***I.J. Engineering and Manufacturing*, 2020, 1, 29-40**
Published Online February 2020 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijem.2020.01.03

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

Multiple Master Communication in AHB IP using Arbiter

Hitanshu Saluja, Dr. Naresh Grover

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

Received: 11 November 2019; Accepted: 28 December 2019; Published: 08 February 2020

Abstract

The major disadvantage of a standard bus topology is the constraint of being able to realize only one communication at a time (the tasks may take place in parallel but the communications are only done in a sequential). As these communications are handled by the bus arbiter, a Bottleneck when the number of communications increases, but also when the bandwidth constraints of several communications become important.This arbitration plays a predominant role because it authorizes communications on the bus but it is also in charge of resolving the conflicts (several requests of communications at the same time). This arbitration implies therefore a limitation on the number of IP connected to the bus to a dozen elements.

This work elaborates the AMBA bus interface with four masters interacting with single memory system, using Arbiter between memory controller and other supporting peripherals. Different module of i.e., AHB MSTER, AHB SLAVE INTERFACE AND AHB ARBITER(round robin algorithm)has been developed with VHDL. Further integration with FIFO, RAM and ROM with memory controller is done. The Four AHB master initiates the operations and generates the necessary control signals on single bus to memory controller with the help of arbiter. The proposed architecture shows the area efficient management as compared to previous researches of multiple data communication in AHB BUS system. The system model is synthesized with Xilinx XC6vx75t-2ff484 and simulated with MODELSIM.

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

* Corresponding author.
E-mail address:

1. Introduction

Interconnection networks play a major role in the performance of on-chip systems today and in the future. Several factors may influence the choice of architecture, since a system-to-chip interconnection must satisfy certain constraints so that it is feasible. It must be efficient, flexible, the cost constraints imposed by the specifications and be physically implantable on the available technologies (FPGA or ASIC for example). Current and future SoCs because the bus lines are getting longer and longer as the number of connected IPs increases. Thus, capacities which cause an increase in the charge time (proportional to the number of connected elements). The frequency of operation of the bus is thus reduced and increased power consumption. Finally, the propagation time of the signals on long data wires and bus control leads to drift of the clocks and generates intrinsically synchronization problems (length of interconnections, band passing, and energy consumption)[1]. Given the growing complexity of SoCs integrating more than fifty IPs with large bandwidths, systems based media become a bottleneck in terms of performance but also of consumption and complexity of physical implementation on silicon. These frameworks are commonly standard busesdisplayed in one language portrayal of equipment like the VHDL. Communications in a AHB bus type starting with the hand of the Master, that in a first phase should be transmitted on the bus of the device address to which desires access and information relating to the typed access, while in the next step obtains information from the interrogated device (or completes the transfer of data to the device previously addressed, if the operation to be performed is writing rather than reading). This operation characterized by two distinct phases of addressing (Address Phase) and transmission of information (Data Phase) allows to implement a communication system that provides for the parallelization of operations on a pipeline[3]. Allowing the Master to submit a new request for information (including to a different slave) while the same Master remains pending the request information in the previous clock cycle, up to speed and in the best operating conditions, it is possible to double the flow of information circulating through the bus. The two phases characterizing the AHB bus are synchronous to the rising edge of the unique clock present in the AHB bus as shown in the timing diagram Figure 1:

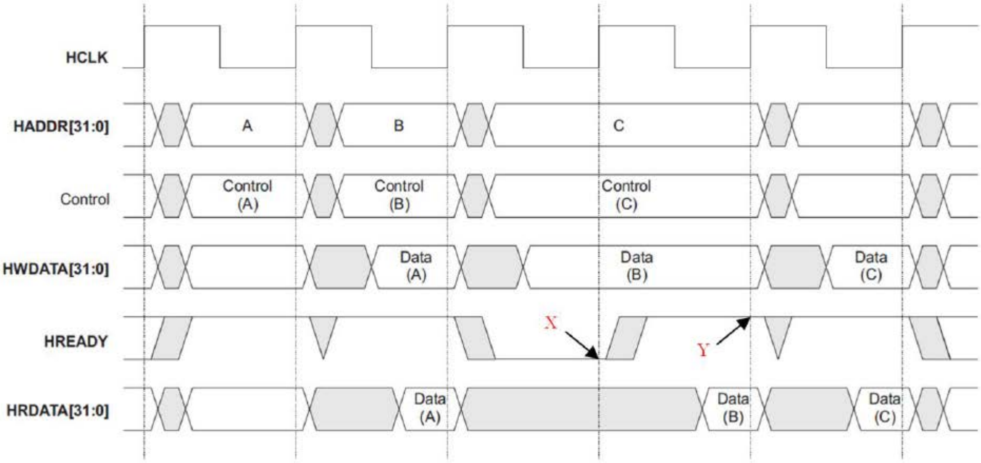


Fig. 1: Data transfer of AHB bus

On the rising edge of the clock the slave samples the address bus and various control signals, while on the next rising edge the Master obtains the requested data through the read bus (or transmitted through the write data bus information to be saved on the slave if the operation to be performed was of type write) and at the same time is ready to interrogate a new device by loading a new address on the address bus.

If the slave interrogated device unable to supply in good time the request information, the slave itself must notify the Master lowering the HREADY signal (as shown in the timing diagram point X), the Master, he took note of what will repeat the same question the next device strictly maintaining the order to inside of the pipeline, until you get an affirmative response from the first slave (Y)[4].

In particular, if as shown in example above the Master meant access to the memory cells A, B, C and so on, and the device relative to the cell B were to require an arbitrary number of clock cycles before being able to provide a valid result of the operations on the cell C would be made slip of the same number of clock cycles, thus avoiding that the result on the C cell can be received before that relating to the cell B.

The AMBA standard therefore prefers a concept of pipeline in-order, in which the sequence of operations that follow one another is determined a priori and will be completed following the same order, regardless of the execution times for the different operations[5]. A type of out-of-order system, which provides the ability to change the order response of devices in the event one of these would require more time to process information, facilitating subsequent those which are with results already available, will surely prove more effective but also more complicated to implement and manage.

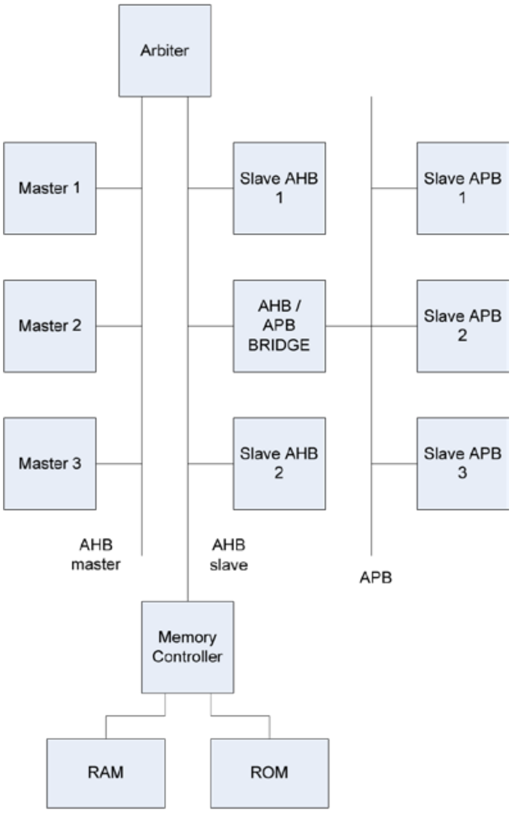


Fig.2: AMBA bus generalized diagram

2. Proposed System Model

Development of the AHB system (Multiple Master) with memory controller compatible with AMBA 2.0 protocol is proposed [8].Data is initiated by master and granted by arbiter to carry the further communication through slave-to-memory interface. If multiple processors are interacting then arbitration requires. AMBA 2.0 supports the arbiter concepts at the cost of system complexity. Following are the details of AHB proposed system model.

