An Approach for Implementation of Bus Arbitration Techniques

Thakkar Abhishek D.[1] Ketan N. Patel[2]
[1,2]Department of Electronics and Communication Engineering
[1,2]Vishwakarma Government Engineering College, Chandkheda-382424, Gujarat, India

Abstract— In modern days, due to down scaling in technologies have led to highly complex billion transistor integrated circuits (ICs).As a consequence, manufacturers are integrating increasing number of components on a chip. A heterogeneous System on Chip (SoC) might include one or more programmable components such as general purpose processor cores, digital signal processor cores, or application specific intellectual property cores (IPs) as well as an analog front end, On-chip memory, I/O devices and other application specific components. In SoC, multiple IPs need to communicate with each other to access the required functionality. When multiple IPs requests the bus at the same time, contention may occur. This makes on-chip bus based communication a major challenge for the system designer in the current SoC technology. Thus, Bus arbiters are proposed. The masters on a SoC bus may requests simultaneously and hence an arbiter is required to decide which master is granted. To do so different algorithms are developed. In this paper an approach for implementation of bus arbitration techniques like RR, FCFS and Token Pass are discussed.

Key words: Bus arbitration, Round Robin, FCFS, Token Pass, Weighted Round Robin

I. INTRODUCTION

Modern System-on-Chip (SoC) architectures comprise several components such as master and slave modules. Masters are active modules such as DMAs, hash generators or graphics engines that send read requests or data to memories. Typical masters are CPUs and hardware accelerators. Slave modules are passive components e.g. memories, on chip buses or simple register banks that react on requests and store data or respond to master requests with appropriate data.Memories can be distinguished into on-chip and off-chip memories. On Chip memories feature low latencies but small capacities whereas the off-chip memories exhibit high latency with high capacities.

When the SoC bus is connected with two or more IPs, IPs need to communicate with each other to access the required functionality. If multiple IPs requests the bus at the same time then contentions occur. This makes on-chip bus based communication a major challenge for the system designer in the current SoC technology. To solve above problem bus arbiters are proposed which plays vital role. The masters on a SoC bus may issue requests simultaneously and hence an arbiter is required to decide which master is granted for bus access. If more number of masters request simultaneously then arbiter must have to allocate bus to each master without suffering from starvation. To do so different algorithms are developed.

II. BASICS OF ARBITER

System in which a large number of requesters require to access a common resources, Arbiters are a fundamental component in systems to access required resource. The common resource may be a complex computational element, a shared memory, a specialized state machine, or a networking switch fabric. An arbiter is required to determine how the resource is shared amongst the many requesters.

Processor will initiate a memory-access signal when it wants access to memory and will deactivate the same when the job is over. If one or more processor request for the bus at a instant, access should be granted on basis of arbitration algorithm. This is in order to ensure that no one independent processor is suffering from starvation while another has continuous access. Continuous access is to be granted to any one processor for a period of time.

While designing the arbiter, many factors must be considered e.g. the interface between the requesters and the arbiter must be appropriate for the size, area, latency, fairness etc. of the arbiter. Also, the coding style used will usually impact the synthesis results [3]. Thus by implementing an efficient arbitration algorithm, the system performance can be tuned to suite the applications better. The basic block diagram representation of an arbiter is shown in below figure1 where N devices are connected to shared resources (i.e. memory elements, I/O devices etc.) through an arbiter.

A. Arbitration Timing

The duration of the grant depends on the application. In some applications either the requester may need uninterrupted access to the resource for a number of cycles or it may be safe to re-arbitrate every cycle. To suit these different applications, we may build arbiters that issue a grant for a single cycle, for a fixed number of cycles, or until the resource is released [6].

B. Fairness

It is a key property of an arbiter. Intuitively, a fair arbiter is one that provides equal service to the different

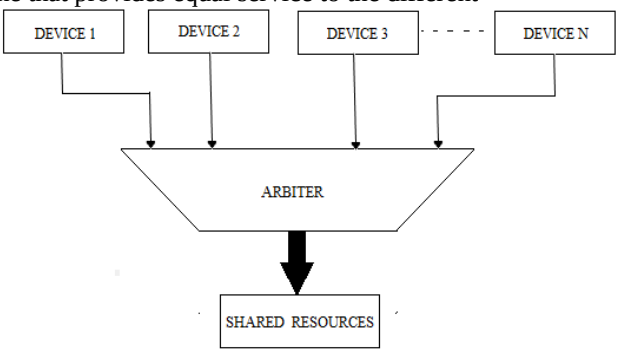

Fig. 1: Arbiter Block Diagram

requesters. However the exact meaning of fairness can vary from application to application. Useful definitions of fairness are:

1) Weak Fairness: Every request is eventually served [6].
2) Strong Fairness: Requesters will be served equally often. This means that the number of times one requester is served will be within n of the number of times some other requester is served when averaged

over a sufficient number of arbitrations, N. Often, we modify strong fairness to be weighted so that the number of times requester is served is proportional to its weight(w_i) [6].

C. Latency

It defines as the time taken by the processor for a value at the input to be processed and the result appear at the output.

III. ARBITRATION TECHNIQUES

Arbitration techniques are basically algorithms that are used for allocation of resources to a particular user among a number of users at the same time. Arbitration techniques are discussed as below [9].

A. Round Robin Arbitration

Round Robin (RR) arbitration is based on rotating priority method. It works on basic principle of time slice scheduling algorithm. It allows each and every device an equal interval of time i.e. equal time slot to access resources in a circular order [6].

