ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of High Speed AMBA Advanced Peripheral Bus Master Data Transfer for Microcontroller

Ch.Krishnam Raju
M.Tech (ES)
Department of ECE
Jogaiah Institute of Technology and Sciences,
Kalagampudi, Palakol - 534 268,
West Godavari Dt., A.P.

Mr. B. Kanna Vijay, M.Tech
Assistant Professor
Department of ECE
Jogaiah Institute of Technology and Sciences,
Kalagampudi, Palakol - 534 268,
West Godavari Dt., A.P.

ABSTRACT

The Advanced Microcontroller Bus Architecture (AMBA) is a widely used interconnection standard for System on Chip (SoC) design. An AMBA-based microcontroller typically consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB), able to sustain the external memory bandwidth, on which the CPU, on-chip memory and other Direct Memory Access (DMA) devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. This paper present three distinct buses and their comparison. By considering merits of APB, AMBA can be design by using HDL.

INTRODUCTION

Introduction about AMBA Protocol

The Advanced Microcontroller Bus Architecture (AMBA) was introduced by ARM Ltd in 1996 and is widely used as the on-chip bus in System-on-a-chip (SoC) design shown in Fig 1.1. AMBA is a registered trademark of ARM Ltd. The first AMBA buses were Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). In its 2nd version, AMBA 2, ARM added AMBA High-performance Bus (AHB) that is a single clock-edge protocol.

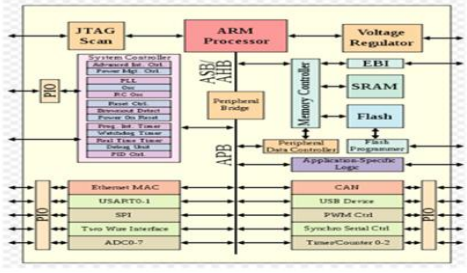

Fig.1. ARM soc Block Diagram

In 2003, ARM introduced the 3rd generation, AMBA 3, including AXI to reach even higher performance interconnects and the Advanced Trace Bus (ATB) as part of the CoreSight on-chip debugs and trace solution.

These protocols are today the de-facto standard for 32-bit embedded processors because they are well documented and can be used without royalties. Some manufacturers utilize AMBA buses for non-ARM designs. As an example Infineon uses an AMBA bus for the ADM5120 SoC based on the MIPS architecture.

The important aspect of a SoC is not only which components or blocks it houses, but also how they are interconnected. AMBA is a solution for the blocks to interface with each other.

Since its inception, the scope of AMBA has gone far beyond microcontroller devices, and is now widely used on a range of ASIC and SoC parts including applications processors used in modern portable mobile devices like smartphones.

Advanced High performance Bus (AHB) Protocol

The advanced microcontroller bus architecture (AMBA) Specification defines an On-Chip Communications standard for designing high performance embedded microcontrollers[1].

Three distinct buses are defined within the AMBA specification

- The Advanced High-Performance Bus (AHB)

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- The Advanced System Bus (ASB)
- The Advanced Peripheral Bus (APB)

A test methodology is included with the AMBA specification which provides an infrastructure for modular test and diagnostic access.

Advanced High Performance Bus (AHB)

The AMBA AHB is for high-performance, high clock frequency system modules. The AHB acts as the high-performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro cell functions. AHB is also specified to ensure ease of use in an efficient design flow using synthesis and automated test techniques.

Advanced System Bus (ASB):

The AMBA ASB is for high-performance system modules. AMBA ASB is an alternative system bus suitable for use where the high-performance features of AHB are not required. ASB also supports the efficient Connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro cell functions.

Advanced Peripheral Bus (APB):

The AMBA APB is for low-power peripherals. AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus.

Objectives of the AMBA Specification:

The AMBA specification has been derived to satisfy four key requirements:

- To facilitate the right-first-time development of embedded microcontroller products with one or more CPUs or signal processors.
- To be technology-independent and ensure that highly reusable peripheral and system macro cells can be migrated across a diverse range of IC processes and be appropriate for full-custom, standard cell and gate array technologies.
- To encourage modular system design to improve processor independence, providing a development road-map for advanced cached CPU cores and the development of peripheral libraries.
- To minimize the silicon infrastructure required to support efficient on-chip and off-chip communication for both operation and manufacturing test.

