ANALYSIS OF DIFFERENT ARBITRATION ALGORITHMS FOR AMBA AHB BUS PROTOCOL IN SOC DESIGN

E.Mythili[1], S. Senbaga priya[2], K.B. Sangavi[3]

[1]Assistant Professor, Department of Electronics and Communication Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore-641062, Tamilnadu, India.
[2]PG Student, Department of Electronics and Communication Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore-641062, Tamilnadu, India.
[3]Assistant Professor, Department of Electronics and Communication Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore-641062, Tamilnadu, India.

---***---

Abstract: Now a days electronic devices are getting smaller and thin, but the performance and their operation range is increasing. This is possible only because of System on chip design mythology. In this mythology number of block are integrated in as a single chip. In System on Chip (SoC) buses, to access the required functionality intellectual properties (IPs) need to communicate with each other. Bus Arbiter plays an important role to handle the requests initiated by the master and responses such as acknowledgement from the slave. This paper gives an informative review about the bus interfaces of Advanced Microcontroller Bus Architecture (AHB) and its Arbitration Algorithms to handle multiple bus requests

Keywords: AMBA (Advanced Microcontroller Bus Architecture), AHB(Advanced High Performance Bus), SoC(System on Chip), Arbitration algorithms

CHAPTER I

INTRODUCTION

Due to the rapidly developing electronics, industries have entered into an era of multimillion gate chips. This developing design technology promises to bring new levels of integration on a single chip called System-on-chip design. This also makes significant changes to the chip designer. SoC is an integrated circuit in which microcontrollers or microprocessors are integrated with the external peripherals. They are widely used in mobile computing and edge computing masks. The efficiency of how the sharing resources can be utilized determines the Performance of Multi core Shared bus Embedded Controller. Common bus in SoC is one of the sharing resources which are shared by the multiple master cores. It acts as a channel between the master core and the slave core (peripherals) or Memories. The overall performance of the System-on-Chip (SoC) design is mainly determined by the On-chip communication architecture. The communication architecture should be flexible for offering a high performance over a wide range of data traffic in the resource sharing mechanism of SoC. The communication architecture topology is a combination of a network of shared and dedicated communication channels, to which various SoC components are connected. The components are (i) a data transaction initiated by the masters (e.g., CPUs,etc.) (ii) Components that respond to the master initiated transactions called slaves (e.g., on-chip memories). Bridges are used to interconnect the necessary channels in the case where multiple channels exist in a topology. Buses are often shared by several SoC masters. Thus, to manage access to the bus, bus architectures require protocols which can be implemented in (centralized or distributed) bus arbiters. Many On Chip Bus architecture by different company are available in market but one of most popular On Chip Bus architecture is the AMBA (Advanced Microcontroller Bus Architecture) by ARM. It has an advantage that it is an open specification. The AHB (Advanced High performance Bus) is a high performance bus in AMBA family. The AHB can have a major advantage that it can be used in a high clock frequency system module. The selection of the bus which includes its type, width and topology is one of the most complex tasks in a SOC design. AMBA (Advanced High Performance Bus) is best suited bus for this purpose. Multiple IPs requests the bus at the same time, when the SoC bus is connected with more IPs. This necessity makes system designer for meeting the on-chip bus based communication a major challenge. Arbiter acts as an authority to effectively use the shared Resource (Shared bus) to make the performance dependent on the arbitration techniques. Master may request to the arbiter which is the bus master to make use of the bus during any cycle. During the rising of the clock, the arbiter will sample the request to make use of the predefined algorithm for deciding which will be the next master to gain access to the bus. Arbitration algorithms ensure that only one master can have the access to the bus at any given time. While a master is accessing a bus, all the other masters are forced to remain in an idle state until they are granted the use of the bus. Centralized arbitration is performed in this paper. Each master has an independent request and grant signals as shown in Fig 1. Priority based for I/O transactions and fairness based policy among processors is used by the multiprocessor. Buses should be designed to minimize the time required for request handling, arbitration addressing, to make most of the bus cycles used for useful data transfer operations and for optimizing the performance. Request signal followed by a response signal which is indicated by a transition signal performs a bus transaction. This may limit the maximum number of bus cycles for master to make use of the bus, through maximum transfer size or else it may split communication functions, when slave devices are slow to respond to the requests initiated by the master.

