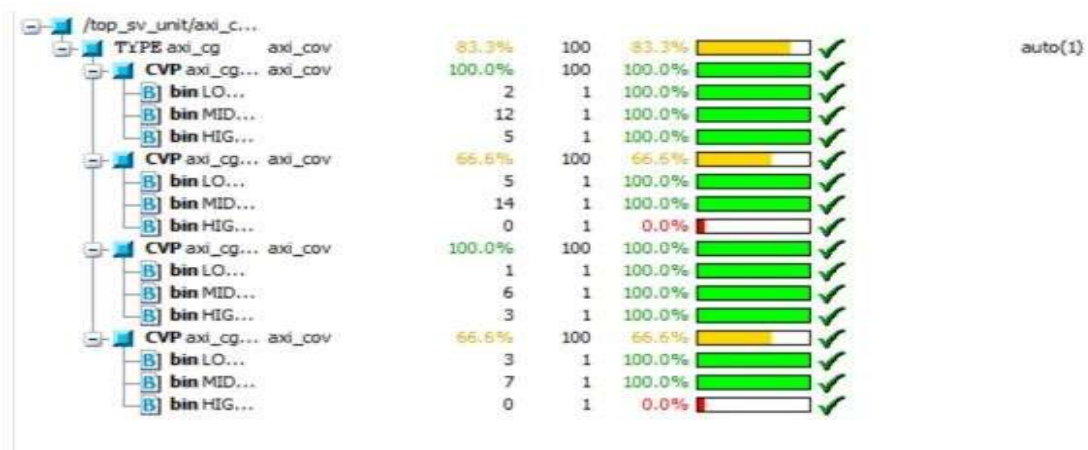


It's just like the Read Phase of the AMBA AXI protocol in that Master/Slave configurations must be verified for AWIDs, AWADDRS, AWLENS, and AWSIZES. Write Address Channel, Write Channel Data, and Write Response Channel are three of the core operations of the Write Phase. In order to get a valid write address and an acknowledgement AWID on each positive clock cycle, the signals WVALID and WREADY

The coverage report is 10 write and read compare transaction compare to previous we are more signals are covering and also 83.3% will be cover the transactions.

Fig 6: coverage report of 10 write read transaction

The coverage report is 16 write and read transaction compare to previous we are more signals are covering and also 91.3% will be cover the transactions



go into high logic state. In the same way, when the BVALID and BREADY transitions are high, the write response operation will occur. Control signals AWLEN and AWSIZE track the size of each transactions, with AWLEN starting at 0000 and incrementing with each new transaction. An illustration of Write Transactions is depicted in the figure 5.

Figure 5: Write Phase Simulation Response

Fig 7: coverage report of 16 transaction

More signals are being covered, and 75.0 percent of the transactions will be covered according to the coverage report, which compares transactions written and read.

