International Conference on Intelligent Sustainable Systems (ICISS 2019)
IEEE Xplore Part Number: CFP19M19-ART; ISBN: 978-1-5386-7799-5

Effective Design and Implementation of AMBA AHB Bus Protocol using Verilog

Deeksha L
Dept. of ECE
NMAM Institute of Technolgy, Nitte
Karkala, India
deekshal557@gmail.com

Shivakumar B.R
Dept. of ECE
NMAM Institute of Technolgy, Nitte
Karkala, India
shivkumarbr@gmail.com

***Abstract*— The computer's performance is intensely reliant on bus interconnect design. An ineffectively designed system bus can interfere with the transmission of guidelines and the information between the memory and processor or between the peripheral gadgets and memory. The Advanced Microcontroller Bus Architecture (AMBA) specification defines high-performance microcontroller on-chip communication standards. AHB is part of the AMBA convention family. The AMBA AHB is designed for high-performance, high-clock system modules. The AHB acts as the system's high-performance backbone bus. AHB supports the efficient connection of low-power macro cell peripheral functions to processors, on-chip memories and external off-chip memory interfaces. All signal transitions relate only to the rising clock edge, so that AHB peripherals can easily be integrated into any design flow. This paper describes AMBA AHB design and implementation using Verilog. The read/write operation was implemented using Xilinx simulator®.**

Keywords—Advanced Microcontroller Bus Architecture; Advanced High-performance Bus; System-on-Chip; Hardware Description Language;

I. INTRODUCTION

In the modern era of VLSI technology, the processing devices depend on the communication protocol System-on-Chip (SOC). The Advanced Bus Architecture Microcontroller (AMBA) was presented by ARM Ltd in 1996 [1] and is widely used as an on-chip bus in the chip system (SoC) outlines. AMBA is an ARM Ltd registered a trademark. Advanced System Bus (ASB) and Advanced Peripheral Bus (APB) were the main AMBA buses. In its third era, AMBA 3, included AXI, which achieves significantly higher execution interconnections and the Advanced Trace Bus (ATB) as a major aspect of the Core Sight on-chip troubleshooting and trace solution.

AMBA protocols are open standards. It is an interconnecting requirement. It is used to connect and manage functional blocks in a computer system. The AMBA controller converts the incoming signal to a memory controller convention and optimizes the performance [2]. It can be designed for ASIC synthesis using synthesizable HDL. It also supports multiple memory devices and a shared memory data path. It reduces the number of pins.

These conventions are today the true standard for embedded 32-bit processors, as they are all well documented and can be used without priorities [3]. The aim of AMBA is to help embedded system designers to tackle difficulties such as low power design, to encourage the right-of-the-first improvement of embedded microcontroller products with at least one CPU or single processors, to be innovation-free and to support modular systems. To limit the silicon infrastructure, effective on-chip and off-chip communication must be supported for both operation and manufacturing testing.

The research is aimed at the design and implementation of an AMBA AHB using Verilog HDL.

The rest of the paper is organized as follows. Section II presents a literature review on various design techniques of AMBA controller. Section III presents the architecture of the AMBA based microcontroller. The working of AMBA AHB and interfacing of AMBA AHB master and slave are given in section IV and V, respectively. Section VI deals with the simulation results and final view of the paper is concluded in section VII.

II. REVIEW ON VARIOUS DESIGN TECHNIQUES OF AMBA CONTROLLER

Scott Morton et al. [1] has given a detailed discussion on plan and execution elements of AMBA High execution bus (AHB) ace and slave with memory controller interface. This paper also provides the resultant productivity in regard to area overhead and speed. The execution was carried out using VHDL. The design has been integrated on Xilinx 13.1 Spartan3 and re-enacted in ModelSim.

Donna Simon and Guru Prasad [2] proposed the technique for designing and usage of a memory controller utilizing AMBA Bus for image transfer applications. The memory controller is designed utilizing conventional finite state machine and AMBA is actualized on Xilinx. The read composes activities are expert with zero holds up states. This architecture upgrades energy to a 412 mw and design is incorporated on Xilinx 13.1 Spartan3.