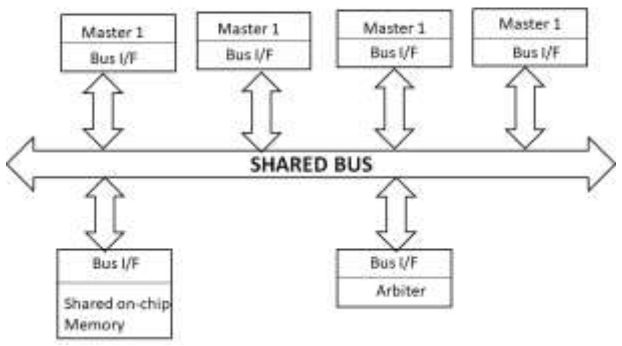

Fig:1 Shared Bus Topology

Bus transaction and arbitration competition takes place on a parallel bus with separate lines concurrently. Hence the communication architecture plays a vital role in improving the performance of SoC design. At present, centralized arbitration is dominating in embedded systems. Arbiter decides the communication between the master and the slave and hence it act as an important functional module in the multiprocessor design. Thus, it is important to carefully design the arbiter in high performance systems. Currently used communication architecture protocols are round-robin, priority based and time division multiplexing. In addition to the arbitration, the communication protocol also handles other communication functions such as to limit the maximum number of bus cycles.

CHAPTER II

ARCHITECTURE OF ADVANCED MICROCONTROLLER BUS ARCHITECTURE (AMBA)

AMBA is an open standard and on-chip interconnect specification used for connecting and managing functional blocks in SoC. AMBA was introduced by ARM in 1996. AMBA has undergone 5 versions. An AMBA-based microcontroller mainly consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB). With the help of this bus, arbiter can sustain the external memory band-width which consists of CPU, on-chip memory and other Direct Memory Access (DMA) devices. AMBA will provide a high-bandwidth interface between the elements which were involved in the major transfers. Fig3 shows an AMBA based Simple Microcontroller in which the high performance bus is a bridge to the lower bandwidth APB. Most of the peripheral devices in the system reside in this APB. AMBA APB is the basic peripheral macro cell communications infrastructure. This acts as a secondary bus from the higher bandwidth pipelined main system bus. Such peripherals typically:

(i) Contains memory-mapped registers such as interfaces

(ii) Contains no high-bandwidth interfaces

(iii) They are accessed under programmed control.

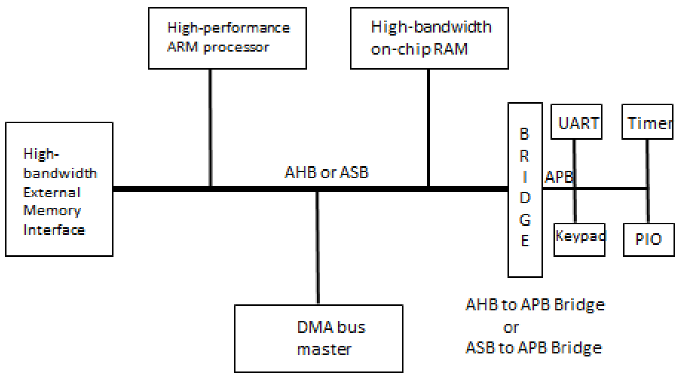
High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth External Memory Interface
AHB or ASB
B R I D G E
UART
Timer
APB
Keypad
PIO
DMA bus master
AHB to APB Bridge or ASB to APB Bridge

Fig2: AMBA BASED SIMPLE MICROCONTROLLER

2.1 DIFFERENT BUSES IN AMBA

The Advanced Microcontroller Bus Architecture (AMBA) takes the advantage of ARM's no-cost, open specification. To design a high performance Embedded Microcontrollers, AMBA is used as an on-chip communication standard. Three distinct buses can be defined within the AMBA specification such as:

1. The Advanced High-performance Bus (AHB)
2. The Advanced System Bus (ASB)
3. The Advanced Peripheral Bus (APB).

CHAPTER III

AMBA AHB

The AHB is a new generation AMBA bus which was mainly designed to address the requirements of high-performance synthesizable designs. AHB is a system bus which can support multiple bus masters at the same time and provides high bandwidth operation. The AHB also implements features that is vital for high clock frequency systems such as: i) burst transfers ii) split transactions iii) single-cycle bus master handover iv) single-clock edge operations v) non-tristate implementations and vi) wider data bus configurations (64/128 bits). The architecture of AMBA AHB is shown in fig4.

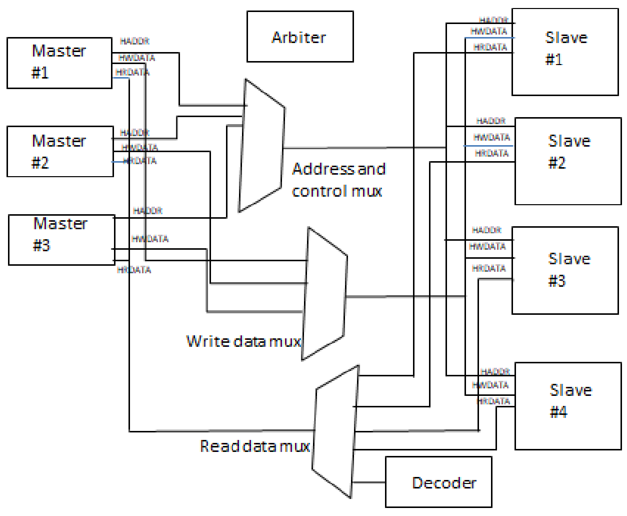

Fig3: ARCHITECTURE OF AMBA AHB

CHATER IV

AHB ARBITERATION ALGORITHMS

Arbitration mechanism ensures that only one master can have the access to the bus at any one time. This function is performed by the arbiter by observing a number of different requests to use the bus. The reconfigurable arbiter has the capacity to serve up to a maximum of 16 masters. But, here we have used only four masters out of sixteen (Master 0 through Master 3). The proposed design is a single AMBA AHB reconfigurable arbiter with three different Arbitration schemes such as Static Fixed Priority, Round Robin and Modified Round Robin algorithm which will be selected employing a signal ARBITRATION. Since it has the reconfigurable functionality, depending on the requirement of IP cores it can assign any arbitration scheme among four which are designed. Among the four, any master can request for an access of the bus. Then we can choose any arbitration scheme using input signal ARBITRATION [1:0] depending on the requirement of an application. As per the chosen arbitration scheme, grant signal will be generated to any particular master. Following this, master will get the bus access. Choice of the arbitration algorithm can be selected from the following table.

TABLE1: SELECTION OF ARBITRATION ALGORITHM

ARBITRATION [1:0]	ARBITRATION SELECTION ALGORITHM
00	Static fixed Priority Algorithm
01	Round Robin Algorithm
10	Modified Round Robin Algorithm

4.1 STATIC FIXED PRIORITY ALGORITHM

In this technique all the masters are given a fixed priority. When several masters request for accessing the same slave at the same time, then the bus grant signal will be given to the master with the highest priority among them. This function is performed by the centralized arbiter. Starvation among elements will occur when the highest priority master repeatedly request for the bus. Fig5 shows the functionality of a static fixed priority algorithm. For example, if the master 2 and master 4 requests for the bus at the same time, the arbiter will allocate the bus grant to the bus master 2. This is because the master 2 has the highest priority. After completion of operation, bus grant is given to the bus master 4. Suppose, in between any other higher priority buses such as bus master 1 request for the bus, it will provide the bus grant signal to the master 1 instead of master 4.

