***I.J. Engineering and Manufacturing,* 2019, 4, 33-44**
Published Online July 2019 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijem.2019.04.03

Available online at ***http://www.mecs-press.net/ijem***

Memory Controller and Its Interface using AMBA 2.0

Hitanshu Saluja[a] , Naresh Grover[b]

[a,b] Department of Electronics & Communication, Manav Rachna International Institute of Research and Studies, Faridabaad, Hrayana, India

Received: 15 January 2019; Accepted: 25 April 2019; Published: 08 July 2019

Abstract

This paper elaborates the AMBA bus interface bridge between memory controller and other supporting peripheral. The work claims the integration with FIFO, RAM and ROM with slave interface and the master of AHB bus. The AHB master initiates the operation and generates the necessary control signal. Memory controller is implemented with finite state machine considering with all the peripheral works in synchronous mode. Despite these shortcomings of the work performed study and development that followed has led the development of a memory controller on AMBA-AHB bus at a very advanced stage and next to prototyping. VHDL code is utilized to develop the design and it is synthesized in Xilinx Virtex 6 device (XC6VCX75T). The design claims a minor area overhead with improvement in speed 185.134 MHz.

Index Terms: AMBA, AHB Master, AHB Slave, SOC, Xilinx.

1. Introduction

Technological developments in the world industrialization have allowed the integration of more and more functions on the same digital integrated circuit. Today, several microprocessors, hardware accelerators, communications systems diverse and with even operating systems. this allows therefore perform all the functions necessary to perform complex computer processing on the same chip, hence the birth of the concept of systems on chip (System on Chip), successors of specialized circuits ASIC (Application Specific Integrated Circuit) [1].

* Corresponding author.
E-mail address:

The choice of the communication system, in a system On-chip, remains a major problem. This choice depends on components constituting the system and especially the processor used. SoCs design software platforms, proposed by the manufacturers of the FPGA circuits, proposes a well-defined type of communication system [2]. These systems are generally standard buses modeled in one language description of hardware like the VHDL. Recently, with the aim of improving quality and speed communication systems in new systems, networks On-chip (NoC: Network-On-Chip) have been introduced [3]. The principle of NoCs is to ensure parallel between the components of a system on chip.

The increase in computational capacity measured in recent years in all device electronic is bound to double flush with the miniaturization of components, and then with the integration of multiple structures on the same chip. Incorporating, for example, on the same die (the chip of semiconductor material that all enclosed inside of the package is the real CPU chip) the communication devices with the external (coprocessors, Ethernet controller, controller USB, memory controller) allows at the same time to reduce in size and consumption but also to increase the overall performance of the overall system increasing the speed with which data can be transferred between the different structures integrated [4].

Despite this fusion of heterogeneous modules, the resulting system, called SoC (System on Chip) continues to meet the classical architecture for a processor formalized by the mathematician John Von Neumann and distinguished by a unit processing (in turn integral to an internal control unit), a bus system, the system memory and the units of Input / Output most suitable according to the destination d 'use of the SoC [5]. In order to achieve this kind of SoC, current technology offers developers the possibility to choose between then utilization of FPGA (Field Programmable Gate Array) or circuits integrated ASIC (Application Specific Integrated Circuit), with these seconds that you make prefer the first to achieve better performance and better utilization resource, first area, while the FPGA appeal more successful thanks to the flexibility offered by the same programming via software that them It makes it highly competitive in terms of costs, in the case of small productions scale or in the prototyping stages, compared with slightly lower performance [6].

Whatever the technology used for the realization of the desired SoC, a point common between the FPGA and ASIC is represented by the languages used in the process of design and interpretable by the software tools dedicated [7, 8], for a description of the behaviour and characteristics of the product that you want to achieve. Among the various languages, said HDL (Hardware Description Language) for their scope application, if they have chosen the one called VHDL (VHSIC Hardware Description Language). This turns out to have some traits in common with the classical languages programming, such as if-then-else constructs and arithmetic operations logical, but unlike the latter it allows to describe some competitive features (Performed simultaneously) in a much more natural compared to other languages that as they can lean on the thread management. This difference is due to the VHDL language and permitted because with this you go to describe directly the hardware dedicated to the performance of a particular operation, while the more traditional programming languages (C, C ++, Java) the High-level programmer describes the operations to be subsequently adapted to be able to be carried out by a microprocessor or some generic system that, not necessarily being optimized for these tasks, could not have the necessary hardware resources required to fully parallelize the operations [9, 10]. The realization of an integrated circuit process involves a series of steps that lead by a higher abstraction level towards a lower level of abstraction. The set of these steps is the design flow VLSI (Very Large Scale Integration) composed of the following main phases: Functional Description: Allows us to draw up the functional specifications and any limits required (area - performance - consumption); algorithmic Description: Specifies the ordered sequence of operations leading all obtaining the desired function; Description RTL (Register Transfer Level) At this level they introduce resources necessary physical (logical arithmetic operations, registers, mux-demux, etc.) and the bond time between these structures represented by the clock; logic description: the circuit is represented using the logic gates combinatorial and sequential.