Shilpa Rao and Arati.S.Phadke [3] have explained the implementing and testing of the Advanced Microcontroller Bus Architecture (AMBA) consistent Memory Controller as

an Advanced High execution Bus (AHB) slave. The work has been implemented utilizing Verilog, designed to an FPGA belonging to Spartan 3A and Spartan 3AN family utilizing Xilinx compiler, and reproduced with ModelSim The proposed architecture went for upgrading the power and it is discovered that it consumes the energy of 15 mW with a maximum frequency of 114.705 MHz.

Nithin Joe and Anjali Brite [4] have disclosed how to manufacture an Advanced Microcontroller Bus Architecture (AMBA) complaint memory controller as an advanced high-performance bus slave. The simulation result demonstrates that the delay for the read and compose tasks is 4 clock cycles which is less than microcontroller and other memory controllers.

Rishabh Singh Kurmi et al. [5] have proposed the strategy for designing and implementation of adaptable arbiter scheme for the AHB bus matrix based on the burst activity. Multiplexer and Decoders are utilized to chooses the suitable signal between ace and slaves that are involved in the transfer. From the plan, it can be concluded that the minimum period utilized in the design is 57.142 ns and frequency is 17.500 MHz which is comparatively good. Depending on the real-time application it can be utilized for designing the high performance implanted microcontroller.

Jayapraveen and Geetha Priya [6] has disclosed how to build an AMBA Advanced High execution Bus (AHB) based memory controller that can work proficiently in multi-master and multi-slave communication model. The AHB bus can be implemented at RTL level by changing the original AHB memory controller. Utilization of the SDRAM bus is increased. That means the performance can be improved. Based on different application case, a 5% to 15% performance improvement can be achieved. The simulation is built on only two AHB masters work at the same time if more masters added; the bus utilization can be improved from 10-20% for typical multimedia application.

Pravin S. Shete and Dr. Shruti Oza [7] have explained a finite state machine design for AMBA AHB ace and implemented it utilizing Verilog HDL. Advanced microcontroller bus architecture master is the major element of the system. An efficient design of this will present area and time efficient designs of the System on-chip.

Zahid Khan et al. [8] introduced a new on-chip bus encoding system for high-performance SoC generic systems. Including its power efficiency, this scheme also reduces delay failures. It is implemented using the AMBA-AHB SoC bus standard and provides results for energy savings of 24 to 38 percent for systems implemented in CMOS technology of 0.18 μm.

Dr. Fazal Noorbasha et al. [9] proposed the method of designing and implementing the Clock Skew Minimization Technique AMBA APB Bridge. The Ripple counter is a great way to reduce the skew of a clock. Using Verilog HDL, the APB Bridge with clock skew minimization technique is implemented. Vivado Design Suite ISim was used for simulation.

III. ARCHITECTURE OF THE AMBA BASED MICROCONTROLLER

An AMBA-based microcontroller usually consists of a high-performance system bus that supports the CPU's external memory bandwidth, on-chip memory and other DMA devices. AMBA system architecture [2] is shown in Fig 1. This high-performance system bus offers a high bandwidth interface between most transfers associated components.

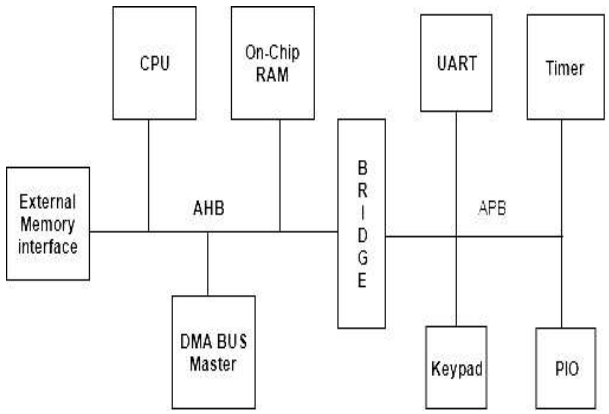

Fig. 1 AMBA based microcontroller [2].

