International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 1, Issue 2, November 2014)

Design and Verification of a MAC controller with inclusion of Promiscuous mode, based on AXI bus

Ms. Anitha Pendyala
Pursuing M.Tech scholar, VLSI
Aurora's Scientific, Technological and Research Academy
Hyderabad, TS, India

Mr. A.Anendhar
Assistant Professor, Dept.of E.C.E
Aurora's Scientific, Technological and Research Academy
Hyderabad, TS, India

Mr. A.Ravichandra
Sr. Assistant Professor, Dept.of E.C.E
Aurora's Scientific, Technological and Research Academy
Hyderabad, TS, India

***Abstract*— Present day's communication became key role. For communication we use different medias or Technologies from those technologies we choose one of the technology is AMBA Bus .In AMBA we have different streams among them let we see about in this paper is AXI. In which AXI is designed and verified of a MAC controller, which system architecture integrates original design components and reusable IP cores and verification platform using System Verilog. The Verification is carried out on Directed random test and constraint random test technology to finish verification plan effectively. Finally SOC chip has been passed for Fabrication.**

***Index Terms*— SoC, Media Access control, AXI bus, MII.**

I.INTRODUCTION

Integrated circuits entering the System-on-a-Chip (SoC) design, which is integrating all components of a computer or any electronic system into a single chip. With the day by day increase in design size, IP is an inevitable choice for SoC design. Wide use of all IPs has changed its nature of the design flow by making On-Chip Buses (OCB) essential to the design.

In this paper we are intend to design and verify the Ethernet MAC controller based on AXI bus, and also it is extended by the concept of Promiscuous mode inclusion. Which will be checking the network media to help identifying malfunction of the network or diagnoses of connectivity issues in networks. The main content of this paper named as AMBA IO System Architecture design of configurable 10/100/1000Mbps data transfer rates. The Ethernet MAC controller of configurable10/100/1000Mbps supports the family of configurable PHY interfaces MII/GMII/RMII/RGMII to communicate with an external Gigabit/ Fast Ethernet PHY. With evaluation of every contention method the throughput has increased immensely, this will be clearly seen in CSMA with collision detection method.

This paper supports AXI 3.0 protocol from the ARM Advanced Microcontroller Bus Architecture (AMBA). AMBA bus protocol has become the main standard SoC bus. That means more and more existing IPs must be able to communicate with AMBA 3.0 bus. Based on AMBA 3.0 bus, this design is an Intellectual Property (IP) core of AXI (Advanced eXtensible Interface).

This verification of whole AMBA IO System Architecture design will be done by layered verification platform using System Verilog, which follow the methods of direct random test, constrained random test technologies to tape-out the designs successfully.

II.DUT DESCRIPTION

This paper investigates the implementation and verification of Ethernet MAC controller, which is incorporated with the project of network processor. it is configurable 10/100/1000 Mbps transfer date rates and It is also configurable MII/RMII/GMII/RGMII Media independent interface then it is verified with the system Verilog verification environment. Then Functional coverage verification process is carried out by directed random test and constrained random test. The main content is named AMBA IO system design of a 10M/100M/1000M Ethernet MAC controller and AXI bus, MAC controller could be configurable, Supports 10/100/1000 Mbps data transfer rates with the following PHY interfaces: IEEE 802.3-compliant GMII/MII interface to communicate with an external Gigabit/Fast Ethernet PHY; RGMII interface optional to communicate with an external gigabit PHY, AXI bus support AMBA 3.0 protocol. The AMBA IO system is tested automatically by a reusable, reliable, layered verification platform, which applies constrained random data frame generation, assertion check method, etc. The chip pass all test cases, is tape out successfully now.

The AMBA IO system comprises of AXI bus, and Ethernet MAC controller IP. MAC controller has AXI bus interface, and could be configurable, supports 10/100/1000 Mbps data transfer rates with the IEEE 802.3-compliant RGMII /GMII/MII PHY interfaces, AXI bus supports AMBA 3.0 protocol. Its architect structure illustrates as figure 1, test bench is beyond of gray frame, which includes device model, Ether PHY VIP CPU BFM Bus Function Model and Memory model.