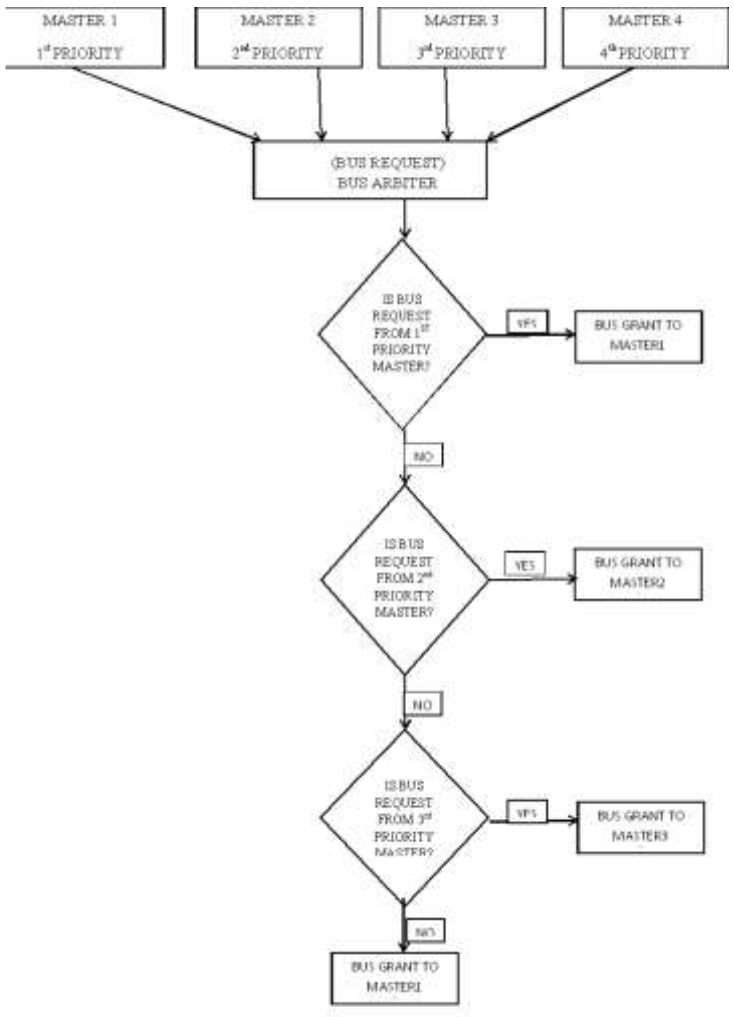

Fig4: STATIC FIXED PRIORITY FLOWCHART

4.2 ROUND ROBIN ALGORITHM

Round Robin algorithm is a token passing methodology where each mater is given fairness in providing the grant signal. There will be no starvation among the masters. It is based on the clock cycle. During each clock cycle, one of the masters will be given the highest priority based on the round robin order for accessing the bus. The highest priority master is provided with a token. If the token holding master doesn't need the bus, it passes the token to the next master in the order and the process continues.

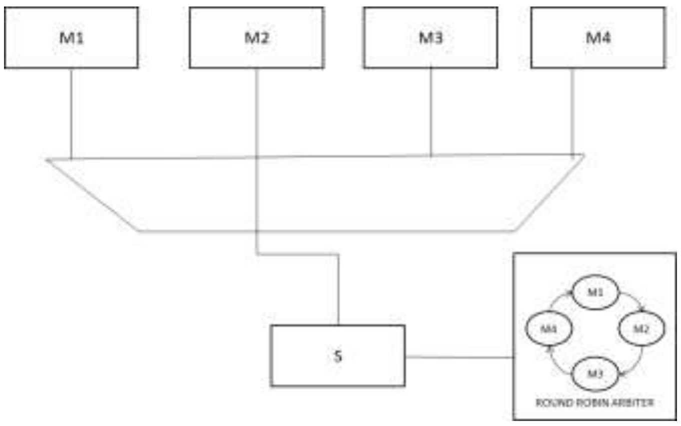

Fig5: LOGIC DIAGRAM OF ROUND ROBIN ALGORITHM

Fig7 shows the logic diagram of Round Robin algorithm in which ring counter is used to generate the tokens. For example, bus master 2 requests for the bus. During the first clock cycle, the token will be with the bus master 1. To check whether it had issued any request, AND gate is used. Since master 1 doesn't issued bus request, it passes the token to the next highest priority master which is master 2. Since it has requested for the bus, it will be given bus grant signal during the second clock cycle.