A typical AMBA Advanced High-performance Bus system design the following components [4] :

A. *AHB Master:* A bus ace can begin reading and composing activity with an address and control data. Only one single ace is allowed to use the bus at any instant of time.

B. *AHB Slave:* A bus slave reacts to the reading activity or composes it within a specified address space. The bus slave reacts to the dynamic ace about the data transfer status.

C. *AHB Arbiter:* The bus arbiter guarantees that just a single bus ace at once is permitted to start the information exchange. Although the arbitration convention is fixed, any intervention calculation, such as the highest priority or reasonable access, can be carried out according to the requirements of the application.

D. *AHB Decoder:* An AHB decoder is used to decode the address of each exchange and provides the appropriate fag selection to the exchange slave. In all AHB executions, a single unified decoder is required.

IV. WORKING OF AMBA AHB

The AMBA AHB bus convention is intended to be used with a central interconnection multiplexer system. All bus aces display the address and control signals showing the transfer they want to perform and the arbiter figures showing which ace has its address and control signals for the majority of slaves using this design [6]. In addition, a central decoder is required to control the read and response signal multiplexer, which selects the correct transmission slave signals.

Each exchange includes an address and control cycle as well as at least one information cycle [7]. The address cannot

be expanded and all slaves must therefore, be the example address in the middle of this time. However, the information can be expanded using the hready flag. When this flag is low, the waiting state is integrated into the exchange and allows the slave to give or sample the information for additional time.

The slave shows the status of the data transfer using the response signal *HRESP [1:0]* [5]*:*

OKAY: The OKAY reaction is used to show that the exchange is progressing normally, and when the hready is high it shows that the exchange has been successfully completed.

ERROR: The ERROR reaction shows that an exchange blunder is occurred also, the exchange was unsuccessful. *RETRY* and *SPLIT*: The exchange reactions of both RETRY and SPLIT shows that the exchange cannot finish quickly, but the bus master continues to attempt the transfer.

Each transfer can be ordered into one of four distinct types, as showed by the *HTRANS*[1:0] signals. Table 1 shows the transfer types.

TABLE I. AHB TRANSFER TYPES [10].

HTRANS	*Type*	*description*
00	IDLE	Specifies no transmission is mandatory.
01	BUSY	Indicates that the bus master is continuing with the burst of transfer.
10	NONSEQ	Indicates the first transfer of a burst or single transfer.
11	SEQ	All the rest of the transfer is consecutive. The deliver will be identified with the past exchange.

V. AHB MASTER AND SLAVE INTERFACE

Over recent years, there have been a few procedures completely developed in fields of data exchange.

A. AHB Master: The most complex AMBA bus interface is provided by an AHB bus master. A master initiates the reading / writing operation by providing the arbitrator with the control information and the slave's address [9]. First the master decides which slave would like to communicate. The master requests the bus to the arbiter via the HBUSREQ signal. If the channel is free, the arbitrator grants the request, and the HGRANT signal goes high, which means that the bus has been delivered. The master can now communicate with the slave through the slave's memory address. Read / write operations can be performed. All operations are performed at the positive clock edge. The diagram of the AHB master interface is shown in Fig. 2.

B. AHB Slave: An AHB bus slave responds to transfers made by bus masters inside the system. The slave uses a signal to select the HSELx decoder to determine when to respond to a bus transfer [8]. The bus master generates all other required transmission signals, such as address and control information. The recognition signal used to check the communication was carried out successfully or fails by the slave signal back to master. The data from the respective memory location is read and returned to the master when the read operation is requested. When writing is requested, the location of the memory where the data must be written in the memory occurs. The slave signals to the master every request. The diagram of the AHB salve interface is shown in Fig.3.