International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 1, Issue 2, November 2014)

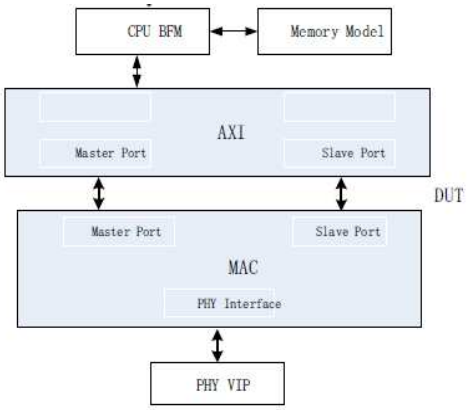


Figure 1.AMBAIO Architecture structure

The AXI specifications describe an interface between a single AXI master and a single AXI slave, representing IP cores that exchange information with each other. Memory mapped AXI masters and slaves can be connected together using a structure called an Interconnect block. The Xilinx AXI Interconnect IP contains AXI-compliant master and slave interfaces, and can be used to route transactions between one or more AXI masters and slaves. The AXI Interconnect IP.

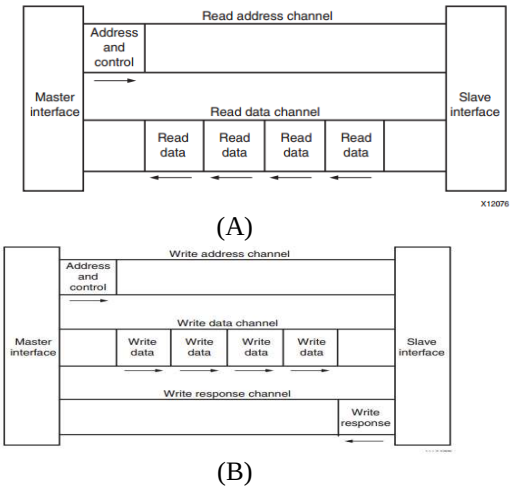


Figure 2.A) Channel Architecture of Reads B) Channel Architecture of writes

Every On-chip-bus is developed for special application, each has his strong point, and it is difficult to compare with their advantages and disadvantages. AMBA bus own many third party supporters, is used in SoC system based on ARM core by 90% cooperation of ARM corp. ,it is becoming one of On-Chip-Bus industry standards gradually. AXI/AMBA Advanced extensible Interface Bus vs Bus protocol is important content of AMBA 3.0, which is designed for high- performance, high-frequency system designs, and includes a number of features that make it suitable for a high-speed submicron interconnect. So, AXI bus is selected in our SOC design.

The AXI bus protocol is an enhanced bus protocol of the existing Advanced High-performance Bus (AHB). AXI is targeted at high performance, high-frequency system designs. AXI protocol has five independent unidirectional channels that carry the address/control and data. Each channel uses a two-way valid and ready handshake mechanism. The five independents channels are the Address Read (shortening, AR) channel, Address Write (AW) channel, Read Data (RD) channel, Write Data (WD) channel, and write response channel (B). AW and AR channel convey the address and control of write and read transactions. The control signals of such channels describe the nature of the read and write transactions. A transaction can be a burst of a different length, or it can be atomic. A burst is composed of a number of transfers whose length is defined. The data is transferred between master and slave port, and vice versa using WD and RD channels respectively. Write response channel (B) allows a salve to signal completion of the write transaction or an error. One of the features of AXI bus is a burst transaction with only the start address issued. The split transaction AXI protocol enables out-of-order transaction completion; it provides a “transaction ID” field assigned to each transaction. Transactions from the same master port, but with different IDs have no ordering restriction while transactions with the same ID must be completed in order. The AXI protocol enables address information to be issued ahead of the actual data transfer, supports for multiple outstanding transactions supports for out-of-order completion of transactions. Out-of-order transactions completion improves system performance it allows a complex slave like memory return data out-of order. Out of order execution and interleaving are the two main features of AXI bus that provide high throughput.

