Implementation of Amba AHB Bus Arbiter using Amalgamate Algorithm

Gulab Singh Yadav[1] Deepika Soni[2]
[1]M. Tech. Student [2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Gyan Ganga Institute of Technology Jabalpur, India

Abstract— On Chip Bus Architecture play a very important role in SOC system. Now days there are many on chip bus architecture are provided by different company. One of the most popular is advance microcontroller bus architecture which is popularly known as AMBA 1.0. AMBA have an advantage that it is open specification i.e. AMBA serve as a framework for system on chip design. In System On chip, problem arises when number of master trying to sense a single data bus. Then resolutions of a required become a big issue. The overall system performance depends upon the ability of system to resolve this resolution problem. The AMBA protocol use logical assignment of chance to different masters according to their priority to take over the bus for data transmission. AMBA architecture defines three specifications. They are advance high performance bus (AHB), advance peripheral bus (APB) and advance system bus (ASB). Among all three specification AHB have high bandwidth and this make AHB as first choice for system designer. While resolving the problem of priority resolution, arbiter plays an important role. Arbiter is a digital circuit and it working is depends upon the arbitration algorithm used. According to arbitration algorithm, arbiter decides to give grant to heights priority master to access bus. In this project we design arbitration algorithm, according to it arbiter give grant to different masters.

Key words: Amalgamate Algorithm, Amba AHB Bus Arbiter

I. INTRODUCTION

A. System on Chip

Ongoing research work done by semiconductor industry make an impressive improvement, now very high density in a single chip can be achieved. New methodologies and techniques are developed by design engineers in order to match with market demand and also to manage increase complexity. Now electronics device are became very small and thinner day by day which increase the complexity, also they support the large varieties of operation. In order to facilitate the demand of market the designers have to developed new technique.

One of the immerging techniques is SOC (system on chip). In SOC methodology designer use predesigned block which are internally obtained by internal sources or third parties, this entire block are integrated in a single chip. In SOC design one or more microprocessor, memory, peripheral device are integrated in a single chip and makes a complete system. Microprocessor or other device can be standard device or it can be custom as per demand.SOC play a very vital rule in today digital life it is found in almost every consumer product.

B. On Chip Bus Architecture

As in SOC design, microprocessor, memory, other peripheral devices are integrated in a single chip, hence required communication medium in order to perform communication between all devices. The objective of communication between the devices is performed by On Chip Bus Architecture. On Chip Bus Architecture connected all the devices together, the overall functional performance of SOC system is depending upon the performance of On Chip Bus Architecture. The SOC system performance is primary determinant by the way, how all devices exchange and synchronized their data with one another. The On Chip Bus Architecture defines the way how the communication takes place between different devices and hence it has a great impact on overall system performance. As increase in the number of device block in SOC, amount of communication between the blocks is also increase and become the performance bottle necks. On Chip Bus Architecture has to satisfy all interface behavior of each device block within the SOC system design. Today in market many commercial organization define their own communication architecture with their own bus protocol .for example AMBA bus of ARM core connect of IBM etc.

The AMBA of ARM is an open specification for high performance buses on low power devices. Take care of capability between different device block connected in SOC is basic functionality of AMBA protocol, it make sure that device block will perform as their vender defined.

C. Amba Bus

The Advanced Microcontroller Bus Architecture protocol describes number of buses and interfaces. In its first version which advance system bus and in its second version it defines AMBA 2.0, which is a high performance bus. Advance high performance bus is a signal clock edge protocol. In its third generation version it defines AMBA 3.0 which include AXI. The first version of AMBA contains only one peripheral bus with two system bus. In current version of AMBA protocol is wide range of high performance buses and interface are present.

The AMBA defined three bus specifications they are as follow-

- Advanced High- Performance Bus (AHB).
- Advanced System Bus (ASB).
- Advanced Peripheral Bus (APB).

In present senior all specification are widely use.ASB is oldest and AHB being introduced later in order to support for higher performance. In AHB/ ASB module APB is generally used as secondary bus. Typically AMBA bus structure is shown in Fig 1.

D. Advanced High Performance Bus

AMBA 2.0 defines the advanced high performance bus.AHB have high bandwidth which make AHB is best choice among ASB and APB. The property required for high performance, high clock frequency systems are as follows.