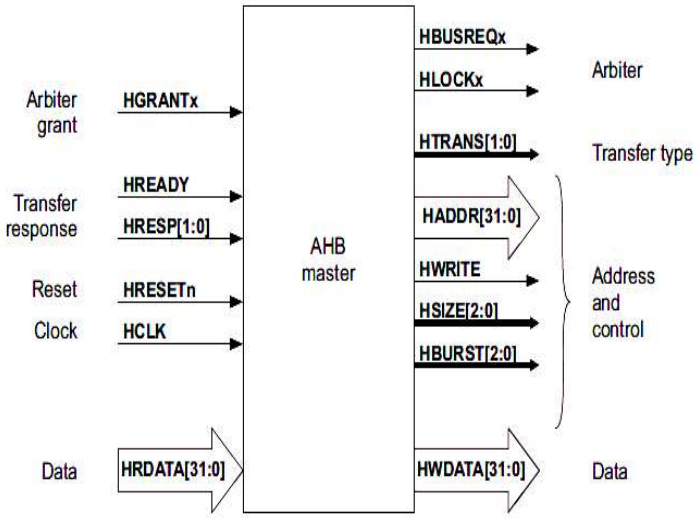

Fig. 2: AHB master interface [10].

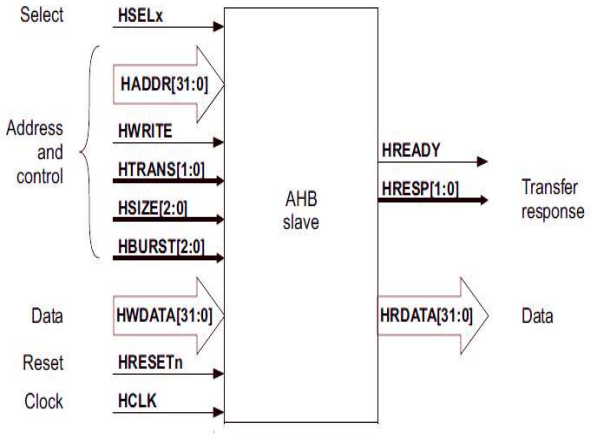

Fig. 3: AHB slave interface [10]

VI. RESULTS AND DISCUSSION

This section presents the results obtained during the course of the study. The pin configuration of the AMBA Controller shown in Fig. 4. Here HCLK is used to get output allocated to the controller, and zero otherwise. HRESTn is used to reset the master to default conditions, BUSREQ is the bus request signal sent by the master to the arbiter , addr_reg is used to store the address, once the bus is granted requested address(ADDREQ) is sent by the master, TRANS is the signal to choose the different types of operation, data required by the slave to write/read will be stored in the register WDATA, HRDATA respectively, RDATA/ HWDATA register will store the output data read or written.

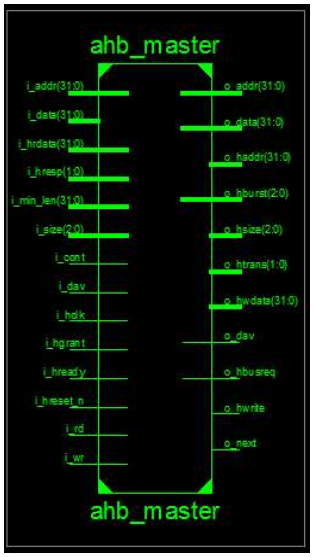

Fig.4: Pin configuration of AMBA controller.

As an illustration, consider the data to be read as *HRDATA*=11111111111111111111111111111111.To complete the reading of the data, following pin configurations are set by the controller: *HRESTn* = 1, *HGRANT* = 1, *!addr_reg* = 1, *ADDREQ* = 1 and *write* = 0. The simulation output for the read operation of the Fig. 5. When the write signal is low and depending on the *trans* value (Table I), data from the *HRDATA* register will be stored in the *RDATA* register.

As an illustration, consider the data to be written as WDATA=11111111111111111000000000000000.To complete the writing of the data, following pin configurations are set by the controller: HRESTn = 1, HGRANT = 1, !addr_reg = 1, ADDREQ = 1 and write = 1. The simulation output for the write operation of the Fig. 6. When the write signal is high and depending on the trans value (Table I), data from the WDATA register will be stored in HWDATA register.

