ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 9, September 2017

16 Channel AMBA AHB multiple Arbitration Technique with Priority selection

Mukesh Kumar sen[1], Neeta Moolani[2]

1. P.G Scholar Department of ECE –SKSITS Indore, 2.Asst. Professor Department of ECE –SKSITS Indore,

Abstract

Most of the FPGA features communicate with each other through bus means. Each block is designed for a specific bus. The buses related to this work are the AMBA bus and the Wishbone bus. It works with the master / slave architecture. The AMBA bus is extensively utilized in the system on chip solution for interaction with different peripheral. The proposed work includes the regular AMBA specifications of burst transfer, several bus masters, single clock edge transition, and Split transaction. The data transfer is accomplished by a single bus master at a while. Despite the fact that the arbitration protocol is settled, any arbitration algorithm, for example, fair access or highest priority can be actualized relying upon the application necessities. VHDL code is utilized to develop the design and it is synthesized on Virtex-4 series. Utilization of on chip bus architecture is to integrate various components like CPU, memory, DSP and availability of bus as on request by master is very important. System efficiency depends large on bus available as on request.

Keyword: - AMBA, AHB, VHDL, SOC, Bus, APB, RTL

I. INTRODUCTION

Innovative advancements on the global industrialization have permitted the joining of an ever increasing number of capacities on the same digital integrated circuit. Now a days, various hardware accelerators, microprocessors, communications systems are classified with operating systems. These permits along these lines achieve every one of the capacities important to perform complex computer processing on the similar chip, thus the introduction of the idea of System on Chip, successors of specialized circuits ASIC (Application Specific Integrated Circuit) .

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 . 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. 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.

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. 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.

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 [8.7], for a description of the behavior 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. 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;

Description: What are geometrically defined masks and interconnections used for the realization of the integrated circuit. The transition from one level of description to the next called synthesis (algorithmic - Logic - RTL - physical) can be automatic or, in some cases, require the designer definition of constraints and the choice between several possible solutions. For each synthesis carried out must necessarily be followed by a verification and simulation phase to ensure that the translation of the circuit from a high level of abstraction to a more low, showing still correspond to what initially requested to the system. To do so a second circuit, or software model is almost always used, which it acts as a test bench (test bench) responsible for providing all the necessary signals for stimulate the component under test. The inputs from the circuit and the answers offered below examination are then analyzed using the simulation tools, mostly software, through which it will be possible to verify the correspondence of responses obtained with those expected.

AMBA Bus System

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 ...), and especially in many of the type of microprocessor systems SoC (System on a Chip).

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 develop a 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.

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.

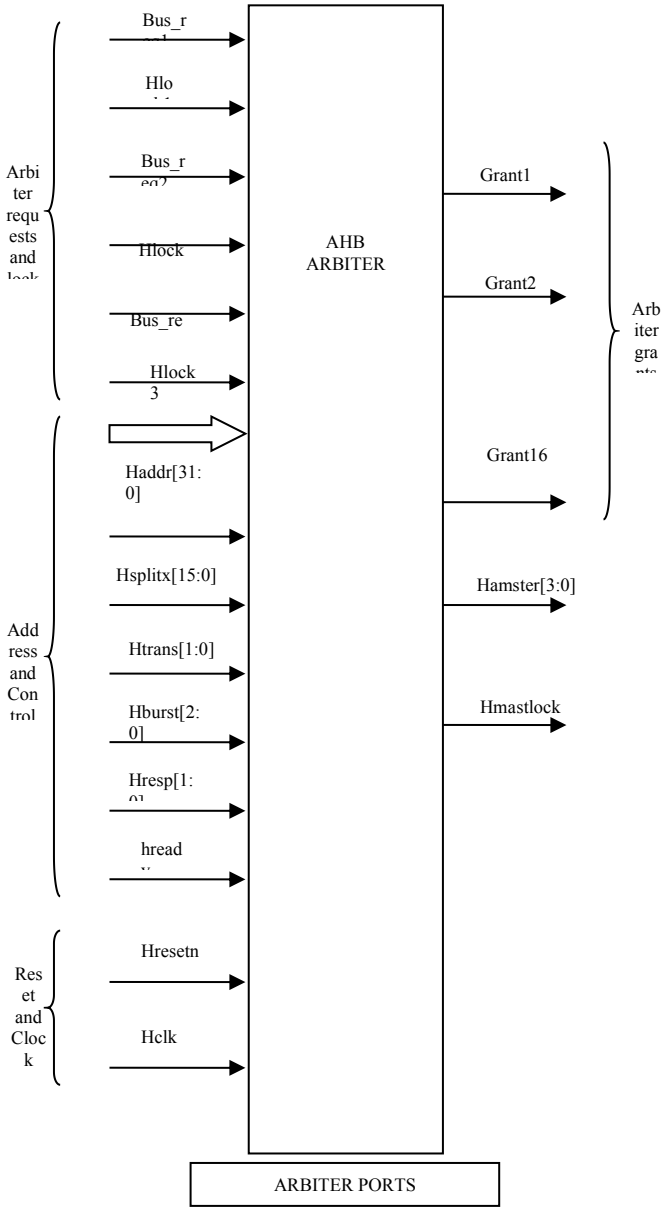
Width of the data and address bus of the bus are among the first discriminating factors for the selection or for the definition of an interconnection apparatus, but are certainly not the only parameters that define the performance, the efficiency or the overall complexity.