4.3 MODIFIED ROUND ROBIN ALGORITHM

In the modified round robin algorithm, the time taken for accessing the bus gets request. It is an advanced form of round robin algorithm in which a priority logic block is used. Priority logic block is a combination of priority encoder and decoder. The token generated during every cycle serves as an enable signal for the priority logic. Bus grant signal will be given based on the enable. Fig8 shows the modified round robin algorithm functionality. For example, bus master 2 requests for the bus. During the first clock cycle, the token will be with the bus master 1. Priority logic checks for the master which had issued the request. Thus, bus grant signal will be given during the first clock cycle itself. Hence, the computation will be reduced.

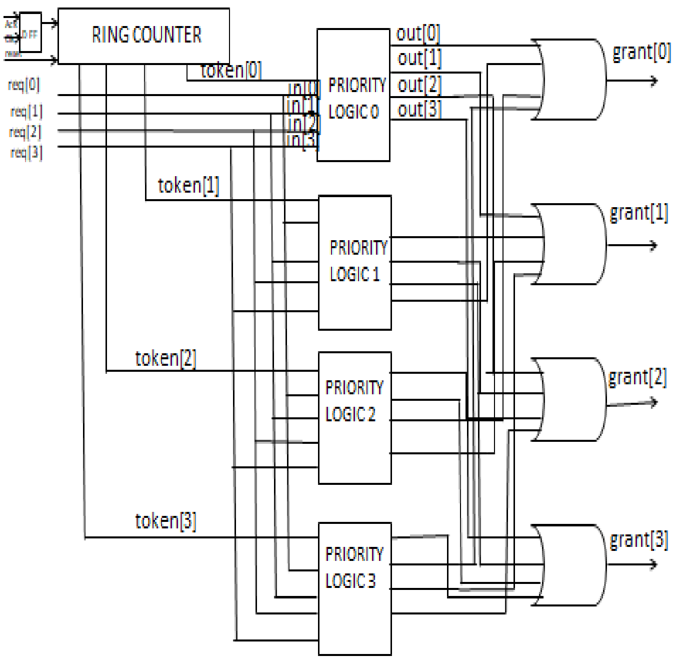

Fig6: LOGIC DIAGRAM OF MODIFIED ROUND ROBIN ALGORITHM

CHAPTER V

RESULT

5.1. SIMULATION RESULTS

Fig 8 shows the simulation result of static fixed priority algorithm when many masters requests for the bus at the same time (HBREQX=1001).

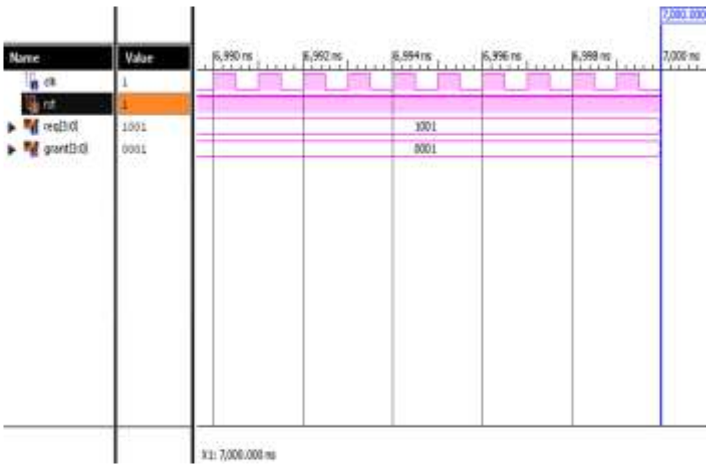

Fig7: SIMULATION RESULT OF STATIC FIXED PRIORITY ALGORITHM

Fig 9 shows the simulation result of round robin algorithm when many masters requests for the bus at the same time (HBREQX=1001).