The MAC (Media Access Control) module enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2005 standard. The MAC IP contains 4 modules: AXI Application Host Interface, DMA controller, MTL/MAC Transaction Layer, MAC core.

The AXI Application Host Interface transfers data to system memory through the AXI master interface. The host CPU uses the AXI Slave interface to access the GMAC subsystem’s Control and Status registers (CSRs), The AXI bus interface provides the characteristics to support highly effective data traffic throughput. The system bus utilization is maximized by allowing simultaneous read and write transfers initiated from read or write DMA channel.

The DMA controller has independent transmit and receive engines, and a CSR space. The transmit engine transfers data from system memory to the device port (MTL), while the receive engine transfers data from the device port to the system memory. The controller use decriptors to efficiently move data from source to destination with minimal host CPU intervention. The DMA is designed for packet-oriented data transfers such as frames in Ethernet. The controller can be programmed to interrupt the host CPU for situations such as frame transmit and receive transfer

International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 1, Issue 2, November 2014)

completion, and other normal/error conditions. The host driver originates DMA communication through control status registers (CSR). The host driver sets OMR (Operation Mode Register OMR) to build DMA operation mode and start DMA.

The MAC core supports 3 interfaces towards the PHY chip. The PHY interface can be configured. The MAC core executes and realizes the IEEE 802.3/802.3z specifications, supports PHY interface GMII (Gigabit Media Independent Interface, GMII)/MII (Media Independent Interface, MII)/RGMII (Reduced GMII, RGMII)/RMII (Reduced MII, RMII). There are 8 types of transfer mode for MAC according to MII/GMII/RMII and RGMII PHY interface in the AMBA IO system.

Mode	Work Frequency	Data Width	Rate
MII	25MHz/2.5MHz	4	10/100Mbps
GMII	125MHZ	8	1000Mbps
RMII	50MHZ	2	10/100Mbps
RGMII	125/25/2.5MHz	4	10/100/1000 Mbps

Table1 Main differences among MII, GMII, RMII and RGMII PHY interfaces

III. VERIFICATION PLATFORM

System verification requires a comprehensive scientific evaluation system AMBA IO system verification places emphasis upon function correctness when lots of different modules integrated in system. Function verification includes interconnect ,timing ,function right or wrong, AXI bus run accord with AXI protocol, MAC transmits or receives data frame correctly in 8 modes. As illustrated in 3, a scalable layered verification platform is designed using system Verilog, which achieves the constraint-random stimulus generation, assertion monitor, perfects the test stimulates and control the progress of verification automatically.

In this paper, the VMM verification methodology is used to build a scalable layered verification platform based system Verilog and sufficient test cases are designed for the DUT to cover all possible boundary conditions. Through the run script to manage and control the entire verification environment, which achieves the constraint-random stimulus generation, assertion monitor, perfects the test stimulates automatically.

The directed test is not entirely adaptive to the complex AMBA IO system, because it is very time-consuming to write all test stimulus manually. Thereby, it is necessary to develop more advanced methodology to speed up the verification process. Many methods have been presented, such as advanced verification methodology (AVM), UVM, and VMM. The verification platform in this paper is built with VMM method and System Verilog language. The Verification Methodology Manual for System Verilog (VMM) describes the framework for developing re-usable verification components and test bench verification environment that provides for higher productivity and enables interoperability. In generic terminology, the VMM consists of coding guidelines and a set of base classes. The VMM book documents advanced functional verification techniques used by industry experts to validate complex SoCs.