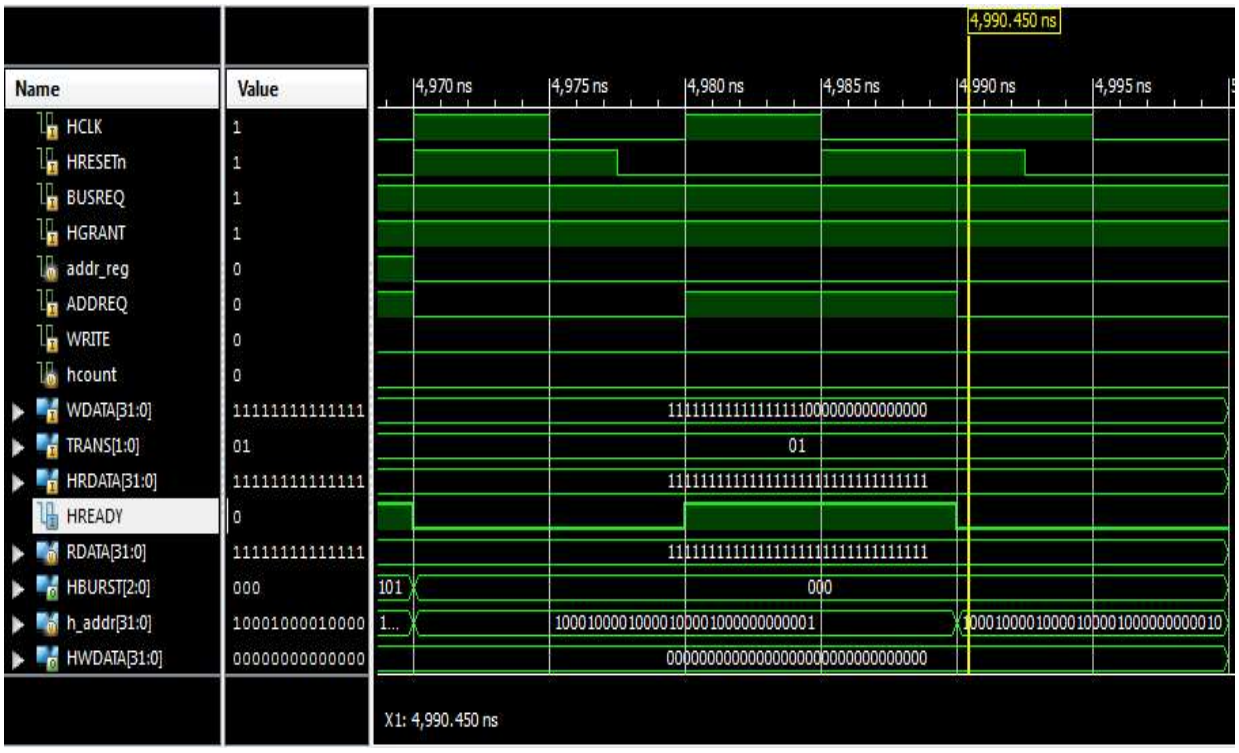

Fig. 5: Simulation waveform of read operation.