Multi-layer AHB

The on-chip bus plays a key role in the system-on-a-chip (SoC) design by enabling the efficient integration of heterogeneous system components such as CPUs, DSPs, application specific cores, memories, and custom logic[2]. Recently, as the level of design complexity has become higher, SoC designs require a system bus with high bandwidth to perform multiple operations in parallel. To solve the bandwidth problems, there have been several types of high-performance on-chip buses proposed, such as the multilayer AHB (ML-AHB) bus matrix from ARM, the PLB crossbar switch from IBM, and CONMAX from Silicore . Among them, the ML-AHB bus matrix has been widely used in many SoC designs. This is because of the simplicity of the AMBA bus of ARM, which attracts many IP designers, and the good architecture of the AMBA bus for applying embedded systems with low power.

The ML-AHB bus matrix is an interconnection scheme based on the AMBA AHB protocol, which enables parallel access paths between multiple masters and slaves in a system. This is achieved by using a more complex interconnection matrix and gives the benefit of both increased overall bus bandwidth and a more flexible system structure. In particular, the ML-AHB bus matrix uses slave-side arbitration. Slave-side arbitration is different from master-side arbitration in terms of request and grant signals since, in the former, the master merely starts a burst transaction and waits for the slave response to proceed to the next transfer.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Therefore, the unit of arbitration can be a transaction or a transfer. The transaction-based arbiter multiplexes the data transfer based on the burst transaction, and the transfer-based arbiter switches the data transfer based on a single transfer. However, the ML-AHB bus matrix of ARM presents only transfer-based arbitration schemes, i.e., transfer based fixed-priority and round-robin arbitration schemes. This limitation on the arbitration scheme may lead to degradation of the system performance because the arbitration scheme is usually dependent on the application requirements; recent applications are likewise becoming more complex and diverse. By implementing an efficient arbitration scheme, the system performance can be tuned to better suit applications. For a high-performance on-chip bus, several studies related to the arbitration scheme have been proposed, such as table-lookup-based crossbar arbitration, two-level time-division multiplexing (TDM) scheduling, token-ring mechanism , dynamic bus distribution algorithm , and LOTTERYBUS.

However, these approaches employ master-side arbitration. Therefore, they can only control priority policy and also present some limitations when handling the transfer-based arbitration scheme since master-side arbitration uses a centralized arbiter. In contrast, it is possible to deal with the transfer-based arbitration scheme as well as the transaction- based arbitration scheme in slave-side arbitration. In this paper, we propose a flexible arbiter based on the self-motivated (SM) arbitration scheme for the ML-AHB bus matrix[2]. Our SM arbitration scheme has the following advantages: 1) It can adjust the processed data unit; 2) it changes the priority policies during runtime; and 3) it is easy to tune the arbitration scheme according to the characteristics of the target application. Hence, our arbiter is able to not only deal with the transfer-based fixed-priority, round-robin, and dynamic-priority arbitration schemes but also manage the transaction-based fixed-priority, round-robin, and dynamic-priority arbitration schemes. Furthermore, our arbiter provides the desired-transfer-length-based fixed-priority, round-robin, and dynamic-priority arbitration schemes. In addition, the proposed SM arbiter selects one of the nine possible arbitration schemes based on the priority-level notifications and the desired transfer length from the masters to ensure that the arbitration leads to the maximum performance.

Multi-layer AHB is an interconnection scheme, based on the AHB protocol, which enables parallel access paths between multiple masters and slaves in a system. This is achieved by using a more complex interconnection matrix.