2. Related Work

The AMBA standard designed, and over the years as updated by ARM, a leading manufacturer of processors and microcontrollers for embedded systems that find many uses in electronic devices of national consumption (media players, mobile phones, etc.), and especially in many of the type of microprocessor systems SoC (System on a Chip) [11, 12].

To have behind him a company as ARM and globally recognized leader in the hold and provide IP (Intellectual Property) standards AMBA led him to become a de facto standard for the industry.

In addition to these libraries provided by the same ARM or by the community of independent developers, a further boost to AMBA standard has come from absence of royalties, which in the case of others could affect the development costs and standard interfaces [13].

Recalling that AMBA defines the specifications for the interconnection of different devices go quickly to evaluate how some parameters can directly affect the performance of the entire system.

Width of the data and address bus of the bus are among the first discriminating factors for the selection or for the definition of an interconnection apparatus, but are certainly not the only parameters that define the performance, the efficiency or the overall complexity.

A. **Types and Variants**

The updates to the protocols set over the years, together under the same AMBA family, have led in some cases to renovations of the same structure, in others to inclusion of new protocols defining specific structures that go to cover particular uses.

Let's see what are the structures defined in the various revisions of still supported AMBA standard:

AMBA-2 introduced for the first time protocols for AHB interfaces (AMBA High Performance Bus) and APB (AMBA Peripheral Bus).

AMBA-3 has provided an update of AHB structures and APB previously defined by introducing in particular of controls related to the behaviour of the Master devices when these are on the bus there are present a plurality, and has introduced a ATB interface (AMBA Trace Bus) with functionality debug and a AXI interface (Advanced eXtensible Interface) for high-efficiency systems that require special precautions including variable frequencies to keep under control the energy consumption, full duplex information transfers and much more [14].

AMBA-4 added three additional types of interfaces AXI4, AXI4-Lite and AXI4-Stream respectively update, simplify and extend the AXI standard defined in AMBA3 specifications.

Among all these interfaces the work done has concentrated on the AHB protocol that allows to obtain satisfactory results without thereby exacerbate the complexity of the bus itself.

3. Problem Definition

The purpose of the memory controller is to perform as a link between the system bus, and then the microprocessor, and the memory modules, so as to relieve the CPU of the control of all those operations defined by the standards of the memories and that are repeated frequently. Such operations if they were directly managed by the CPU, would entail for the same an excessive occupation of time and consequent reduction of the calculation capacity [15].

Among these operations in particular are highlighted those initialization of memory, all of the controls latencies related to reading and writing and refresh operations necessary to prevent the deprivation of the data contained in the dynamic type memory.

4. objective

- As increasing numbers of companies adopting the AMBA system, it has rapidly emerged as the de-facto standard for SoC interconnection and IP library development. AMBA enhances a reusable design methodology by defining a common backbone for SoC modules.
- In this work, the design and implementation of an AMBA based Memory controller is proposed.The AMBA based Memory controller gives an ease of integration for sub-frame extraction of various data structures in SOC.AMBA based interaction deals with role specific operation. It is majorly categorized in two dedicated feature i.e. decision (AHB MASTER) and response (AHB SLAVE).
- Moreover AHB master enables transfer types i.e. burst mode and AHB Master-to-AHB Slave supports incrementing and wrapping addressing modes and completes data transfer which the data width of read and write is different by asymmetric asynchronous FIFO.
- A bridge between AHB with application of memory controller will be shown and there digital efficiency in terms of area and speed will be discussed. Control structure will be designed with finite state machine. The IP of AHB Master and AHB Slave will been implemented and its interface with memory controller will be designed and tested.