A. AHB MASTER

The operation of the AHB master and AHB slave buses shown in Figure 3. On the master side, the arbiter has as its output a single bus (containing the data and control bits) which will be the master input (ahbmi) and has as input all the output busses (ahbmo) of each master. In general, the master makes a query on the ahbmo bus to read or write on a slave. The referee takes care of sending it to the good slave via the bus ahbsi. The slave answers on the ahbso by specifying the master number. Finally, the master receives the answer through the referee via the ahbmibus[9].

There is therefore an "In" and "Out" part for each AHB bus. The following table shows the signals of the ahbsi bus:

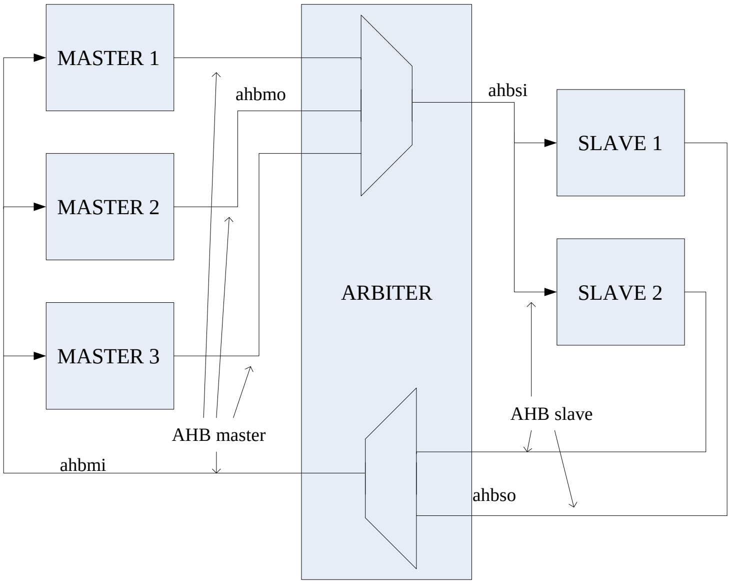


Fig. 3: AMBA Arbiter

Table 1. Signals for arbiter

Name of signal	No. of bits	Function
haddr	32	Address of transfer to court
hwdata	32	Data for a write operation
hsel	16	Slave selection (one bit per slave)
hwrite	1	0: reading, 1: writing
hburst	3	Indicates whether there is a "burst transfer" and its characteristics, detailed below
hsize	3	(0: 8 bits, 1: 16 bits, 2: 32 bits, ..., 7: 1024 bits) can be configured for the following reasons:
htrans	2	Type of transfer to court, detailed below
hprot	4	Protection signals, detailed below
hready	1	0: wait state, 1: normal
hmaster	4	Index of the master doing the operation
hmastlock	1	0: normal, 1: indivisible operation
hmbsel	4	Bank selection
hcache	1	Data "cacheable"
hirq	32	Interrupt vector

B. AHB ARBITER

Activity of Arbiter is clearly understood by the figure 4. There are 4 masters and 2 salves in shared manner, master can send their request to priority block and as indicated by characterized need on master will be allowed for its activity. Master 1 will be the default master of activity if all4 not mentioning the award. Following are the subtleties of the full scale blocks utilized for Arbiter design[10].

1) 4 priority logic for round robin implementation
2) Counter
3) 4:1 mux
4 Priority shift
5) Controller
6) Encoder