Key advantages are:

- You can develop multi-master systems with an increased available bus bandwidth.
- You can construct complex multi-master systems that have a flexible architecture. This removes the requirement to fix design decisions about the allocation of system resources to particular masters at the hardware design stage.
- Each AHB layer can be very simple because it only has one master, so no arbitration or master-to-slave muxing is required. These layers can use the AHB-Lite protocol, meaning that they do not have to support request and grant, or retry and split transactions.
- The arbitration effectively becomes point arbitration at each peripheral and is only necessary when more than one master wants to access the same slave simultaneously..
- Because the multi-layer architecture is based on the existing AHB protocol, shown in Figure.1.3 you can reuse previously-designed masters and slaves without modification

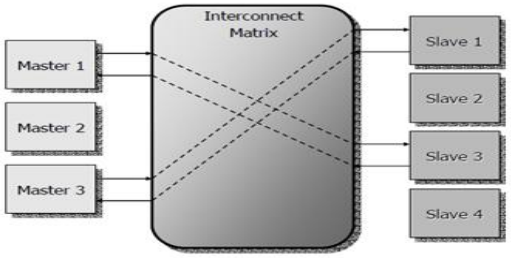

Fig .2. Basic multi-layer concept

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

APPLICATIONS

AMBA-AHB can be used in the different application and also it is technology independent.

- ARM Controllers are designed according to the specifications of AMBA. In the present technology, high performance and speed are required which are convincingly met by AMBA-AHB Compared to the other architectures AMBA-AHB is far more advanced and efficient.
- To minimize the silicon infrastructure to support on-chip and off-chip communications
- Any embedded project which involve in ARM processors or microcontroller must always make use of this AMBA-AHB as the common bus throughout the project.

Block Diagram

The block diagram of the Advanced High-Performance Bus Protocol is shown in the Figure 2.1. Totally this block diagram comprises of four components.

- Arbiter
- Master
- Slave
- Decoder

Arbiter

The arbitration mechanism is used to ensure that only one master has access to the bus at any one time. The arbiter performs this function by observing a number of different requests to use the bus and deciding which is currently the highest priority master requesting the bus.

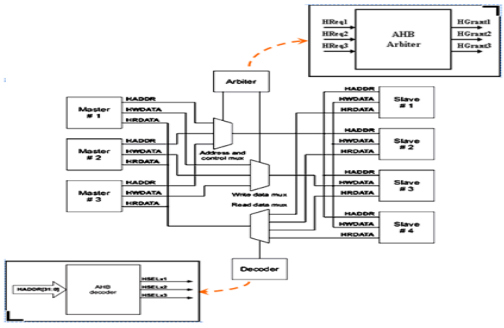

Fig.2.1 AMBA – AHB block diagram

Master

A bus master is able to initiate read and write information by providing address and control information. Only one bus master can use the bus at the same time An AHB bus master has the most complex bus interface in an AMBA system.

Typically an AMBA system designer would use predesigned bus masters and therefore would not need to be concerned with the detail of the bus master interface. No provision is made within the AHB specification for a bus master to cancel a transfer once it has commenced.

Slave

After a master has started a transfer, the slave then determines how the transfer should progress. Whenever a slave is accessed it must provide a response which indicates the status of the transfer. The HREADY signal is used to extend the transfer and this works in combination with the response signal HRESP which provide the status of the transfer.

The slave can complete the transfer in a number of ways. It can:

- Complete the transfer immediately
- Signal an error to indicate that the transfer has failed
- Delay the completion of the transfer, but allow the master and slave to back off the bus, leaving it available for other transfers.

Decoder

The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A central address decoder is used to provide a select signal ‘HSELx’ for each slave on the bus. The select signal is a combinatorial decode of the high-order address signals. A slave must only sample the address and control signals and HSELx is asserted when HREADY is HIGH, indicating that the current transfer is completing.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Working of AHB

The AMBA AHB bus protocol is designed with a central multiplexor interconnection scheme.

Using this scheme all bus masters drive out the address and control signals indicating the transfer, they wish to perform and the arbiter determines which master has its address and control signals routed to all of the slaves. Before which initially the master who needs to perform the operation should give the request signal to the arbiter and the arbiter will give the grant signal to the master for further proceedings. Similarly, a decoder is used to select the slave which has to be active during the operation based on the address given by the master. A central decoder is also required to control the read data and response signal multiplexor, which selects the appropriate signals from the slave that is involved in the transfer. These make the read and write operation smoothly.

Thus the working of AMBA AHB protocol is explained with the help of its block diagram shown in Figure .2.1.

Overview of AMBA AHB Operation

Before an AMBA AHB transfer can commence the bus master must be granted access to the bus. This process is started by the master asserting a request signal to the arbiter. Then the arbiter indicates when the master will be granted use of the bus.