5. Proposed Method

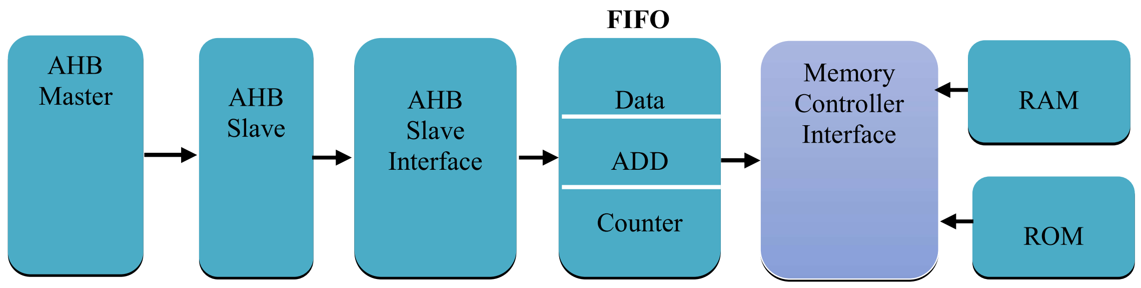

Fig 1: Top architecture of AHB memory controller

AHB compliant’s master’s top architecture is depicted in Figure 1. Here in the proposed method FIFO is used for data and control buffering. This is done with the goals of assigning different clock cycles (CLK) to slave and memory which in turn gives a foolproof communication and also decrease the complexity. Depending upon the READ and WRITE operations data is further communicated through RAM or ROM.

A. **AHB Bus Master**

An AHB bus master shown in Figure 2 has the most complex bus interface in an AMBA system. It initiates the read and write operation and send necessary control signals to salve form transfer of communication.

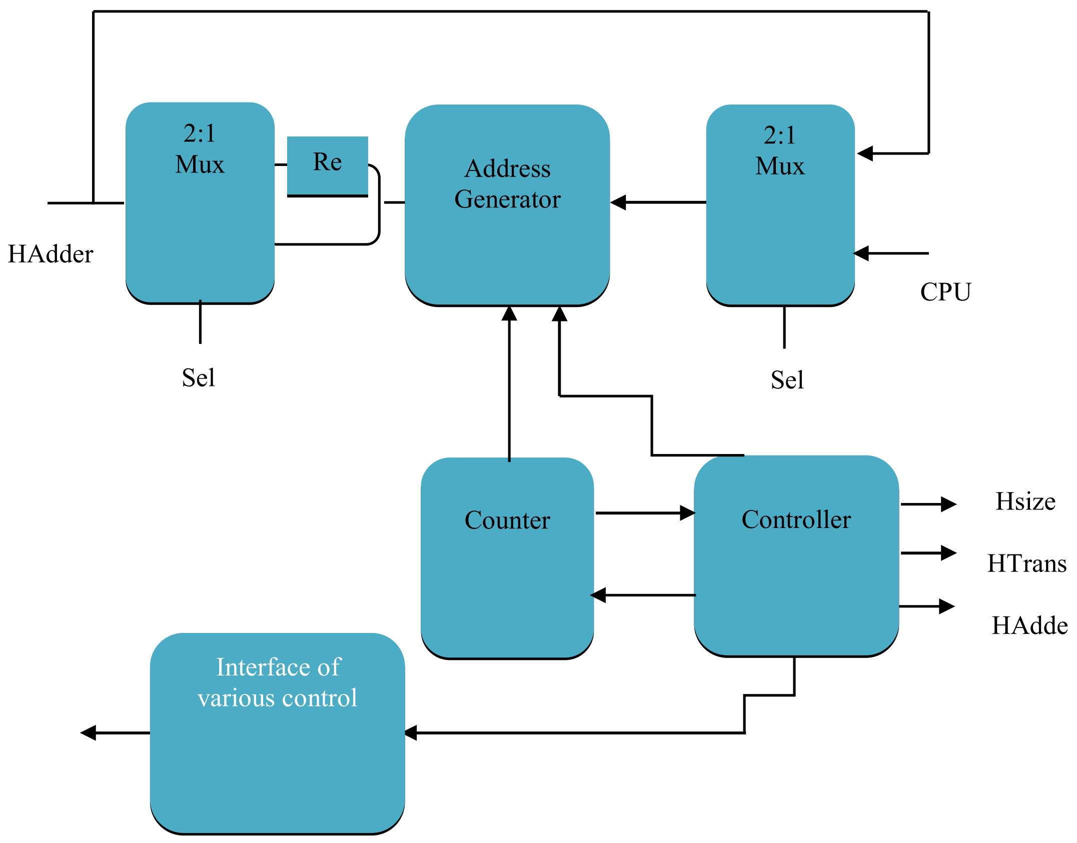

Fig 2: Top architecture of AHB master

B. AHB Bus Slave

An AHB bus slave responds to transfers initiated by bus masters within the system.The state diagram for slave interface is shown in Figure 3.