- Burst transfers (4/8/16 bit burst)
- Split transactions
- Bus master handover in single cycle

– Single clock edge operation

E. Advanced System Bus

AHB is a high performance bus. Some application where high performance features of AHB is not required there we uses ASB bus. ASB is also high performance bus use as alternatives for AHB. Some property of ASB is as follows.

– High performance
– Pipelined operation
– Multiple bus masters

F. Advanced Peripheral Bus

APB design to supports low-power peripherals devices. APB is generally used as secondary bus. APB is utilization for minimum power consumption and reduces complexity for interface. Main further of APB is as follows.

– Low power.
– Latched address and control.
– Simple interface.
– Suitable for many peripherals.

G. Property of Amba AHB

To under the proper functioning of AHB it is compulsory to understand the properties of AHB bus system.

– AHB support burst transfers.
– Split transaction by split capable slave is also support by AHB.
– AHB provide single-cycle bus master handover.
– Single-clock edge operation is done in AHB.
– Non-tristate implementation is done in AHB.
– Wider data bus configurations it support 64 and 128 bits.

H. Arbiter

Arbiter is a digital circuit. The main operation of arbiter is to grant access to master to shared resource. Arbiter block play an important role in SOC bus architecture. In SOC design many master are integrated in a single chip and they all try to access bus. The problem arises when two or more master want to access bus at same time. Then arbiter decides which master get grant to access bus and forces the other master to remain in ideal states. The process of choice or provide grant to master is according to arbiter algorithm. Arbiter is important block in SOC design as it efficiently handles the requests from all master and also responses of all slave devices. Arbiter ensures that master requirement should be fulfilled. For example in some application master request real time or bandwidth requirement, then arbiter ensure that transaction accomplished with in fixed number of cycle for real time requirement and for fixed bandwidth requirement it ensure that master must occupy a fixed fraction of bandwidth of bus.

The arbiter algorithm could be implemented in two ways. They are centralized and distributed. In distributed arbitration algorithm slave side arbitration is absent where as in centralized arbitration algorithm it is present. To handle the configuration the arbitration algorithm must be optimized. In soc design for particular application power utilization by arbitration technique varies significantly.

I. Arbiter

Arbiter is a digital circuit. The main operation of arbiter is to grant access to master to shared resource. Arbiter block play an important role in SOC bus architecture. In SOC design many master are integrated in a single chip and they all try to access bus. The problem arises when two or more master want to access bus at same time. Then arbiter decides which master get grant to access bus and forces the other master to remain in ideal states. The process of choice or provide grant to master is according to arbiter algorithm. Arbiter is important block in SOC design as it efficiently handles the requests from all master and also responses of all slave devices. Arbiter ensures that master requirement should be fulfilled. For example in some application master request real time or bandwidth requirement, then arbiter ensure that transaction accomplished with in fixed number of cycle for real time requirement and for fixed bandwidth requirement it ensure that master must occupy a fixed fraction of bandwidth of bus.

The arbiter algorithm could be implemented in two ways. They are centralized and distributed. In distributed arbitration algorithm slave side arbitration is absent where as in centralized arbitration algorithm it is present. To handle the configuration the arbitration algorithm must be optimized. In soc design for particular application power utilization by arbitration technique varies significantly