A granted bus master starts an AMBA AHB transfer by driving the address and control signals. These signals provide information on the address, direction and width of the transfer, as well as an indication if the transfer forms part of a burst. Two different forms of burst transfers are allowed.

- Incrementing bursts, which do not wrap at address boundaries
- Wrapping bursts, which wrap at particular address boundaries

A write data bus is used to move data from the master to a slave, while a read data bus is used to move data from a slave to the master.

Every transfer consists of:

- An address and control cycle
- One or more cycles for the data.

The address cannot be extended and therefore all slaves must sample the address during this time. The data, however, can be extended using the HREADY signal. When LOW this signal causes wait states to be inserted into the transfer and allows extra time for the slave to provide or sample data.

During a transfer the slave shows the status using the response signals, HRESP OKAY. The OKAY response is used to indicate that the transfer is progressing normally and when HREADY goes HIGH this shows the transfer has completed successfully.

Design of Self-Motivated Arbitration Scheme for the Multilayer AHB Bus matrix

The ML-AHB bus matrix of ARM consists of the input stage, decoder, and output stage, including an arbiter Figure.3.1 shows the overall structure of the ML-AHB bus matrix of ARM.

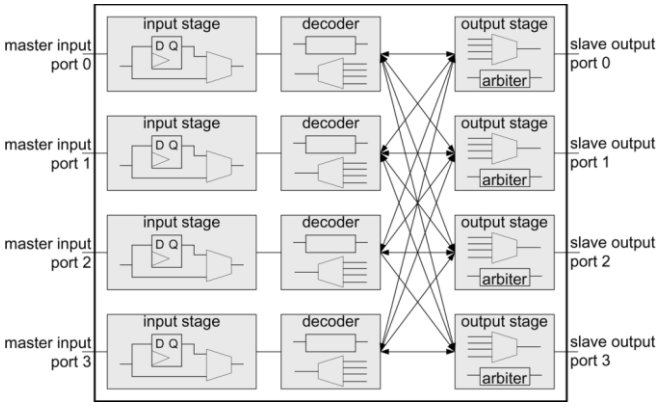

Fig.3.1 Overall structure of the ML-AHB bus matrix of ARM

The input stage is responsible for holding the address and control information when transfer to a slave is not able to commence immediately. The decoder determines which slave that a transfer is destined for. The output stage is used to select which of the various master input ports is routed to the slave. Each output stage has an arbiter. The arbiter determines which input stage has to perform a transfer to the slave and

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

decides which the highest priority is currently. The ML-AHB bus matrix employs slave-side arbitration, in which the arbiters are located in front of each slave port, as shown in Fig.3.1; the master simply starts a transaction and waits for the slave response to proceed to the next transfer. Therefore, the unit of arbitration can be a transaction or a transfer. However, the ML-AHB bus matrix of ARM furnishes only transfer-based arbitration schemes, specifically transfer-based fixed-priority and round-robin arbitration schemes. The transfer-based fixed-priority (round-robin) arbiter multiplexes the data transfer based on a single transfer in a fixed-priority or round-robin fashion.

PROPOSED METHODOLOGY

The ARM processor and AHB bus are more popular in SoC design. Bus has many advantages in high-performance devices but has limitations on interface with low bandwidth devices. So, bridge is necessary between them. AMBA AHB and ASB is high performance bus and have higher bandwidth. High bandwidth RAM, DMA bus controller, memory interface and high performance ARM processors which require high bandwidth are connected with AHB or ASB. Whereas APB is low bandwidth and low performance bus. Peripheral devices such as UART, Timer, and keypad require low bandwidth, so connects with APB. So bridge is required to connect AHB or ASB and APB.

AHB-to-APB Bridge interfaces AHB and APB. It is require to bridge communication gap between high bandwidth AHB and low bandwidth peripheral like serial, Ethernet devices on APB .

There are many differences between these two buses. AHB uses full duplex communication where as APB uses massive memory-I/O accesses. Unlike AHB, there is no pipelined structure in APB. Compared with AHB, APB has low bandwidth control accesses.

If comparing usage, APB is simpler than AHB. APB is mainly proposed for connecting to simple low bandwidth peripherals. APB also can be optimized for reduce the interface complexity and power consumption.