C. Memory Controller

The two main tasks for the core memory controller are to handle all the timings between different commands and to keep track of which rows that are currently activated. The activation of rows are time consuming and therefore the core memory controller has a look ahead functionality where the arbitrator can notify which command that is in turn to be executed after the current one has finished. This makes it possible to activate the row in advance if the next command is not accessing the same bank or chip as the current command. The Figure 4 shows the state diagram of memory controller.

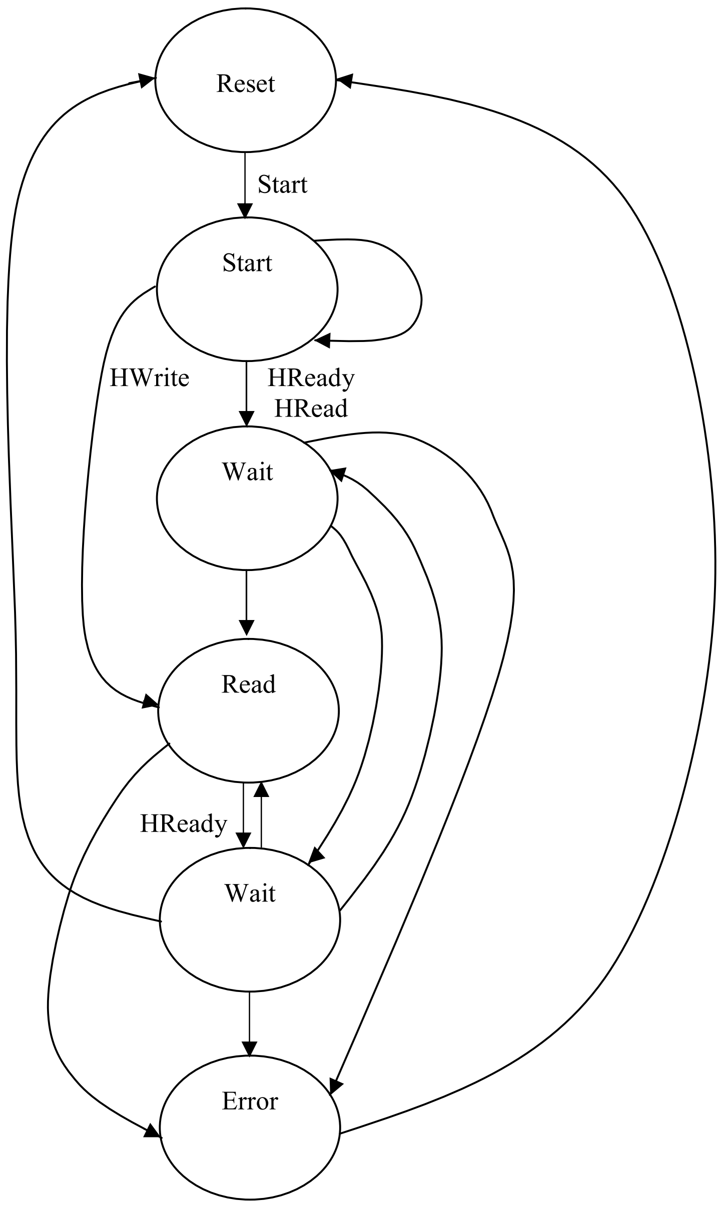

Fig.3. AHB Bus Slave

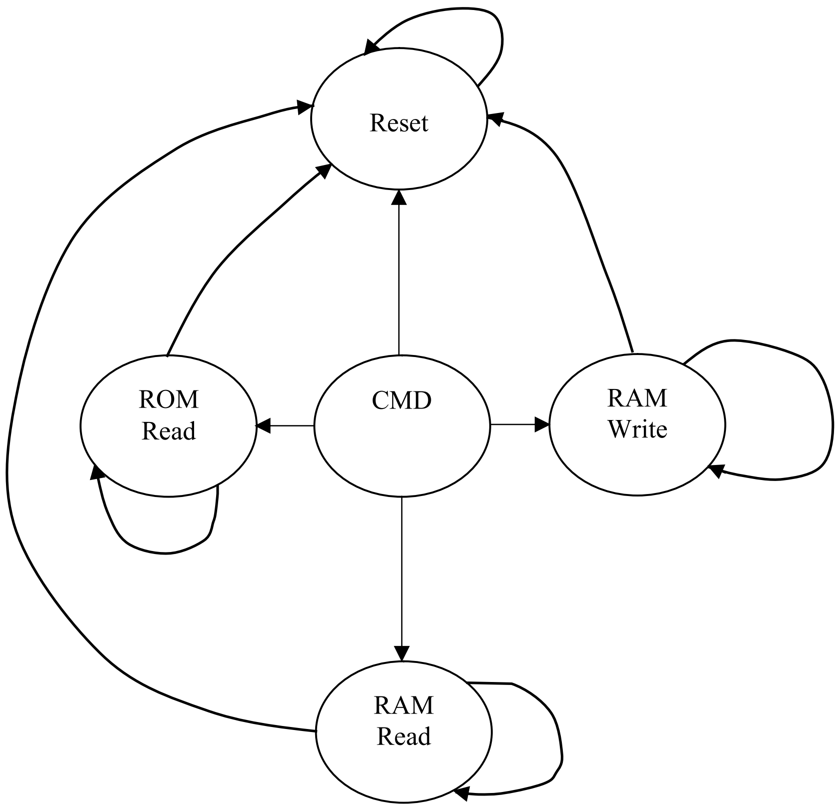

Fig 4: Memory controller

D. FIFO

FIFO is a method of processing and retrieving data. In a FIFO system, the first items entered are the first ones to be removed. In other words, the items are removed in the same order they are entered.

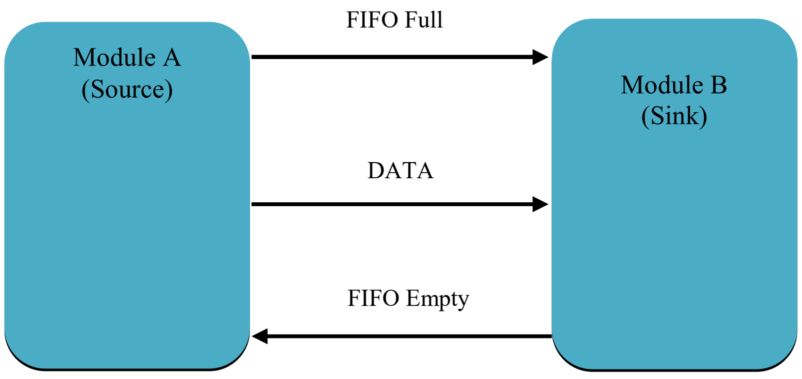

Fig 5: FIFO

Above figure shows the connection of sink and source. Synchronous FIFO low latency, for interfacing components of operating speeds very different. Their architecture is very modular, the design of the interfaces of each Side of the FIFO being perfectly decoupled. They can provide:

- Mixed FIFOs asynchronous to synchronous and reciprocally synchronous to asynchronous component for interfacing a synchronous component and an asynchronous component;
- Double-clock FIFOs to interface two synchronous components;
- Fully asynchronous FIFOs for interfacing two asynchronous components;
- These fully asynchronous FIFOs can be used as repeaters for lines long interconnection.

The full and empty signals indicate the full and empty states of the FIFO,

In advance of the actual state of the FIFO, in order to prevent the synchronous device of the synchronization latency (look ahead technique). The valid get signal is a signal of additional control that tells the device that the FIFO is already empty:

- Empty = 0 and valid get = 1: the FIFO still contains data
- Empty = 1 and valid get = 1: the FIFO delivers the last data
- Empty = 1 and valid get = 0: the FIFO is empty

The asynchronous interface does not need these additional control signals. For example, if the FIFO is full, the put_ack acknowledgment signal is simply maintained invalid until release of a place in the FIFO the mixed asynchronous to synchronous FIFO (FIFO AS) is associated with two DFF synchronizers to robustly sample the valid_get and empty signals. The FIFO synchronous to asynchronous (FIFO SA) is associated with a DFF synchronizer for Sample the full signal.