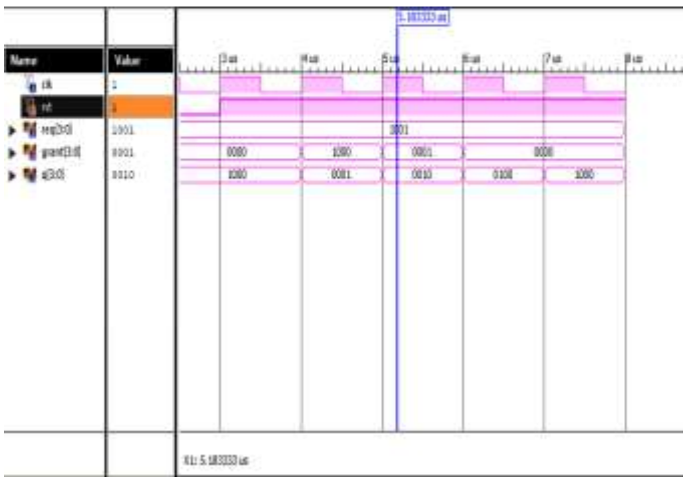

Fig8: SIMULATION RESULT OF ROUND ROBIN ALGORITHM

Fig 10 shows the simulation result of Modified round robin algorithm when many masters requests for the bus at the same time (HBREQX=1001).

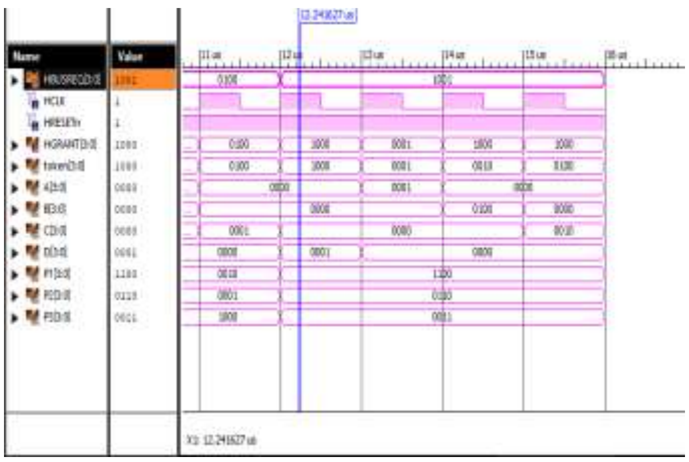

Fig9: SIMULATION RESULT OF MODIFIED ROUND ROBIN ALGORITHM

5.2. COMPARISION OF ALGORITHMS

The following Table 2 gives a comparison of arbitration algorithms of AMBA AHB protocol. This is synthesized in Xilinx XC3S500E device of the family SPARTAN3E with package FG320. Consider a scenario in which two Bus Masters request for a bus at the same time. For example if Master 2 and Master 4 requests at the same time (HREQX=1010), then the time taken by the arbitration algorithms to respond to the request signal from the Master 4 is compared in the Table 3. From the table shown below, it is clear that the Modified Round Robin algorithm is more efficient compared to other algorithms in terms of its speed and handling multiple requests.

TABLE 2: COMPARISION OF ARBITRATION ALGORITHMS

TYPE OF ARBITRATION ALGORITHM	NUMBER OF LUTs USED	DELAY/CYCLE (ns)
Static Fixed Priority Algorithm	4	8.123
Round Robin Algorithm	6	11.798
Modified Round Robin Algorithm	15	18.617

TABLE 3: COMPARISION ON DELAY OF ARBITRATION ALGORITHM

TYPE OF ARBITRATION ALGORITHM	NUMBER OF CLOCK CYCLES TAKEN	DELAY (ns)
Static Fixed Priority Algorithm	Mater 4 gets the grant signal only when the Master 2 disable the request signal	
Round Robin Algorithm	4	47.192
Modified Round Robin Algorithm	2	37.617