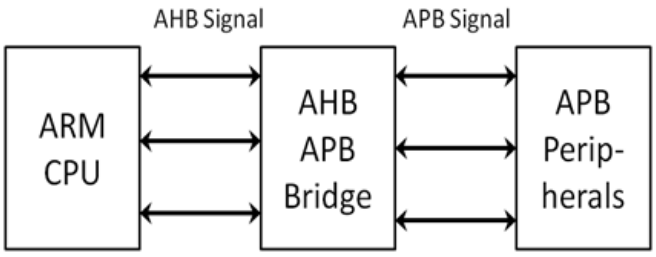

Fig.4.1 . Application Block Diagram of AHB to APB

Figure. 4.1, shows application block diagram for AHB to APB Bridge. The master is ARM CPU and slaves are APB peripherals. It is assumed that speed ratio of AHB2APB is 2:1 i.e. if AHB executes with clock frequency 200MHz, APB should be with 100MHz. So, Bridge is required for frequency and speed compensation between system and local bus.

DESIGN OF AHB to APB BRIDGE

AHB to APB Bridge operates on HCLK and APB access sub module operates on PCLK. AHB response and Control transfer is together termed as AHB interface and APB access is termed as APB interface to ensure the correct generation of suitable control signals and address we use three internal signals in the bridge module namely:

1) PENDWR (Pending Write).
2) PENDRD (Pending Read)
3) PDONE (Peripheral operation done).

The capture of address & control for Write or Read operation is done when HREADY, HTRANS and HSEL are valid. READY is the only signal that is the output from the bridge to AHB master to cope up the communication between AHB and APB. Hence the generation of HREADY signal is very significant in the bridge module. By using the internal signals PENDWR and PENDRD and double synchronized signal (Double synchronization is explained later in this section) PDONE, HREADY generation is made easy to capture the next control for Write or Read operation from AHB to APB. Since the sub modules

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

operate on different clock domains namely HCLK and PCLK, there is a need for interfacing these clock domains. Any two systems are considered asynchronous to each other:

- When they operate at two different frequency.
- When they operate at same frequency, but at two different clock phase angles.

This interfacing is difficult in the sense that design becomes asynchronous at the boundary of interface, which results in setup and hold time violation, Meta stability and unreliable data transfers. Hence we need to go out for special design and interfacing techniques. In such a case if we need to do data transfer, there are very few methods to achieve this namely:

- Handshake signaling method.
- Asynchronous FIFO.

Both have its own advantages and disadvantages. In our paper we have used Handshake signaling Method. In Handshake signaling method the AHB interface sends data to APB interface based on the handshake signals PENDWR (or PENDRD) and PDONE signals. The protocol for this uses the same method that is found with 8155 chip used with 8085 based on handshake signals Request and Acknowledge

Operation of AHB to APB Bridge

The AHB to APB interfaces AHB and APB. It buffers address, controls and data from the AHB, drives the APB peripherals and return data along with response signal to the AHB. The AHB to APB interface is designed to operate when AHB and APB clocks have the any combination of frequency and phase .The AHB to APB performs transfer of data from AHB to APB for write cycle and APB to AHB for Read cycle.

Features of AHB to APB Bridge

Interface between AMBA high performance bus (AHB) and AMBA peripheral bus (APB) , provides latching of address, controls and data signals for APB peripherals.

Supports for the following

- APB compliant slaves and peripherals.
- Peripherals which require additional wait states.

RESULTS

Figure 8.1 shows the Simulation Result for AMBA APB MASTER BURST. Back annotation is the translation of a routed or fitted design to a timing simulation Net list. Back annotation was performed on the Xilinx generated synthesis file for the AHB2APB Bridge module. In our paper only Bridge module is the synthesized module and AHB driver and APB monitor were test bench modules.

1) With HCLK and PCLK having a ratio of 1:2.