E. RAM

The read / write memories (RAM). RAM or RAM are also integrated circuits. Since the contents of each cell can be read or written, it must be able to vary. Unlike the case of ROMs, the output corresponding to a series of input address bits is not fixed as soon as it is built, but it can on the contrary change according to the program used and the data which feed it. A reading or a writing is carried out. In the case of a reading, the bits which constitute the address are received by an address decoder which locates the desired cell. Depending on whether this cell contains a 0 or a 1, the data is routed to the output on the read / write line of a 0 or the read / write line of a 1. For a write operation, the address is also decoded by the address decoder which locates the search cell, and Wants to write a 0 or a 1, the read / write line of a 0 or the read / write line of a 1 is used to route the data to the desired cell.

F. ROM

A ROM is firstly an integrated circuit. This means that it is a set of circuits interconnected for the purpose of performing a function. In this case, the function is to store a certain number of Information and return it as needed. The ROM enclosure therefore comprises the various units (Bits) and the circuits needed to deliver the required information part when and requests the address of the requested information cell.

6. Simulation Results

A. Simulation Waveforms

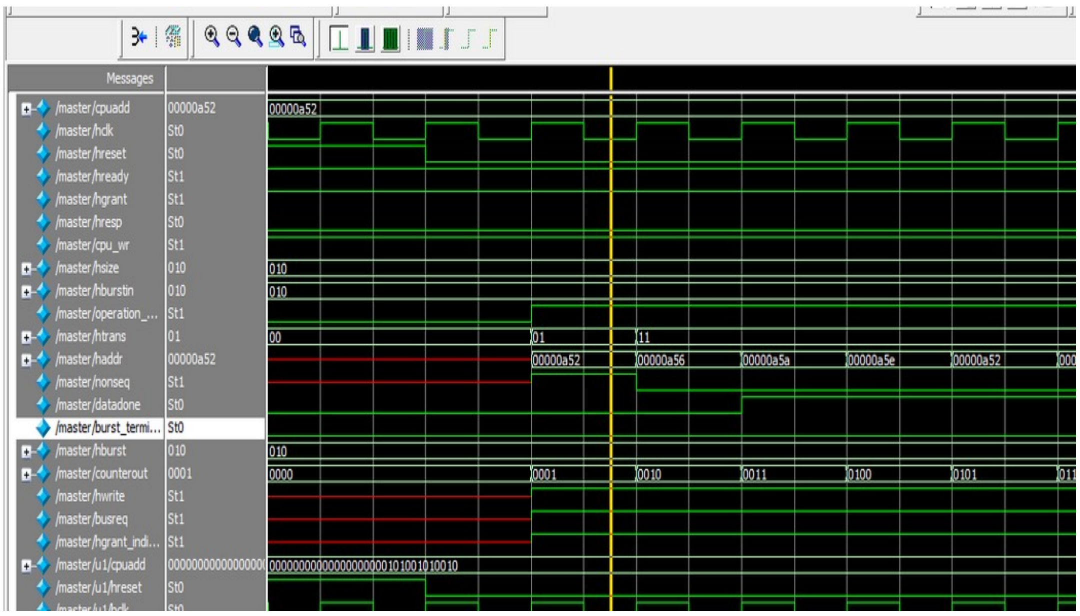

Fig 6: Simulation waveform of master interface of memory controller

The above figure shows the simulation results of master. CPU initiates the first address and rest of address is generated by master according to HBURST and HSIZE. Master will generate HTRANS signal as a response. When hready is low then master will not generate it next address. Once hready is high then master starts is next address generation.