It is sufficient to add some variables such as the clock frequency used for the synchronization of the transmissions, or the degree of parallelism obtainable by differentiating the read / write bus and still the procedures for the transfer of information and for the control of the communication bus between two devices connected to the same bus, to obtain a huge variety of structures that can be grouped all under the generic definition of d interconnection bus, but that perform the same task with very different methods and operations from each other.

also we should not underestimate the weight assumed by interconnection bus on the complexity of accomplishment of the same (which must extend and reach a good part of the area of the SoC) and of all the devices that will be connected to it; most of the design choices that can be embraced to enhance the execution of the framework bus performance resulting in an increase of the realization cost of the bus and often also of compatible devices that they will have to implement particular controls on the different procedures that constitute a data transfer on the bus

I. Architectural Overview

Arbitration to pick the subsequent bus master utilized a round robin arbitration scheme. This guarantees no master gets empty. At the point when a master has sealed the bus, the round robin arbitration is abrogated and the master with the seal holds most noteworthy need to the bus.



Pin Diagram for AHB Arbiter

There are 16 master and 1 salve in shared manner.16 master can send their request to priority block and according to defined priority on master will be granted for its operation. Master 1 will be the default master of operation if all 15 are

not requesting the grant. Following are the details of the Marco blocks used for Arbiter design.

1) 16 priority logic for round robin implementation
2) Counter
3) 16:1 Mux
4 Priority shift
5) Controller
6) Encoder

III. RTL Schematic

RTL is register transfer level graphical representation; here we are using Vertex -2 board of Xilinx 13.1 ISE. All files will be with extension .ngr, it will produced by Xilinx.

RTL view gives close results to HDL codes and hence very effective tool for synthesis of data. RTL having standard blocks including adder, Multiplier, register, Gates etc. This will help defining circuit accordingly. Standard block can be utilized as per HDL requirement and gives RTL output close to customized code.

IV. SIMULATION

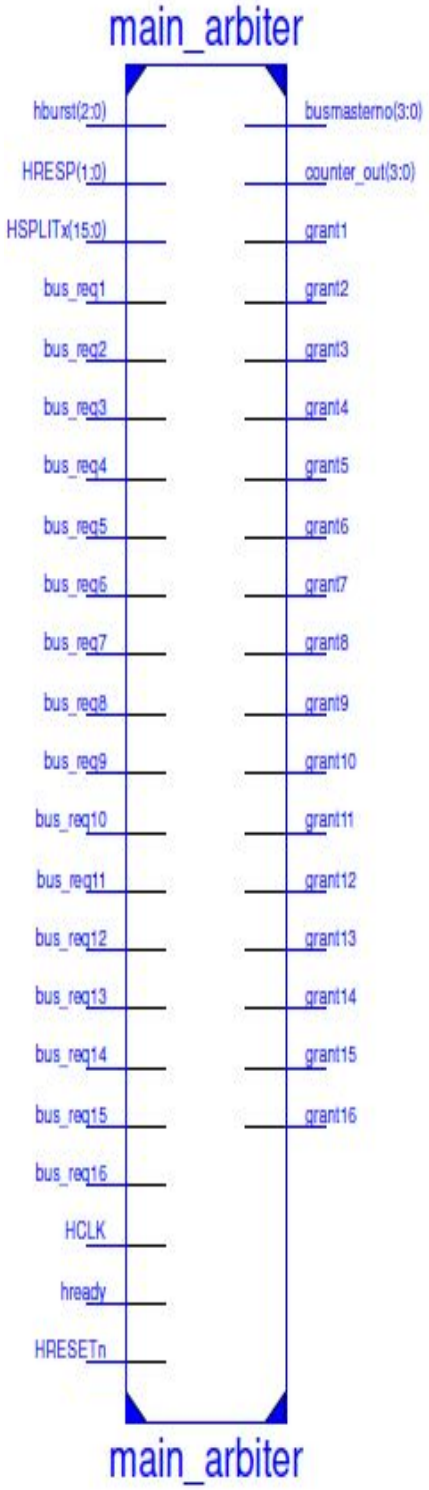
To validate coding, we will use Modelsim 6.5 e for simulation. Simulation of Main arbiter, counter interface, Encoder. Priority logic will give wave form based on selected input.

