International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-1, June 2013

A Dynamic Priority Based Arbitration Algorithm

Sandra Mohan, Anish Joseph

Abstract— Today's electronic industry consists of chips with multimillion gates. This new level of integration on a single chip is called the System on Chip (SoC) design. In an SoC, on-chip interconnection networks are mostly implemented using buses. The performance of the SoC design heavily depends upon the efficiency of its bus structure. The bus used in the SoC platform requires an arbitration process since multiple components connected to it can act as masters and hence initiate a transaction. As the number of system components in SoC design increases, it becomes that an efficient arbiter is one of the most critical factors for high system performance. This paper deals with an Advanced High-performance Bus (AHB) arbiter with a dynamic arbitration mechanism.

Index Terms— AHB, AMBA, Arbiter, SoC.

I. INTRODUCTION

A typical System-on-Chip (SOC) design contains many different cores linked together with complex on chip bus communication architectures. The on-chip bus communication architecture determines how these different functional units exchange and synchronize data. This has a great impact on the system's performance. The SOC design standard relies on well-defined interfaces and reuse of Intellectual Property (IP). The amount of communication between the IP's increases and this becomes the source of the performance bottlenecks as more and more IPs gets integrated into the design platform. On the other hand the arbiter plays a very important role to manage the resource sharing on the SOC platform.

With the increasing number of system components in SOC design, it becomes that an efficient arbiter is one of the most critical factors for high system performance. Most existing buses have their own special protocols. The communication architectures defined by commercial standards are widespread and available in the market. For example, the PI-Bus of OMI, the AMBA bus of ARM, the FISP bus of Mentor Graphics, the CoreConnect of IBM, the Silicon Backplane of Sonics, the Wishbone of Silicore and others. The CoreConnect and AMBA make use of a fixed priority arbiter. Although the arbitration protocol is fixed, the choice of an arbitration scheme usually depends on the application requirements. The most frequently used on-chip interconnect architecture is the shared medium arbitrated bus, where all the peripheral devices share the same system bus.

Manuscript published on 30 June 2013.
***Correspondence Author(s)**
Mrs. Sandra Mohan, Electronics and Communication Engineering Department, Sree Buddha College of Engineering, Alappuzha, India.
Mr. Anish Joseph, Technical Staff Engineer, DSipher Design Solutions Pvt Ltd, Bangalore, India.

The bus used in the SOC platform requires an arbitration process since multiple components connected to it can act as masters and hence initiate a transaction and in order to avoid collision the arbitration scheme is necessary co that only one device transmits at any given time. The Advanced Microcontroller Bus Architecture (AMBA) is an open System-on-Chip bus protocol for high-performance buses on low-power devices. The AHB is a pipelined system backbone bus, designed for high-performance operation. It can support up to 16 bus masters and slaves that can delay or retry on transfers. It consists of masters, slaves, an arbiter and an address decoder. It supports burst and split transfers. The address bus can be up to 32 bits wide, and the data buses can be up to 128 bits wide. The AMBA uses conventional fixed priority arbiter [5].

Thus by implementing an efficient arbitration algorithm, the system performance can be tuned to suite the applications better. This paper presents a centralized arbiter with dynamic priority based arbitration mechanism. The rest of this paper is organized as follows. A brief description about the arbiter and different arbitration mechanisms are described in the next section. The design and implementation results of the newly proposed dynamic priority arbitration mechanism are described in Section III. Finally the paper is concluded in Section IV.

II. ARBITRATION MECHANISMS

An arbiter is a logical element serving to select the order of access to a shared resource. An arbiter would typically employ a scheduling algorithm to decide which one on several requestors would be serviced. Arbiter provides arbitration between bus masters that are competing for access to the bus. AMBA bus specification is a multi-master bus standard. A bus arbiter is needed to ensure that only one bus master has access to the bus at any particular point of time. Each bus master requests control of the bus and the arbiter decides which master should be granted accordingly.

Every system consists of large number of requesters that need to access a common resource. Thus an arbiter is essential to determine the resource sharing among different requesters. Properly designed arbitration policies are adopted to suit the bus access demands of the system bus masters. The arbitration policy refers to the algorithm or logic by which the arbiter decides to give the grant or access to the bus for the requester when multiple masters request the bus simultaneously. The arbitration policy also decides what to do when none of the masters are accessing the bus.