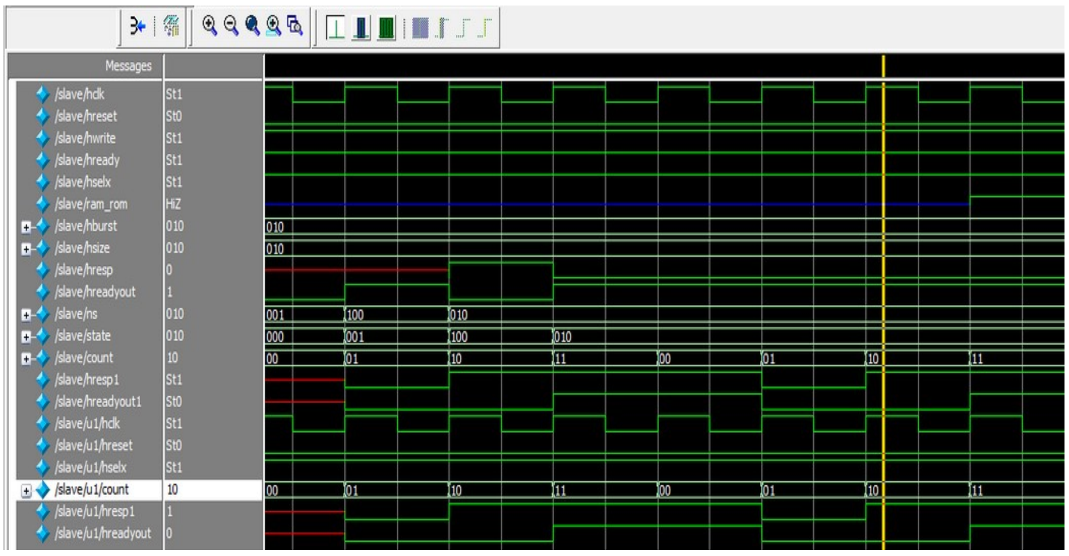

Fig 7: Simulation waveform of slave interface of memory controller

The above result shows the handshaking between slave and slave interface where slave is directly interacting with master and its response is given by slave, Further the read write operation is forwarded to memory controller by slave interface

B. RTL Schematic

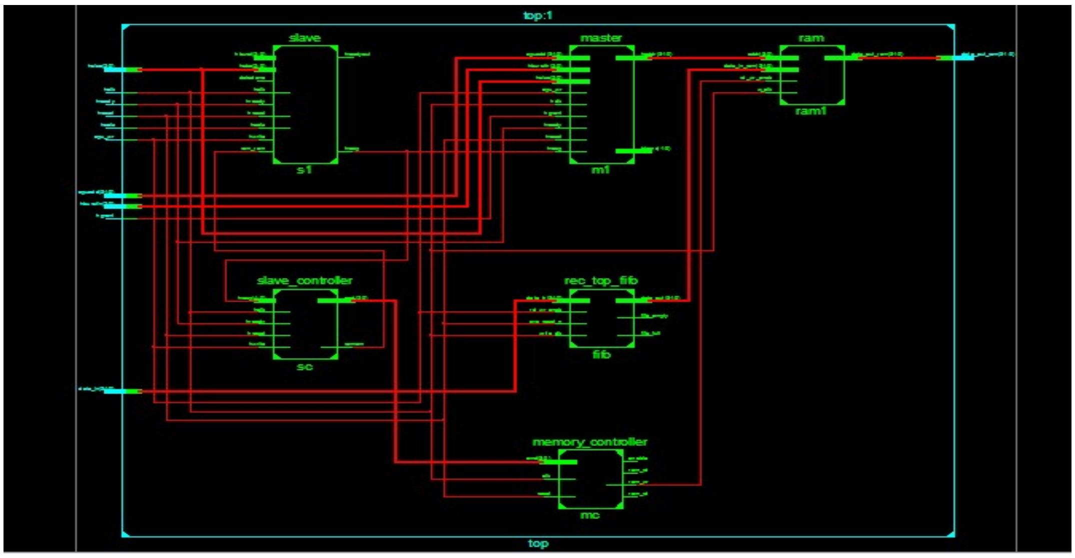

Fig 8: Top architecture of AHB master-slave memory controller

C. Result

Table 1: Result

Logic	No. of Slices	No. of Slice Flip Flops (FFs)	No. of 4-input look-up tables (LUTs)	No. of bonded input output blocks (IOBs)
Used	208	146	233	83
Available	4800	297	2400	102
Utilization	4%	49%	9%	81%

7. Conclusion

The opportunity of utilizing more innovative technology will enable software tools to synthesize the different structures in a much more effective and hence to earn higher margins both on time and area constraints because of those related to occupied area and consumption, these aspects do not cover a fundamental role in the primary personification yet clearly end to enhance the overall performance of the final product. Despite these shortcomings of the work performed study and development that followed has led the development of a memory controller on AMBA-AHB bus at a very innovative phase and next to prototyping.

The design has been developed using VHDL code and synthesized using Xilinx Virtex 6 device (XC6VCX75T). The design claims improvement in both speed and area as shown in table 1.