Input can be changed randomly to see corresponding changes in waveform. This will insure proper working of code and subsequently verify functionality of system also.

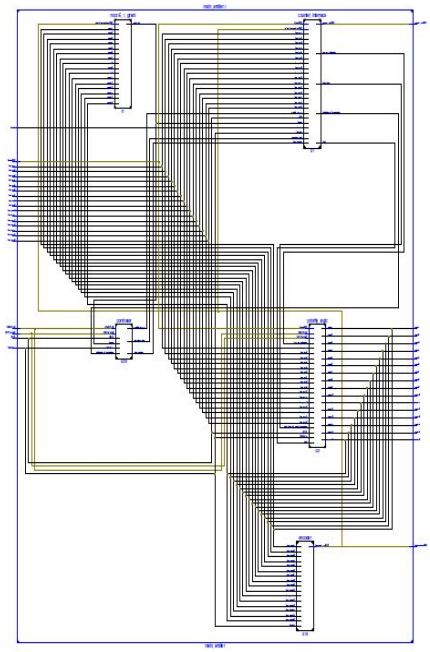
Modelsim 6.5 e is very powerful tool to see such kind of waveform and useful for wave analysis.

Arbiter:

PIN :



Pin diagram for Arbiter



RTL view for Arbiter

Synthesis Report:

Device utilization summary:

Selected Device: 4vlx15sf363-12

Number of Slices:	1329	out of 6144	21%
Number of Slice Flip Flops:	894	out of 12288	7%
Number of 4 input LUTs:	2345	out of 12288	19%
Number of IOs:	64		
Number of bonded IOBs:	64	out of 240	26%
IOB Flip Flops:	4		
Number of GCLKs:	2	out of 32	6%

Timing Summary:

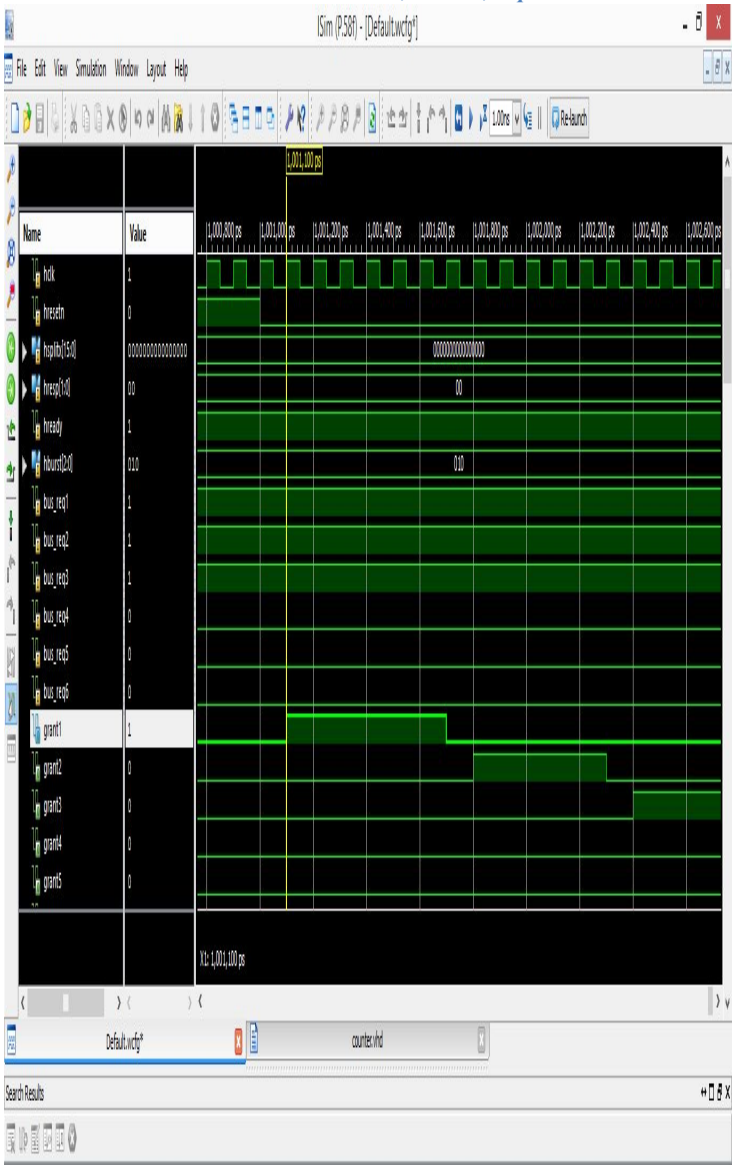
Speed Grade: - 12

Minimum period: 3.130ns (Maximum Frequency: 319.519MHz)

Minimum input arrival time before clock: 6.164ns

Maximum output required time after clock: 5.388ns

Simulation

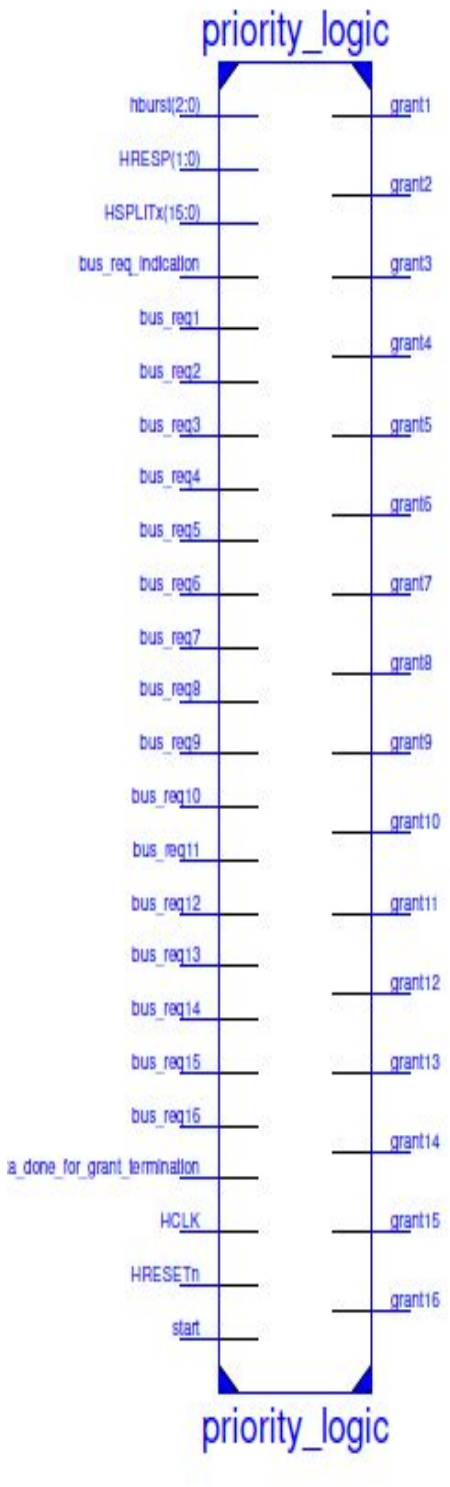


Simulation of integrated arbiter

The above figure depicts the functionality of arbitration, where three requests occurs simultaneously and grant1 gets first access, once the operation of Master 1 is over then grant is shifted to request 2 and so on accordingly.

Priority Logic:

PIN :



Pin diagram for priority logic

Synthesis Report:

Device utilization summary:

Selected Device : 4vlx15sf363-12

Number of Slices:	1369	out of 6144	22%
Number of Slice Flip Flops:	879	out of 12288	7%
Number of 4 input LUTs:	2381	out of 12288	19%
Number of IOs:	58		
Number of bonded IOBs:	58	out of 240	24%
Number of GCLKs:	2	out of 32	6%

Timing Summary:

Speed Grade: -12

Minimum period: 1.503ns (Maximum Frequency: 665.203MHz)