CHAPTER VI

CONCLUSION

In this paper, several techniques for implementing the AMBA AHB (Advanced High performance Bus) arbitration has been analyzed and compared. Arbitration algorithms such as static fixed priority algorithm, Round Robin algorithm and Modified Round Robin algorithm have been compared in terms of its area and speed. From the results, Static fixed priority is simple to implement and has less delay. But it fails in handling the request from the lowest priority bus masters. Round Robin and Modified Round Robin algorithm can handle more than one request. But, Modified Round Robin (MRR) Algorithm is more efficient than the Round Robin algorithm in terms of speed.

REFERENCES

[1] Anurag Shrivastava and Dr.Sudhir Kumar Sharma "Various Arbitration Algorithm for On-Chip(AMBA) Shared Bus Multi-Processor SoC" published in IEEE Students' Conference on Electrical, Electronics and Computer Science on April 2016

[2] Flynn, D. in Adv. RISC Machines Ltd., Cambridge, "AMBA: enabling reusable onchip designs", IEEE Micro, Published on Jul/Aug1997.

[3] Guoling Ma,Hu He in "Design and Implementation of an Advance DMA Controller on AMBA-Based SOC "published on 2009.

[4] Gulab Singh Yadav and Deepika Soni "Implementation Of Amba Ahb Bus Arbiter Using Conflation Algorithm" published in International Journal For Technological Research In Engineering Volume 4, Issue 9 on May-2017

[5] Hu Yueli,Yang Ben in "Building an AMBA AHB compliant Memory Controller" published on 2011.

[6] Jaehoon Song, Student member, IEEE, Hyunbean Yi, Member,I EEE, Juhee Han, and Sungju Park, Member, IEEE, in " An Efficient SOC Test Technique by Reusing On/Off-Chip Bus Bridge" IEEE Transactions on Circuits and Systems-I: Regular Papers, published on Vol,56,No.3,March2009.

[7] K. Manikanta Sai Kishore and M. Naresh Kumar "Design And Implementation of Efficient FSM For AHB Master And Arbiter" published in IJMETMR on March 2015

[8] Keerthana Keerthu "The Design and Verification of AMBA AHB Protocol by using System Verilog" published in IJEECS ISSN 2348-117X Volume 7, Issue 4

on April 2018

[9] Montek Singh and Rohit Agrawal "Modified Round Robin Algorithm (MRR)" published in IEEE International Conference on Power, Control, Signals and Instrumentation Engineering on March 2017

[10] Milica Mitic and Mile Stojcev, in "An Overview of On- Chip Buses" published on SER.: ELEC. ENERG. vol. 19, no. 3, December 2006, 405-428.

[11] Ramesh Bhakthatchalu, Deepthy G R, Shanooja S. in "Implementation of Reconfigurable Open Core Protocol Complaint Memory System using VHDL"published oin 2010.

[12] Rohit Hardia, Prof.Jai Karan Singh, Prof..Mukesh Tiwari in "Design And Simulation of a typical high performance AHB Reconfigurable Master For On- chip bus Architecture using Verilog HDL",published on International Conference on Communication Systems and Network Technologies, IEEE 2011.

[13] Rinku and Pawan Kumar Dahiya "Advance High Performance Bus Arbitration Techniques (AHB): A State-of-the-Art Review" published in IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 7, Issue 4, Ver. I on Aug. 2017

[14] Sangik Choi and Shinwook Kang, Mobile Samsung Electronics Co.,Ltd, in "Implementation of an On-Chip Bus Bridge between Heterogeneous Buses with Different Clock Frequencies". Published on IEEE, IDE-AS'05,1098-8068/2005.

[15] Vani. R. M. and Roopa. M in "Design of AHB2APB Bridge for different phase and Frequency" in International Journal of Computer and Electrical Engineering, Vol. 3, No. 2,pp. April, 2011.