References

[1] Shilpa Rao and Arati S. Phadke, “Implementation of AMBA compliant Memory Controller on a FPGA”, IJETEE, 2013.

[2] Archana C. Sharma1, Prof.Zoonubiya Ali, “Construct High-Speed SDRAM Memory Controller Using Multiple FIFO's for AHB Memory Slave Interface”, IJETAE, 2013.

[3] S. Lakshma Reddy, A. Krishna Kumari, “Architecture of an AHB Compliant SDRAM Memory Controller”, International Journal of Innovations in Engineering and Technology, 2013.

[4] Arun G, Vijaykumar T, “Improving Memory Access time by Building an AMBA AHB compliant Memory Controller”, IJARCET, 2012.

[5] S. Ramakrishna, K. Venugopal, B. Vijay Bhasker, R. Surya Prakash Rao, “HDL Implementation of AMBA-AHB Compatible Memory Controller”, IJCER, 2012.

[6] Jayapraveen. D and T. GeethaPriya, “Design of memory controller based on AMBA AHB protocol”, Elixir International Journal, 2012.

[7] Ch.Vijayalakshmi, Mr B.Raghavaiah, “Implementation of AMBA AHB Compliant Memory Controller with Peripherals”, ICITEC, 2012.

[8] KareemullahShaik, Mohammad Mohiddin, Md. Zabirullah, “A Reduced Latency Architecture for Obtaining High System Performance”, IJRTE, 2012.

[9] Hu Yueli; Yang Ben, “Building an AMBA AHB Compliant Memory Controller”, IEEE, 2011.

[10] VarshaVishwarkama, Abhishekchoubey, ArvindSahu, “Implementation of AMBA AHB protocol for high capacity memory management using VHDL”, IJCSE, 2011.

[11] Zhao, B., “High speed DDR memory interface design”, IEEE, 2009.

[12] McGee, S.W.; Klenke, R.H.; Aylor, J.H.; Schwab, A.J., “Design of a processor bus interface ASIC for the stream memory controller”, IEEE, 1994.

[13] Datta and V. Singhal, \Formal veri_cation of a public-domain ddr2 controller de-sign," VLSI Design, 2008. VLSID 2008. 21st International Conference on, pp. 475{480,Jan. 2008.

[14] KeesGoossens, Om Prakash Gangwal, Jens R¨over, and A. P. Niranjan. Interconnectand Memory Organization in SOCs for advanced Set-Top Boxesand TV-Evolution, Analysis, and Trends. In JariNurmi, HannuTenhunen, JouniIsoaho, and Axel Jantsch, editors, Interconnect-Centric Design for Advanced SoC and NoC, chapter 15, pages 399–423. Kluwer, April 2004.

[15] Arm AMBA 2 speciation. [Online]. Available at: http://www.arm.com/products/solutions/AMBA Spec.html

[16] ShashisekharRamagundam, Sunil R. Das, Scott Morton, Satyendra N. Biswas, VoicuGroza, Mansour H. Assaf, and Emil M. Petriu, “Design and Implementation of High-Performance Master/Slave Memory Controller with Microcontroller Bus Architecture”, IEEE International Conference on Instrumentation and Measurement Technology (I2MTC) Proceedings, pp. 10 – 15, May 2014.

Authors' Profiles

Mr. Hitanshu Saluja is a Ph.D. scholar from Manav Rachna International Institute of Research & Studies, Faridabad. Had completed Post Graduation in VLSI Design and did Graduation in Electronics and Communication Engineering, with sound working experience over 8.6 years in various electronic works, related to Teaching, Research & Industry side.

Prof. (Dr.) Naresh Grover did his B.Sc (Engg.) in 1984 and M.Tech in Electronics and Communication Engineering in 1998 from REC Kurukshetra (Now NIT Kurukshetra). He has a rich experience of 33 years in academics. He has authored two books on Microprocessors and is a co-author of a book on Electronic Components and Materials. His core area of interest is Microprocessors and Digital System Design. Presently he is Dean-Academics at Manav Rachna International University, Faridabaad.

How to cite this paper: Hitanshu Saluja, Naresh Grover," Memory Controller and Its Interface using AMBA 2.0", International Journal of Engineering and Manufacturing(IJEM), Vol.9, No.4, pp.33-44, 2019. DOI: 10.5815/ijem.2019.04.03