Minimum input arrival time before clock: 6.616ns

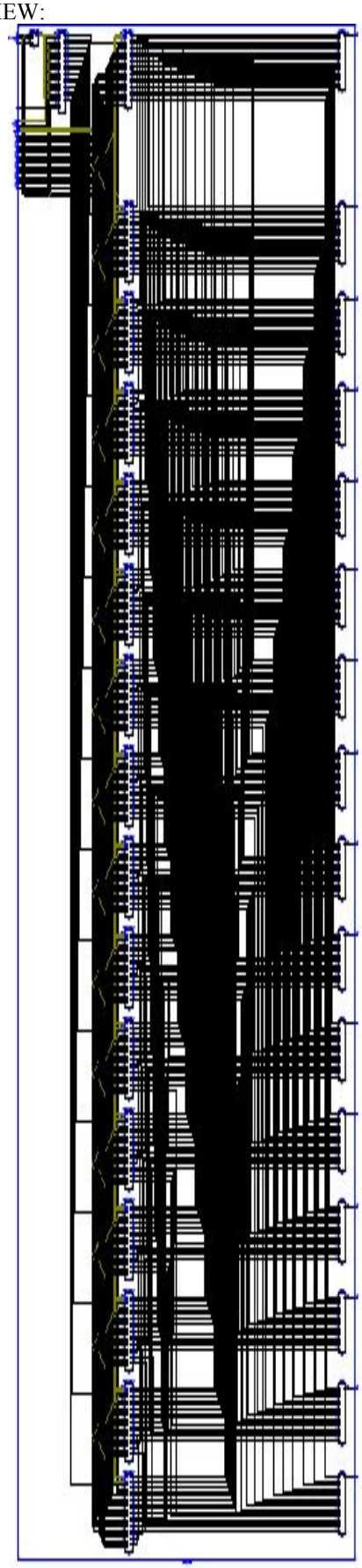
Maximum output required time after clock: 5.294ns

Priority Shift

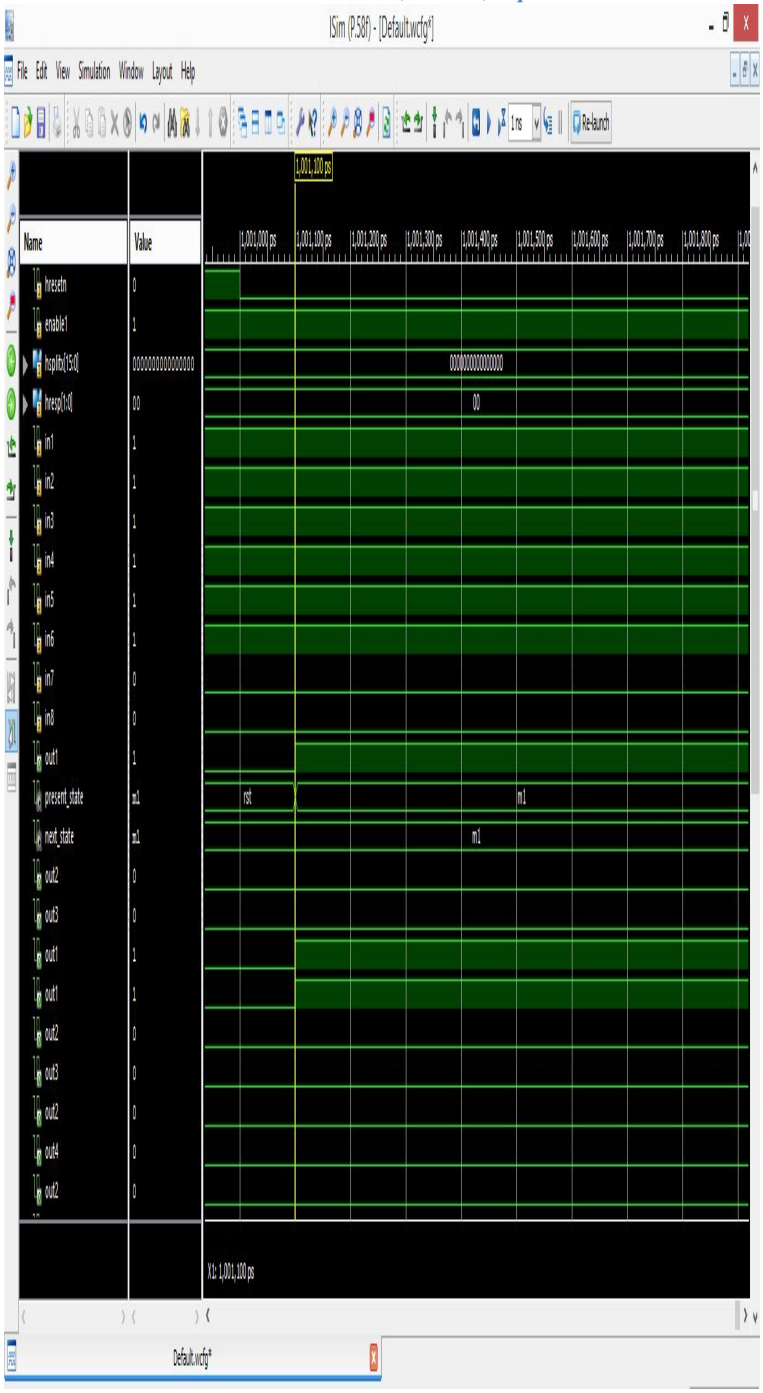
The priority shift block is responsible to store the priority level. It is the block which is responsible to enable the priority Block. It is a shift register with initial value of "0000000000000001".At the start of operation master 1 will get enable if it is requesting the bus. According to data done the shift will occur. For next shift the enable 2 will operate based on the shift value ""0000000000000010".