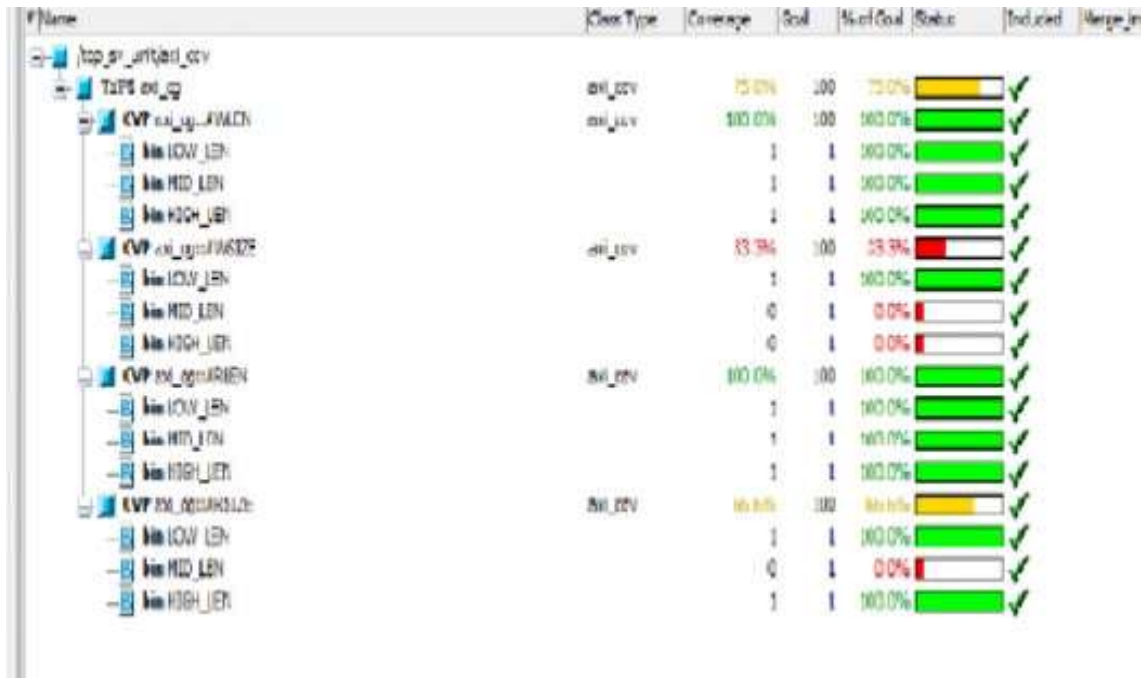


Fig 8: coverage report of 16 transaction repeat2time

CONCLUSION

We introduced a resource allocation algorithm deadlock avoidance condition and an AXI protocol architecture allowing tagged transactions. The proposed RAG simulation results using the AXI protocol can show the extensions for out-of-order transactions (Resource Allocation Graph). The efficiency of the system is increased by utilizing the orbiter techniques and multicasting components.

FUTURESCOPE

Master-slave communication can be improved with the addition of the suggested system's modular componentry. Reduces the complexity of the hardware, making it more useful for system verification.

REFERENCES

- [1] A Specification for Advanced Microcontroller Bus Architecture (1997) [Online].
- [2] For further information, visit <http://www.arm.com/>
- [3] [2] Open Core Protocol Specification. (2006) [Online].
- [4] Available:
- [5] <http://www.ocpip.org/home>
- [6] This paper was published in the proceedings of the 2011 IEEE 29th International Conference on Computer Design, and it is titled "RoShaQ: High-

performance On-Chip Router with Shared Queues."

[7]Proc. IEEE 13th International Symposium on High-Performance Computing Architectures, February 2007, pages 285–294.

[4] Shao and Davis, 'A burst scheduling access reordering technique'

[9]

[10] "A priorityexpression-based burst scheduling of memory reordering access," in the Proceedings of the International Conference on Parallel and Distributed Systems (PPDDS), by J. Pang, T. Zhang, D. Wang, and C. Hou.

Conf.EmbeddedComput.Syst.,Archit.,Model.,Simul., Jul. 2008, pp. 203–209.

[11]

X.XiaoandJ.J.Lee,"AtrueO(1)paralleldeadlockdetectionalgorithmforsingle-unitresourcesystemsanditshardwareimplementation,"IEEETrans.ParallelDistrib. Syst., vol. 21, no. 1, pp. 4–19, Jan.2010.

[12] A. Silberschatz, P. B. Galvin, and G.Gagen, Operating System Concepts, 7th ed.NewYork,USA: Wiley,1993.

[13] T. S. Cummins, "Method and apparatusfor detecting a bus deadlock in an electronicsystem,"U.S.Patent6292910,Sep.18, 2001.

[14]

TechnicalReferenceManualofPrimeCel

IAXIConfigurableInterconnect(PL300),ARM,
Cambridge,U.K., 2010.

[15]

K.Lahiri,A.Raghunathan,andG.Lakshmi
narayana, "The LOTTERYBUS on-
chipcommunicationarchitecture,"IEEETrans.V
eryLargeScaleIntegr.(VLSI)Syst.,vol. 14, no. 6,
pp. 596–608, Jun. 2006.

[16] K. Sekar, K. Lahiri, A. Deyand
Raghunathan, "dynamically customizable bus
topologies for high-performance on-chip
communication," IEEE Trans. Very Large Scale
Integrity (VLSI) Syst., volume 16, issue 10
(October 2008), pages 1413-1426,.

[17] F.-M. Xiao, D.-S. Li. G.-M. Du, Y.-
K.Song, D.-L. Zhang, and M.-L. Gao,
"DesignofAXIbusbasedMPSoConFPGA,"inProc.
3rdInt.Conf.AntiCounterfeit.,Security,
Identificat. Commun., Aug. 2009,pp.560–564.

[18] H.-W.Wang,C.-S.Lai,C.-F.Wu,S.-
A.Hwang,andY.-H.Lin,"OnchipInterconnection
Design and SoC Integrationwith OCP," in Proc.
IEEE Int. Symp. VLSIDesign,Autom., Test, Apr.
2008, pp.25–28.

[19] N. Y.-C. Chang, Y.-Z. Liao, and T.-
S.Chang, "Analysis of shared-link AXI,"
IETComput. Digit. Tech., vol. 3, no. 4, pp. 373–
383, Jul. 2009.

[20]

O.Ogawa,S.BayondeNoyer,P.Chauvet,
K.Shinohara,Y.Watanabe,H.Niizuma,T.Sasaki,a
ndY.Takai,"Apracticalapproachforbusarchitect
ureoptimization at transaction level," in
Proc.Design, Autom., Test Europe Conf.
Exhibit.,2003,pp. 176–181.

[21]

SynopsysDesignCompiler.(2010)[Onlin
e]. Available:
[http://www.synopsys.com/Tools/Implementa
tion/RTLSynthesis/Pages/default.aspx](http://www.synopsys.com/Tools/Implementation/RTLSynthesis/Pages/default.aspx).