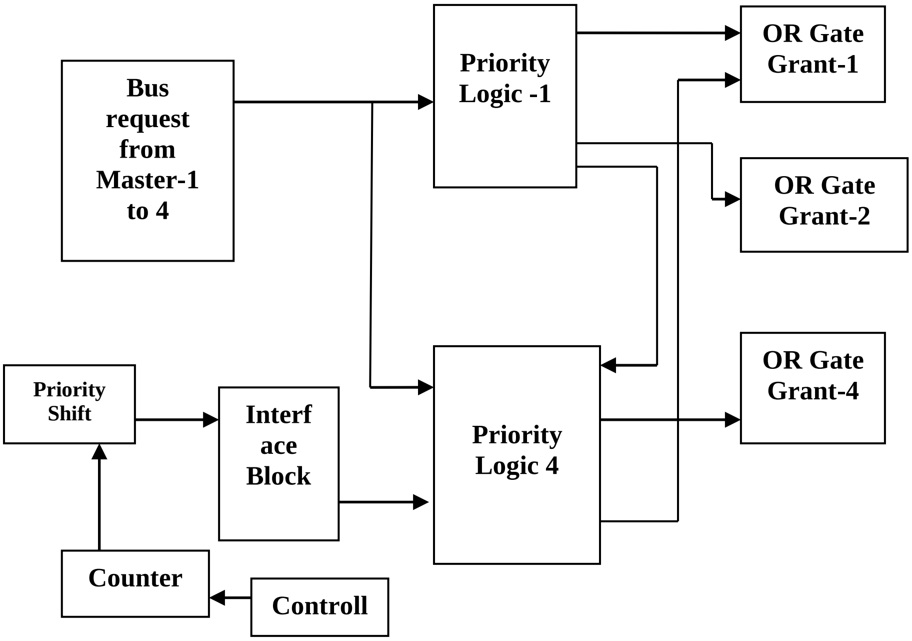


Fig.4: Block diagram of Arbiter

The priority logic block is implemented through FSM approach the priority scheme follows the Round Robin theorem of priority. The bus demand have most elevated need will get grant first and rest of solicitation will hang tight for their need. In1 signal is the input for this block that is nothing but the Bus_req, Out1,Out2…….Out4 these are the Grant signal as output, these sign is additionally gated OR and sent to the yield port, as appeared in figure. There are 4 priority logical block, they are named as priority 1, priority 2……….priority 4.

C. AHB Slave

An AHB slave interface flow diagram is shown in Figure 5, it is a finite state machine implementation initial condition is reset state which is an idle state when no operation is there. When start signals arrived then this finite state machine (FSM) triggers, depends upon the HREADY and HWRITE signal it decides in which further state it has to move. If HRAEDY is low then it will be start state only, if HREADY is one then depends upon HWRITE it move to read or write state, if HWRITE is one then state move to write otherwise it moves to read state. If HREADY will become low between these states then it moves to wait state, and it will remains in this state until HREADY will become one. If any error occurs which is indicated by HRESP then state moves to error state.