Device utilization summary gives report of total percentage utilization; here we can see number of flip flop, number of I/O ,Number of LUT's etc

RTL VIEW:



RTL view of priority logic



Simulation of priority logic block

The above figure depicts the simulation of priority which is implemented as finite state machine. There are 16 bus requests coming from different master, and according to priority particular master will get grant. There are 16 state define the priority, in the above simulation in1 has highest priority hence it goes to state M1 and master 1 will get grant. Enable signal is there for shift of priority. The defined priority block will execute only when respective enable is high.

V. 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 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 an AHB-ARBITER for AMBA at a very innovative phase and next to prototyping. The design has been developed using VHDL code and synthesized using Xilinx ISE 14.2 ISE. The design is simulated on Modelsim 6.5 & verified through effective test bench. The advantage of this design is that we have taken care of latch formation, as it is a FPGA implementation hence with fewer latches & maximum flip-flop have enhanced our area efficiency.

VI. FUTURE WORK

Future scope of work is to interface various data's including images. Also data can be transmitted through alternate devices such as Generators. Further result of 16 channels can optimize by selecting random devices through alternate ways. Result for the same can be analyzed in utility data

VII. REFERENCES

[1] Shraddha Divekar, Archana Tiwari, "Multichannel AMBA ARB with Multiple Arbitration Technique", IEEE, International Conference on Communication and Signal Processing, April 3-5, pp. 1854-1858, 2014.

[2] Mrs. R. Ramya, Ms. Preethi Vadivel, "Slave-Side Arbitration and Implementation For Multilayer - AHB Bus Matrix", International Journal of Engineering Research and Reviews, ISSN 2348-697X, Vol. 2, Issue 3, pp: 56-60, September 2014

[3] Shashidhar R., Sujay S. N., Pavan G. S., "Implementation of Bus Arbiter Using Round Robin Scheme", International Journal of Innovative Research in Science, Engineering and Technology, ISSN: 2319-8753, Vol. 3, Issue 7, July 2014.

[4] Pravin S. Shete, Dr. Shruti Oza, "Design of an AMBA AHB Reconfigurable Arbiter for On-chip Bus Architecture", International Journal of Application or Innovation in Engineering & Management (IJAIEM), ISSN 2319 – 4847, Vol. 3, Issue 5, May 2014.

[5] Jalle, Javier, Jaume Abella, Eduardo Quinones, Luca Fossati, Marco Zulianello, and Francisco J. Cazorla. "AHRB: A high-performance time-composable AMBA AHB bus." IEEE 20th Symposium in Real-Time and Embedded Technology and Applications (RTAS), , pp. 225-236, 2014

[6] Dr. Fazal Noorbasha, B. Srinivas, Venkata Aravind Bezawada, V. Sai Praveen, "Implementation of an Adaptive-Dynamic Arbitration Scheme for the Multilayer AHB Busmatrix", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue 4, pp.825-831, July-August 2012

[7] E. Raja, K.V. Ramana, "Implementation of Multilayer AHB Bus matrix for ARM", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2011.

[8] Shanthi, D., and R. Amutha, "Design Approach to Implementation Of Arbitration Algorithm In Shared Bus Architectures (MPSoC)", Computer Engineering and Intelligent Systems 2, no. 4, pp. 185-196, 2011.

[9] Huang, Yu-Jung, Yu-Hung Chen, Chien-Kai Yang, and Shih-Jhe Lin. "Design and implementation of a reconfigurable arbiter", In 7th WSEAS International conference on signal, speech and image processing (SSIP'07), Beijing, China. 2007.

[10] 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