On this basis, Verification platform is designed and carried out every subassembly of the platform in the form of planning and implementation based on VMM. At the same time, a coverage-driven combined method of assertion verification has been introduced to the platform. So we are able to perfect the test stimulates, control the progress of verification by means of analyzing code coverage, functional coverage and assertion coverage. In addition, since the properties described by assertions can be reused, the assertion checker designed in platform can be applied to verify other modules in system. Verification results show that there are several characters in verification platform that are designed for AMBAIO system, such as the clearness of structure and arrangement, the rational use of verification platform can saved time greatly of debugging and reduced verification cycles. If other functional modules need to be verify in system besides MAC, only a small amount of modification is necessary to create a new test. It effectively enhances the confidence of the design verification process.

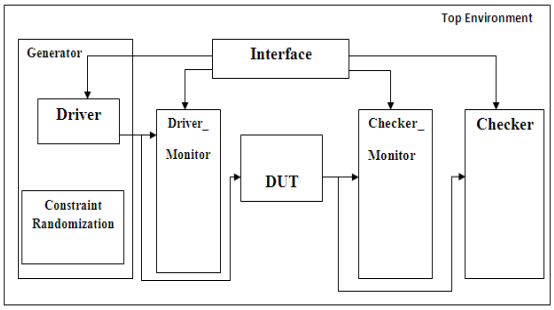


Figure 4.Verification Environment using System Verilog

AMBA IO verification platform is placed in a layer and built in a layered format and architecture, illustrated as figure 3. The major benefit is that the communication mechanisms or channels of communication can independently observe and manage the flow of traffic and data between layers and each modular layer can be designed and tested to its appropriate abstraction level. Test layer generates stimulus, transfers it to the design through drivers; and monitors the response and checks it through the self- checking components as well as provide coverage feedback., the various objects in the verification platform, which are the VMM base class or extension class instantiation. The VMM base class include such as vmm_scenario_gen, vmm_xactor,

International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 1, Issue 2, November 2014)

vmm_data, vmm_env, vmm_channel, and vmm_xactor_callbacks and so on, it is effective to build verification platform by using them. Verification is a major work in order to making sure the design meets the requirements set out by the architects and designers. Armed with product specification which leads into functional specification/architecture of the design, verification planning. Based on the plan and architecture the verification environment is drawn up, with appropriate components, such as transactors, data models in transactions, scoreboards, monitors and coverage gathering mechanisms.

Verification plan: Verification plan summarizes more than 20 test function points according with product specification of AMBA IO system, which includes AXI bus protocol test, MAC controller register access, MAC reset and initial test, single frame transmit test, single frame receive test, multi frame transmit test, multi frame receive test, multi frame parallel transmit and receive test, MAC interrupt test, multi frame transmission in 8 types of PHY mode, etc.. Test sceneries have random test, directed test, regression test, DUT design includes RTL code, DC netlist, sdf netlist(Standard Delay File netlist).

Automatic verification environment: Design automatic verification environment is very necessary for AMBA IO system must run a lot of stimulus. UNIX Makefile tool is used to automate verification process The Makefile is a file which holds the commands which control how to compile or recompile, and link, execute large numbers of programs. 'Compile' command generates executable file based on command options, 'run' command executes object files, generates relevant records, wave files, log files and places classified directory. It is convenient to debug, observe verification, 'make test_*' command finish executing appointed test case. 'make test_regress' command finish executing all regression test cases.

Constrained Random test generation: Random test generation allows finding bugs which they may not have thought about and which may be hard to detect with directed and manual testing. Functional coverage capabilities built in to the press test environment allows the random simulation to quickly converge and identify areas that have not been tested. For example, Ethernet data frames must follow IEEE 802.3-compliant frame standard, its definition applies VMM base class 'vmm_dat'.

Class ethernet_transaction extends vmm_data;
Static vmm_log log = new ("Ethernet Trans", "class");
//four types data frame: general, VLAN, control, jumbo frame
typedef enum {UNTAGGED,TAGGED,CONTROL,JUMBO} frame_type_enum;

Many data frame definition limit its data domain, for example, destination field of flow control frame needs to be 48'h0180_c200_000l Length_Type field is 16'h8808, Opcode field is 16'h0001, other fields can be random, using System Verilog keywords 'constraint'

frame type field of general data frame is constrained 'UNTAGGED', for one AXI transfer byte is a cache line, cache line length is 64 byte, length will not exceed 1536 bytes, Len_type defines byte length of valid load, it is constrained 46,54,114,178,252,306,370 or 434bytes.