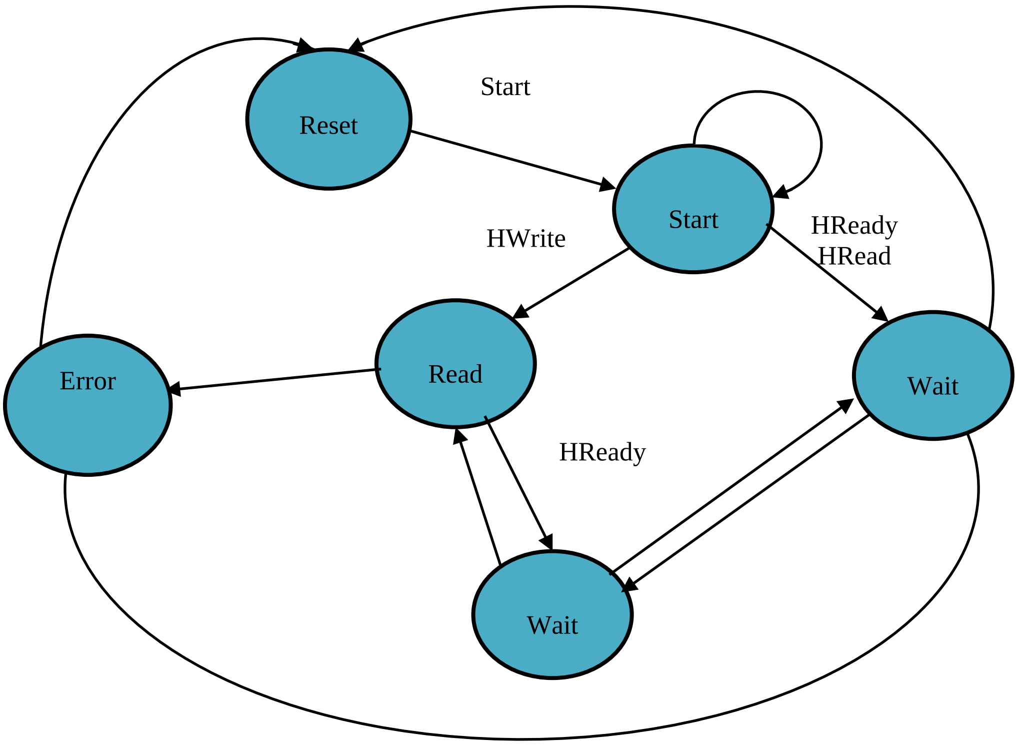


Figure 5: Slave Interface

3. Simulation Results

After simulating the proposed design the device utilization summary is shown in table 2
Using xilinx device XC6vx75t-2ff484

Table 2. Device utilization summary of proposed method in Xilinx XC6vx75t-2ff484

DEVICE UTILIZATION	Proposed Method(XC6vx75t-2ff484)
No. of Slice registers	180
No. of Slice LUT	215
No of LUT-FF	137
NO. of IOB	91

The main arbiter pin diagram is having 20 pins as shown in fig 6, name of each pin is mentioned in fig, each pins having its different functionality as mentioned in table no 3.

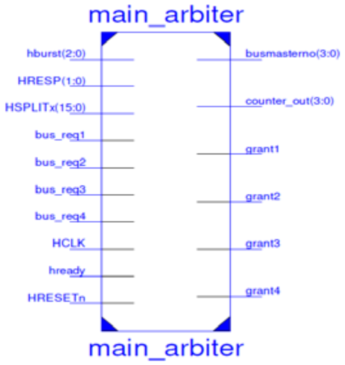


Fig.6: Pin diagram of AHB arbiter for four master request

Table 3. Functioning of main arbiter pins

Name of signal	No. of bits	Function
Hburst[2:0]	3	Defines the type of address generation
Hresp[1:0]	2	Response of slave for data reception status
Bus req	1	There are 4 bus request to arbiter given master for slave access..
hclk	1	System clock
hready	1	0: wait state, 1: normal
Hrestn	1	Asynchronous reset.
Hbusmasterno[3:0]	4	Defines which bus master is selected.
Gramt 1,grant4	1	There are 4 grant selected according to their priority