2) With HCLK and PCLK having a phase difference of 900 and same frequency

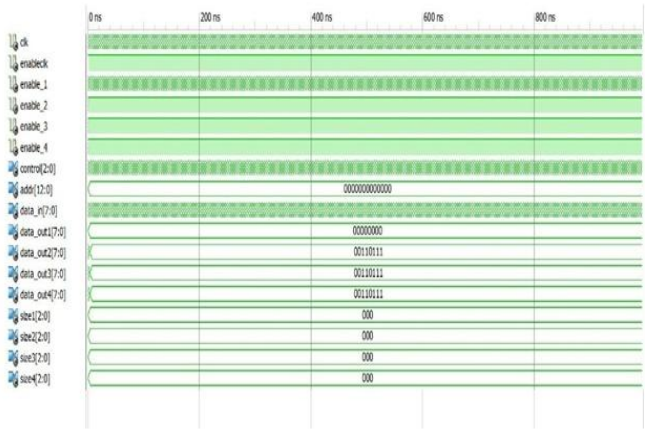

Fig.8.1 Simulation Result for AMBA APB MASTER SLAVE BURST

CONCLUSION

In this paper, the proposed RTL Simulation of AHB2APB Bridge has been verified and validated by using suitable test benches namely AHB Driver/Monitor and APB Driver/Monitor.

The Synthesis of AHB2APB Bridge has been successfully completed by the extraction of Synthesized Netlist with unit delays & verified by comparing the Gate level Simulation with RTL Simulation results. The Back Annotation of AHB2APB Bridge has also been successfully completed by the extraction of Synthesized Netlist with suitable delays & verified by the comparison of Gate level simulation with RTL simulation results.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

AMBA APB provides the basic peripheral macro cell communications infrastruc-ture as a secondary bus from the higher bandwidth pipe-lined main system bus .Such peripherals typically have interfaces which are memory-mapped registers,have no high-bandwidth interfaces and these are accessed under programmed control.

Thus AHB2APB Bridge is a standalone solution to extract the advantages of newly developed ARM based AMBA AHB bus by bridging the common gap between AHB and the existing APB bus.

FUTURE SCOPE

As future work

- This work can be improved by implementing with Advanced Risc Machines(ARM) Processors
- This Design Can also be used in all SOC's Applications where debugging and performance analysis is difficult.
- This work can be improved by implementing with video codec controller.

REFERENCES

[1] M. Drinic, D. Kirovski, S. Megerian, and M. Potkonjak, "Latencyguided onchip bus-network design," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 25, no. 12, pp. 2663–2673, Dec. 2006.

[2] S. Y. Hwang, K. S. Jhang, H. J. Park, Y. H. Bae, and H. J. Cho, "An ameliorated design method of ML-AHB busmatrix," ETRI J., vol. 28, no. 3, pp. 397–400, Jun. 2006.

[3] ARM, "AHB Example AMBA System," 2001 [Online]. Available: http://www.arm.com/products/solutions/AMBA_Spec.html

[4] Akhilesh Kumar, Richa Sinha, "Design and Verification analysis of APB3 Protocol with Coverage," IJAET, Nov 2011

[5] R. Usselmann, "WISHBONE interconnect matrix IP core," Open- Cores, 2002. [Online].

[6] N.-J. Kim and H.-J. Lee, "Design of AMBA wrappers for multipleclock operations," in Proc. Int. Conf. ICCCAS, Jun. 2004, vol. 2, pp. 1438–1442.

[7] SanthiPriya Sarekokku, K. Rajasekhar, "Design and Implementation Of APB Bridge based on AMBA AXI 4.0," IJERT, Vol.1, Issue 9, Nov 2012.

[8] S. S. Kallakuri and A. Doboli, "Customization of arbitration policies and buffer space distribution using continuous-time Markov decision processes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 2, pp. 240–245, Feb. 2007.

[9] Samir Palnitkar, "Verilog HDL: A guide to Digital Design and Synthesis (2nd Edition), Pearson, 2008.

[10] Chris Spear, "SystemVerilog for verification (2nd Edition): A guide to learning the testbench features, Springer, 2008..

WEBSITES

[1] http://www.arm.com/products/solutions/axi_spec.html,

[2] http://www.opencores.org/?do=project=wb_conmax

[3] www.arteris.com/arm-amba-protocols-axi-ac

[4] www.dauniv.ac.in/downloads/...PPTs/Chap_3Lesson23Emsys Ne

[5] http://wwwmicro.deis.unibo.it/~magagni/amba99.pdf