Assertion checker: Assertion checkers are a mechanism used to enforce design rules by trapping undesirable behavior during the verification process (e.g., check for an illegal event or an invalid design assumption). Assertions, like events, can be classified as either static or temporal. A static assertion can be implemented in a manner similar to that used with an event monitor, which combines assertion checkers, coverage analysis and pseudo-random test generation. Measuring the effectiveness of their various bug detection mechanisms revealed that 34% of all design bugs were identified through assertion checkers (compared with 11% of bugs identified with self-checking directed test). Likewise, Taylor revealed that assertion checkers identified 25% of their total design bugs. Clearly, a verification process that includes event monitors and assertion checkers will quickly identify and isolate errors while improving observability required for functional coverage analysis.

AMBA IO system verification uses system Verilog assertion check AXI bus protocol, MAC PHY timing, boundary, etc.

For example, an AXI bus protocol checker is when AXI master handshake process read bursts happen, all outstanding read burst complete last data transfer, rlast_m_i change form high to low, and rvalid_m_i is asserted low after next cycle, AXI clock is defined as ack_i, then m_rdata_last can be accessed.

AXI master read address channel checks ARADDR remains stable when ARVALID asserted and ARREADY is high. Then m_araddr can be accessed.

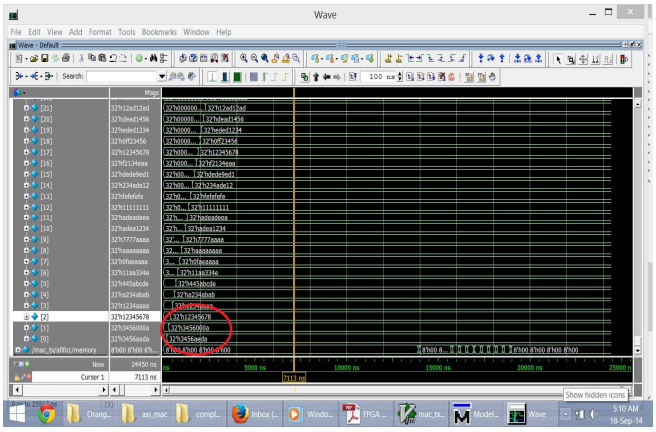
Boundary check is register address space cannot cross boundary, for instance, defined MAC register space is from 32'h32'h0101_0000 to 32'h0101_ffff, AXI slave port write address boundary checker is awaddr remains stable in scope of 32'h32'h0101_0000 to 32'h0101_ffff, when AWVALID and AWREADY is high. Then scope detection can be checked.

Key word assert in system Verilog is to specify the property as an obligation for the design that is to be checked to verify that the property holds. Key word cover is to monitor the property evaluation for coverage. The immediate cover statement specifies that successful evaluation of its expression is a coverage goal. Tools shall collect coverage information and report the results at the end of simulation or on demand via an assertion. The results of coverage for an immediate cover statement shall contain Number of times evaluated and Number of times succeeded. Using assertion coverage can measure how often these assertions are triggered during a test, coverage information can also measure function verification evolution.

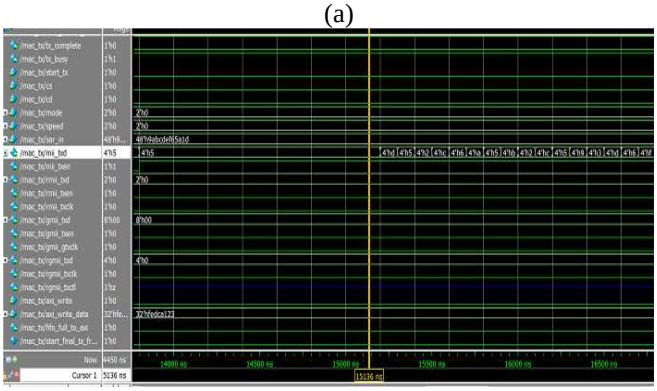
International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 1, Issue 2, November 2014)