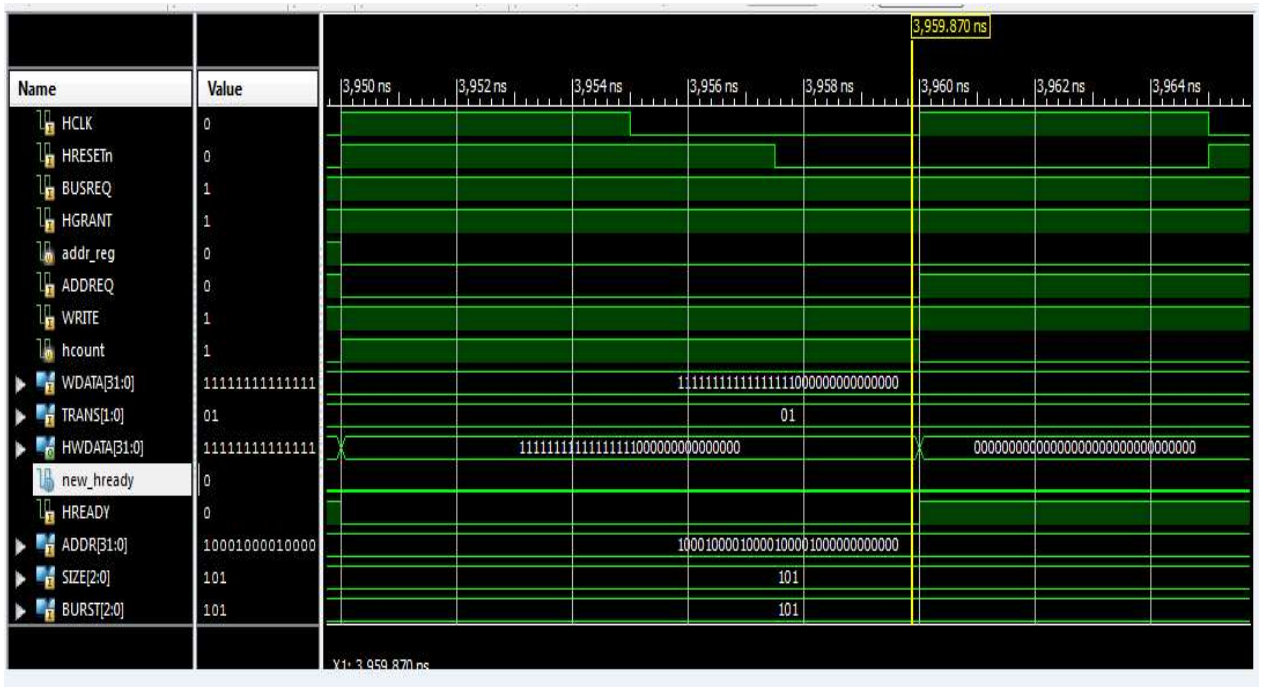

Fig. 6: Simulation waveform of write operation

VII. CONCLUSION

In this paper, efficient design of an AMBA controller is designed and tested for read and write operations using a Xilinx simulator®. The read and write operations using AMBA are illustrated with simple examples. Future scope of the work includes designing AHB based memory controller for image transfer applications.

REFERENCES

[1] S. Ramagundam *et al.*, "Design and implementation of high-performance master/slave memory controller with microcontroller bus architecture," *Conf. Rec. - IEEE Instrum. Meas. Technol. Conf.*, no. May, pp. 10–15, 2014.

[2] D. Simon and U. Guruprasad, "Design and implentation of AMBA-Memory controller for image tansfer applications," *IJRET Int. J. Res. Eng. Technol.*, vol. 5, no. 4, pp. 290–293, 2016.

[3] S. Rao and A. S. Phadke, "Testing of AMBA Compliant Memory Controller using Pattern Generator / Logic Analyser," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 2, no. 6, pp. 2227–2233, 2013.

[4] A. B. Nithin Joe, "D Esign and a Nalysis of a N Ovel D Igital," *Int. J. Adv. Eng. Technol.*, vol. 4, no. 2, pp. 2053–2063, 2013.

[5] R. S. Kurmi and A. Somkuwar, "Design of AHB Protocol Block for Advanced Microcontrollers," *Int. J. Comput. Appl. (0975*, vol. 32, no. 8, pp. 23–29, 2011.

[6] D. Jayapraveen and T. G. Priya, "Design of memory controller based on AMBA AHB protocol," *Elixir Comp. Sci. Engg. 51A*, vol. 2, pp. 11115–11119, 2012.

[7] P. S. Shete and S. Oza, "Design of an Efficient FSM for an Implementation of AMBA AHB Master," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 3, pp. 267–271, 2014.

[8] Z. Khan, T. Arslan, and A. T. Erdogan, "A novel bus encoding scheme from energy and crosstalk efficiency perspective for AMBA based generic SoC systems," *Proc. IEEE Int. Conf. VLSI Des.*, no. January, pp. 751–756, 2005.

[9] M. K. Kumar, A. Sajja, and F. Noorbasha, "Design and FPGA Implementation of AMBA APB Bridge with Clock Skew Minimization Technique," vol. 7, no. 3, pp. 42–45, 2017.

[10] A. R. M. Ihi, "AMBA™ Specification."