A. Related Works

A high fairness arbitration mechanism is proposed for shared bus architecture in a system-on-chip, and analyzed in terms of bandwidth and throughput in the paper [1]. The proposed scheme is compared with other arbitration mechanisms, such as static priority arbitration and round-ring priority arbitration.

A reconfigurable arbiter with various combinations of arbitration algorithms is presented in [2]. The performance analysis for the various combinations of the arbitration algorithms under different traffic loads is simulated. High performance communication architecture, SAMBA bus, is proposed in the paper [3]. In SAMBA-bus architecture, multiple compatible bus transactions can be performed simultaneously with only a single bus access grant from the bus arbiter.

The multilayer advanced high-performance bus (ML-AHB) bus matrix employs slave-side arbitration [4]. Slave-side arbitration is different from master-side arbitration in terms of request and grant signals since, in the former, the master merely starts a burst transaction and waits for the slave response to proceed to the next transfer.

B. Fixed Priority Arbitration

This algorithm is a commonly used arbitration mechanism on most common buses. Here each master is assigned a fixed priority value. When several masters request simultaneously, the master with the highest priority will be granted. The advantage of this arbitration mechanism is that it is easy to implement and have small area cost. The fixed priority based architecture does not provide a means for controlling the fraction of communication bandwidth assigned to a component. If masters with high priority requests frequently, it will lead to the starvation of the one with low priority. i.e. in heavy communication traffic, master that has low priority value cannot get a grant signal.

C. Round Robin Arbitration

The round robin arbitration, in its basic form, is a simple time slice scheduling, allowing each requestor an equal share of the time in accessing a memory or a limited processing resource in a circular order. A round robin arbiter allows every requester to take a turn in order. The advantage of the round robin arbitration is even access of the shared resource among different masters. However this method of arbitration has some disadvantages. It doesn't give special priority to more important request.

D. Dynamic Priority Arbitration

To overcome the disadvantages of Round Robin and fixed Priority algorithm, a new arbitration mechanism with dynamic priority is introduced. In Fixed Priority arbitration and Round Robin Arbitration, there is no provision to adjust the priority of the requesters during run time. In Dynamic Priority Arbitration, the default priorities of the requesters can be changed during the run time. The changed priority will override the default priority.

III. DESIGN AND IMPLEMENTATION OF DYNAMIC PRIORITY ARBITRATION

The flow chart for Dynamic Priority arbitration for a single request is shown in Fig 1.

Initially the request lines are read. If any of the request lines are high, then corresponding priority values of the master that is requesting is checked. And if the priority value is the lowest one, the grant is given to that particular master. Otherwise the count value is incremented. If the count reaches the preset value, the priority value of the corresponding request line is decremented.

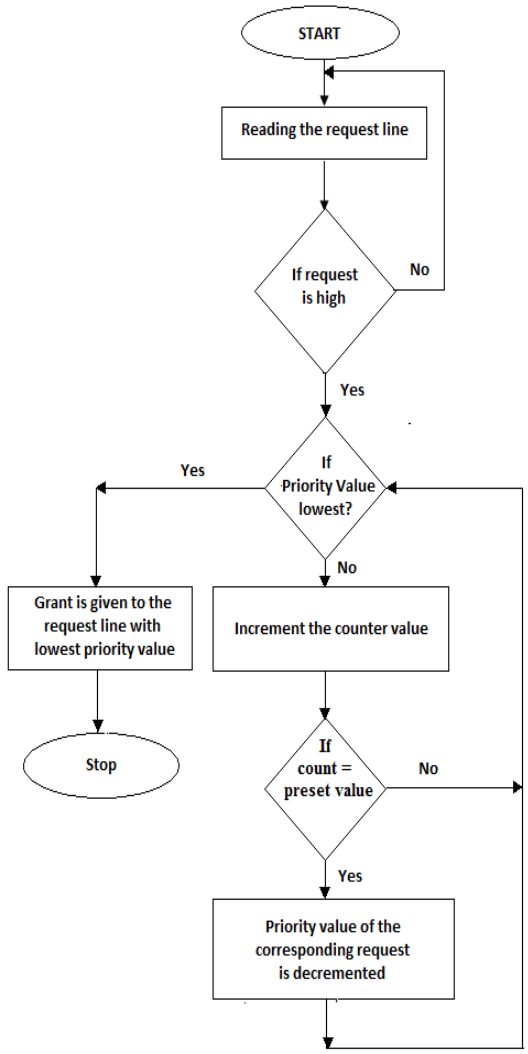

Fig.1. Flow chart for Dynamic Priority Arbitration for a single request

The dynamic priority based arbitration scheme has the advantage for throughput when there are few masters with long job lengths in a system.