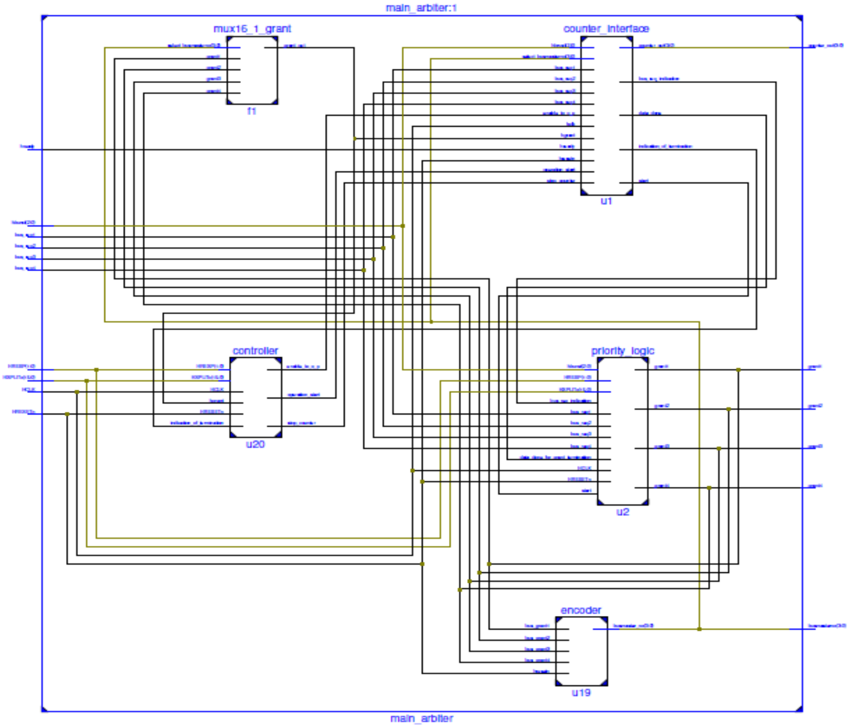


Fig.7: RTL view of arbiter to support 4 master

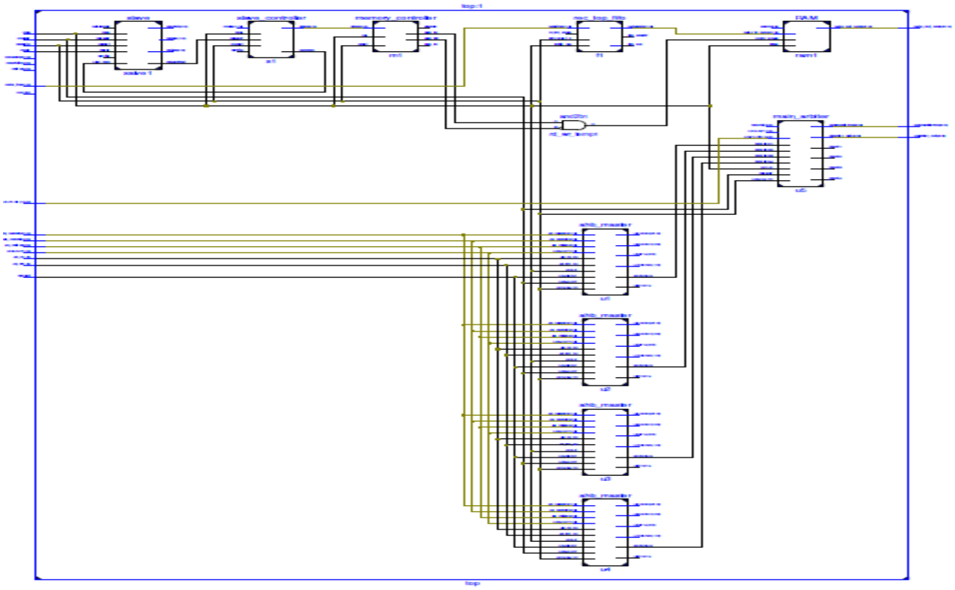
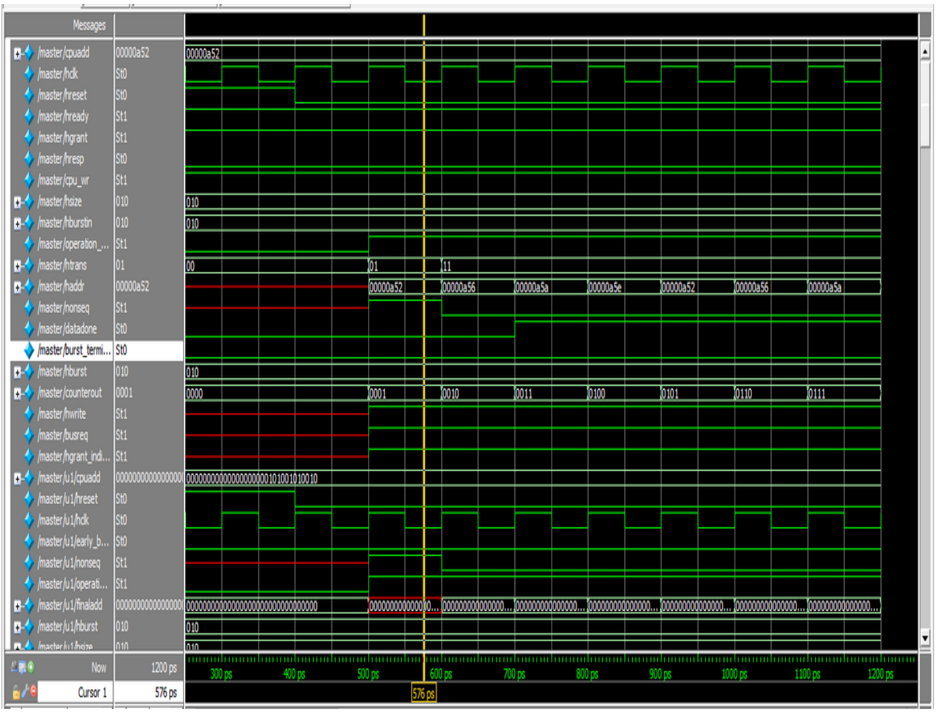