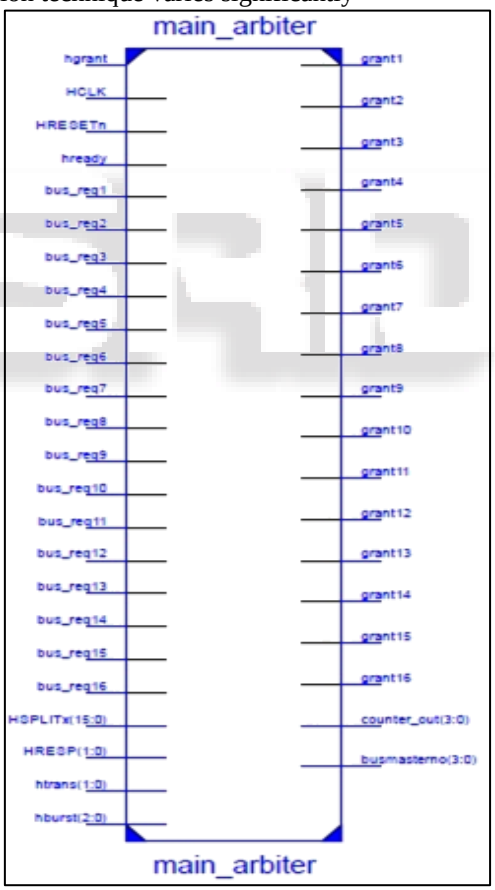

Fig. 1: RTL view of arbiter

II. PROPOSED METHODOLOGY

Arbiter selected the heights priority master based on arbitration algorithm used. Arbitration algorithm use round robin or other scheme to choose next bus master. Round robin arbitration is overridden when master has locked the bus and retain highest priority. For next bus master, arbiter block monitors all master requests and choose high priority request and if arbiter not find any request it give grant to default

master to access the bus for next transaction. The controller which is mealy state machine keep track of all transaction of all different stations and it first state is start state and in next state it check the grant signal and if it is high then it make necessary signal high for further block interface.

The bus-req blocks pass the entire master bus request through it to other logical block. The priority storage block enables the bus-req block. The priority logic is enable by interface block through enable signal, also interface block is responsible for monitoring of data transaction, it monitor it by using data done signal and based on result it assert and dessert the enable pin.

When priority logic block received bus request it further decided that which master have highest priority and give grant to it by generating grant signal. This grant signal interacts with master by passing through priority storage block, encoder block and out-put port, then master will send address, burst signal which define the type of transfer. After getting grant signal it responses to it slave. Arbiter grant signal pass through multiplexer and bus master number will serve as select line for multiplexer which indicate master which accessing the bus. Then mux output is given to controller block which will generate necessary signal for counter block .The output grant signal from priority logic block is pass through OR gate and send to priority storage block .Depend upon the grant signal priority storage block pass the enable signal to next priority and depend upon the transaction mode ,whole operation repeated itself.

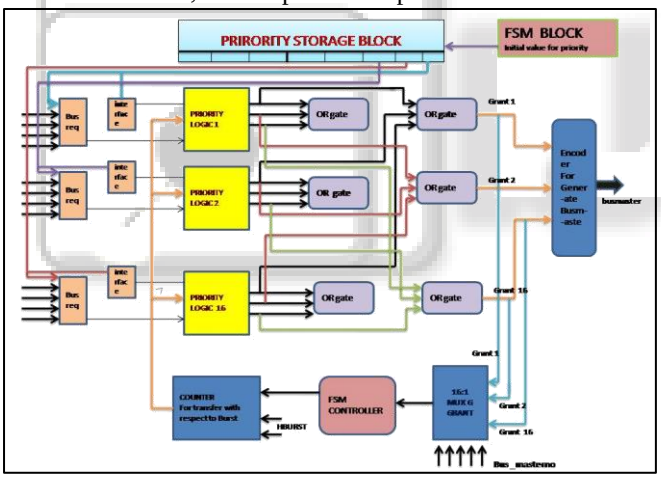

Fig. 2: Proposed methodology

III. SYNTHESIS AND SIMULATION

Synthesis and simulation done in MODELSIM 6.5, implementation is done in Xilinx using vetex4 IC number xc4vfx12.The speed of device is 9.516ns and max frequency 105.092MHz.

A. Devices Utilization Summary

- Selected devices: vetex4 IC number xc4vfx12
- Number of slices: 2183
- Number of flip flops: 858
- Number of 4 inputs LUTs: 3779
- Number of IOBs: 64
- Number GCLKS: 2
- Speed: Minimum period: 9.516ns
- Maximum Frequency: 105.092MHz
- Minimum input arrival time before clock: 20.167ns
- Maximum output required time after clock: 7.516ns

Figure 3 show the simulation result table in Xilinx platform

main_arbiter Project Status			
Project File:	arbitor4.0.xise	Parser Errors:	No Errors
Module Name:	main_arbiter	Implementation State:	Synthesized
Target Device:	xc4vfx12-12ff363	•Errors:	No Errors
Product Version:	ISE 14.5	•Warnings:	486 Warnings (486 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2183	5472	39%
Number of Slice Flip Flops	858	10944	7%
Number of 4 input LUTs	3779	10944	34%
Number of bonded IOBs	64	240	26%
Number of GCLKs	2	32	6%

Fig. 3: Simulation Result Summary

IV. CONCLUSION

AMBA AHB proposed arbiter is capable to perform arbitration process based on the combination of round robin and fixed priority algorithm. The proposed AHB Arbiter is design using vetex4 IC no.xc4vfx12. In reference paper four masters is used whereas proposed arbiter can handled the request of sixteen masters hence over all device utilization is increased respectively in different manner. The proposed arbiter gives grant to master according to the arbitration algorithm. From the simulation result it is found that the overall speed of device is 9.516ns and maximum frequency 105.092 MHz.