A. Simulation Result

The simulation result of dynamic priority arbitration algorithm is shown in Fig 2. Here the request lines 3, 5 and 7 are high. The priority values of these request lines are 3, 5 and 7 respectively. The request lines are considered in the ascending order. Here the request line 3 is considered first.

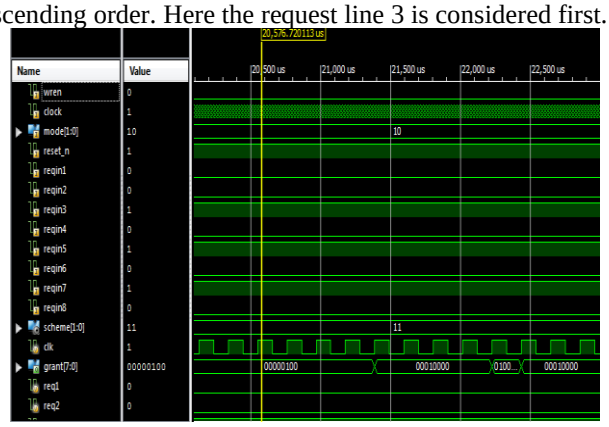

Fig.2. Simulation Result of Dynamic Priority Arbitration

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The priority value is dynamic. Priority is changed according to the waiting time. If a particular request line is high then the corresponding count is incremented and when the count reaches the preset value, the priority value of the corresponding request is decremented, that is the priority of the request is increased.

B. Implementation Details

The software tool used for the simulation purpose is ISim Simulator from Xilinx ISE Design Suite 13.2. The output is verified with Spartan 3E FPGA. The Device Utilization summary is shown in the Table I.

Table I. Device Utilization Summary

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of slices	267	960	27%
Number of Slice Flip Flops	140	1920	7%
Number of 4 input LUTs	514	1920	26%
Number of bonded IOBs	25	66	37%
Number of GCLKS	2	24	8%

IV. CONCLUSION

The performance of the system mainly depends upon the design of the communication architecture. Shared bus architectures have gained wide popularity. Therefore the need for the arbiters is inevitable. Arbiters are used for deciding which master to be granted the access to the bus in case of multiple simultaneous bus requests. This dynamic priority arbiter concept described in this paper has received increased attention due to the flexibility. This arbiter can be custom-tuned to obtain high bandwidth utilization, low latency, and power effective for on-chip bus communication.

REFERENCES

1. Jisuhn Suh, Jongsun Kim, Hoi-Jun Yoo, “An Analysis and Implementation of High Fairness Arbitration Mechanism by Using Level-table and Static Priority Orders in Shared Bus Architecture” IP Based SoC Design 2003 - November 13-14, 2003.
2. Yu-Jung Huang, Yu-Hung Chen, Chien-Kai Yang, And Shih-Jhe Lin, “Design and Implementation of A Reconfigurable Arbiter” Proceedings of the 7th WSEAS International Conference on Signal, Speech and Image Processing, Beijing, China, September 15-17, 2007
3. Ruibing Lu, Aiqun Cao, and Cheng-Kok Koh, Senior Member, IEEE, “SAMBA-Bus: A High Performance Bus Architecture for System-on-Chips” IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 1, January 2007.
4. Soo Yun Hwang, Dong Soo Kang, Hyeong Jun Park, and Kyoung Son Jhang, Member, IEEE, “Implementation of a Self-Motivated Arbitration Scheme for the Multilayer AHB Bus matrix”, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 5, May 2010.
5. ARM, “AHB Example AMBA System,” 2001 [Online].Available:http://www.arm.com/products/solutions/AMBA_Spec.html

AUTHOR PROFILE

Mrs. Sandra Mohan completed B.Tech degree from Univerity of Kerala in the year 2007.Currently doing M.Tech in Embedded Systems under Univerisity of Kerala.

Mr. Anish Joseph completed B.Tech degree from Calicut University in the year 2008 and M.Tech from M.G university Kottayam in VLSI and Embedded Systems. Currently working as the technical staff engineer at Dsipher Design Solutions Bangalore.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)