Fig.8: Top RTL view of four master interacting with arbiter and different memory elements



Messages	
/master/cpuadd	00000a52
/master/hclk	St0
/master/hreset	St0
/master/hready	St1
/master/hgrant	St1
/master/hresp	St0
/master/cpu_wr	St1
/master/hsize	010
/master/hburstin	010
/master/operation_...	St1
/master/htrans	01
/master/haddr	00000a52
/master/nonseq	St1
/master/datadone	St0
/master/burst_termi...	St0
/master/hburst	010
/master/counterout	0001
/master/hwrite	St1
/master/busreq	St1
/master/hgrant_indi...	St1
/master/u1/cpuadd	0000000000000000
/master/u1/hreset	St0
/master/u1/hclk	St0
/master/u1/early_b...	St0
/master/u1/nonseq	St1
/master/u1/operati...	St1
/master/u1/finaladd	0000000000000000
/master/u1/hburst	010
/master/u1/hsize	010
Now	1200 ps
Cursor 1	576 ps

Fig.9: Simulation of AHB MASTER

4. Conclusion

Themulti master correspondence in AHB IP module has been structured with Arbiter, so which can communicate with the processor for it to arrange, it additionally can watch moves between the processor and a peripheral. It is fit for controlling the correspondence that happen in such moves. The structure considers the framework model with transport solicitation of four masters and one slave, additional component of split control is moreover considered in proposed IP interface. The plan has been created utilizing VHDL code and simulation using Xilinx ISE. The structured is performed in Virtex gadget of Xilinx and claimed less in area occupancy in devices. The speed and power is determined as 319.519MHz and 198.21mW and found better than the previous researches. The upside of this plan is that we have used maximum no of flip flops while taken care of formation of less no of latches have improved our area efficiency as well as better speed and power.

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] Ragab, E., El Ghany, M.A.A. and Hofmann, K., 2018, December. DDR2 Memory Controller for Multi-core Systems with AMBA AXI Interface. In 2018 30th International Conference on Microelectronics (ICM) (pp. 224-227). IEEE.

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

[7] Bevoor, S. and Gowda, P.N., 2017, May. Design of high performance master/slave memory controller for grey scale image transfer. In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 611-614). IEEE.

[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] Saluja, H. and Grover, N., 2017. A REVIEW OF AHB PROTOCOLS WITH MEMORY CONTROLLER. International Journal of Pure and Applied Mathematics, 114(10), pp.349-361.

[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] Acasandrei, L. and Barriga, A., 2017. Open Library of IP Module Interfaces for AMBA Bus. In IAENG TRANSACTIONS ON ENGINEERING SCIENCES: Special Issue for the International Association of Engineers Conferences 2015 (pp. 281-294).

[15] Aljumah, A. and Ahmed, M.A., 2015. Design of high speed data transfer direct memory access controller for system on chip based embedded products. Journal of Applied Sciences, 15(3), p.576.

[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. "Multiple Master Communication in AHB IP using Arbiter", International Journal of Engineering and Manufacturing(IJEM), Vol.10, No.1, pp.29-40, 2020. DOI: 10.5815/ijem.2020.01.03