IV. EXPERIMENTAL RESULTS

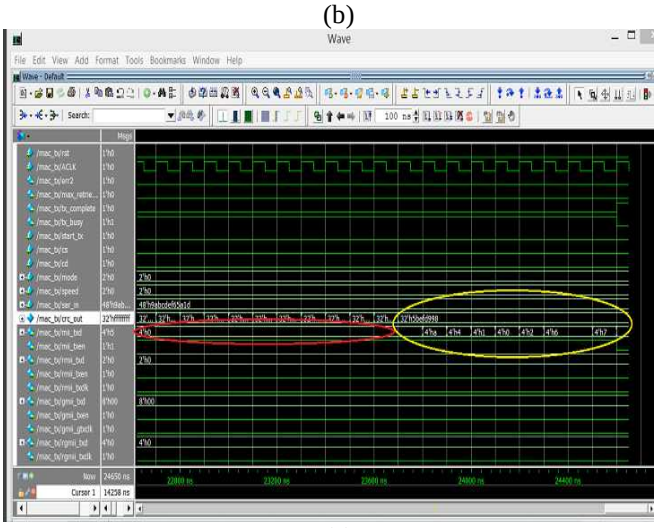
The paper designed and verified of a MAC controller based on AXI bus



(a)



(b)



(c)

Figure 5.a) Data input to MAC Controller fig. b) The Output of MAC Controller Fig. c) Padding & CRC of the Ethernet frame.

V. CONCLUSION

This project presents a design and verification of configurable 10/100/1000Mbps Ethernet MAC controller based on AXI bus including the promiscuous mode in the receiver section to receive all the frames in order to monitor the channel and diagnose the connectivity issues of the media. This design can support the RMII along with MII/GMII/RGMII PHY interfaces. The verification is done by System Verilog hardware description and verification language.

REFERENCES

[1] Jerraya, W. Wolf. Multiprocessor Systems-on-Chips [M], 2007

[2] D.Flynn,AMBA:enabling reusable on-chip designs[J].IEEE Micro Magazine,July- Aug.1997.Volume:17 [3] R.Hofman and B.Drerup.Next generation CoreConnect processor local bus archit-ectture[A],Annual IEEE International ASIC/SOC Conference[C],2002: 221-225.

[3] Silicore.http://www.silicore.net

[4] Avalon Bus Specification Reference Manual[S/OL] . :http://www.altera.com. 2003,7.

[5] D.Wingard.MircoNetwork-based intergration for SOCs[A].In:Design Automation Conference,2001.Proceedings[C].2001:673-677.

[6] B,Cordan,An efficient bus architecture for system on-chip design[A].In: Custom Integrated Circuits, Proceedings of the IEEE 1999[C].1999:623-626.

[7] Open Core protocol specification[S/OL].: http://lad.dsc.ufcg.edu.br/ip/OCP-IP- _Open Core Protocol Specification-1.0.pdf

About the authors:

MS. ANITHA PENDYALA received B.Tech degree in ECE from JNTUH in 2012, pursuing M.Tech (2012-2014) in the stream of VLSI from the ASTRA, JNTU-Hyderabad..

Mr. A.Anendhar obtained his B.Tech in Electronics and Communication Engineering from the University of JNTU, his M.Tech in VLSI System Design from the ASTRA, University of JNTU-Hyderabad. He is currently the Assistant Professor in Department of Electronics &Communication Engineering, ASTRA in University of the JNTUH Hyderabad.

Mr. A.Ravichandra obtained his B.Tech in Electronics and Communication Engineering from the University of JNTU, his M.Tech in VLSI and Embedded System Design from the , University of JNTU-Kakinada. He is currently the Sr. Assistant Professor in Department of Electronics &Communication Engineering, ASTRA in University of the JNTUH-Hyderabad.