V. FUTURE WORK

The proposed AHB Arbiter is capable of handling the requested of sixteen master. The arbitration algorithm is combination of Round Robin and Fixed Priority algorithm. The further research work can be done as there are nine possible combination of arbitration for example combination of Fixed Priority algorithm and Dynamic Priority algorithm, combination of Round Robin algorithm and Dynamic Priority algorithm etc.

REFERENCES

[1] https://www.arm.com/products/amba-open-specifications.php
[2] S. Divekar, Archana Tiwari "Multichannel AMBA ARB with Multiple Arbitration Technique",IEEE trans,published in International Conference on Communication and Signal Processing, April 3-5,2014
[3] D.Shanthi and R.Amutha "Design Approach to Implementation of Arbitration Algorithm in Shared Bus Architectures (MPSoC)" published in Computer Engineering and Intelligent Systems Vol 2, No.4, 2011
[4] E. Raja and K.V. Ramana "Implementation of Multilayer AHB busmatrix for ARM" published in International Journal of Soft Computing and Engineering, Volume-1, Issue-5, November 2011
[5] Shashidhar R., Sujay S.N and Pavan G.S "Implementation of Bus Arbiter Using Round Robin Scheme" paper published in International Journal of Innovative Research in Science, Engineering and Technology Vol. 3, Issue 7, July 2014

[6] Pravin S. Shete, Shruti Oza "Design of an AMBA AHB Reconfigurable Arbiter for On-chip Bus Architecture" published in International Journal of Application or Innovation in Engineering & Management Volume 3, Issue 5, May 2014
[7] F. noorbasha, b.srinivas, venkata aravind bezawada, v.sai Praveen "Implementation Of An Adaptive-Dynamic Arbitration Scheme For The Multilayer Ahb Busmatrix" paper published in International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue 4, July-August 2012, pp.825-831
[8] R.Ramya, and Preethi Vadivel "Slave-Side Arbitration and Implementation For Multilayer - AHB Bus Matrix" paper published in International Journal of Engineering Research and Reviews Vol. 2, Issue 3, pp: (56-60), Month: July - September 2014,
[9] A. K. Singh,Anurag Shrivastava and G.S. Tomar "Design and Implementation of High Performance AHB Reconfigurable Arbiter for Onchip Bus Architecture" in 2011 International Conference on Communication Systems and Network Technologies page 456-459
[10] D. N. Kashinath and S.I. Nipanikar "AMBA Bus with Multiple Masters Using VLSI" paper published in International Journal of Engineering Development and Research Volume 3, Issue 2
[11] S.D. Girdhar N.k. Shukla "Design and Verification Of AMBA APB Protocol" paper published in international journal of computer applications (0975 – 8887) volume 95– no.21, june 2014
[12] P.Harishankar,Chusen Duari and Ajay ShARMa "Design and synthesis of efficient FSM for master and slave interface in AMBA AHB" paper published in International Journal of Engineering Development and Research Volume 2, Issue 3
[13] Mohit N. Kandiya,M.K. Harniya and Prof. Kishan K. Govani "Implementation of Read/Write operation for AMBA AXI4 Bus using VHDL".
[14] S. Aithal ,J.S. Baligar , Guruprasad S.P "FPGA Implementation of AHB to APB Protocol" paper published in International Journal Of Engineering And Computer Science Volume 5 Issue 5 May 2016
[15] M. Chaithanya "Building an AMBA compliant memory controller using AHB protocol" paper published in International Journal of Advances in Engineering & Technology, Sept.2014
[16] Pravin S. Shete and Shruti Oza "Design of an efficient fsm for an implementation of AMBA ahb master" paper published in International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 3, March 2014
[17] Muzammel Hoque and Owais Shah "Design and Implementation of Efficient FSM for AHB Master and Arbiter" paper published in International Journal of Science and Research Volume 4 Issue 6, June2015