It is a scheduling scheme which gives to each requestor its share of using a common resource for a limited time or data

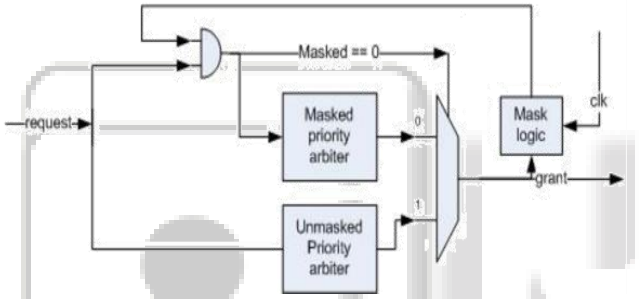

Fig. 2: RR using Mask and Unmask-logic[10]

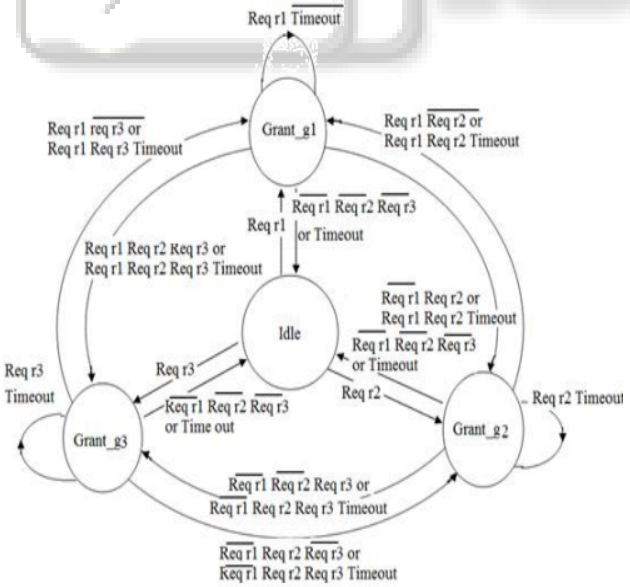

Fig. 3: FSM diagram for RR elements

In round-robin arbiter, a request that was just served should have the lowest priority on the next round of arbitration. This can be accomplished by generating the next priority vector from the current grant vector [6].Block diagram of RR using Mask and Unmask logic is shown in Figure 2.To understand working of RR, Finite State Machine diagram shown in figure 3. The round-robin arbiter exhibits strong fairness.

Some applications require an arbiter that is unfair to a controlled degree, so that one requester receives a larger number of grants than another requester. A weighted round-robin arbiter exhibits such behavior. Each requester i is assigned a weight w(i) that indicates the maximum fraction f(i) of grants that request r(i) is to receive according to following equation [6].

$$f(i) = \frac{W(i)}{W} \qquad \text{where } W = \sum_{j=0}^{n-1} W_j \qquad (1)$$

A requester with a large weight will receive a large fraction of the grants while a requester with a small weight receives a smaller fraction.

B. Lottery Bus Techniques

In this technique, a lottery manager collects request to own the shared communication devices from one or more users. Each user is assigned a static or dynamic lottery tickets. The user which owns the maximum number of tickets will be assigned with the device. Basic block diagram of lottery bus is shown in figure 3 [8].

In order to design multiprocessor system according to their bandwidth requirement, an arbiter is proposed called dynamic arbiter in which a lottery bus algorithm is designed where an arbiter can adjust the bandwidth proportion assigned to every processor automatically due to the situations of bus transactions aiming to reduce total task execution time [7]. Compared with conventional architectures their architecture reduces the system latency but it does not allocate fair bandwidth to the processor

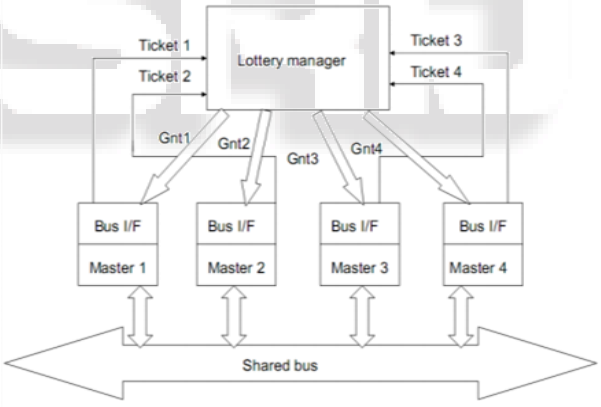

Fig. 4: Lottery Arbiter for shared bus

C. Token Pass Technique

This is special ring based architecture. The token ring technique uses a special data word named as token that circulates over the ring. The device interface receives a token after which transaction is allowed. When transaction is completed, the token is released and sends it to the neighbour element. Figure 4 shows arbiter with grant & hold circuit block diagram.

D. First Come First Serve

First-Come-First-Served algorithm is the simplest scheduling algorithm. FCFS algorithm basically working on First-In-First-Out principle. Processes are served according to their arrival time on the queue. Once a process has a grant, it executes to completion. The FCFS scheduling is fair in the formal sense or human sense of fairness but it is

unfair in the sense that long jobs make short jobs wait and unimportant jobs make important jobs wait[11].

FCFS is more predictable than most of other schemes since it offers time. FCFS scheme is not useful in scheduling interactive users because it cannot guarantee good response time. The code for FCFS scheduling is simple to write and understand. One of the major drawback of this scheme is that the average time is often quite long [11].

The First-Come-First-Served algorithm is rarely used as a master scheme in modern operating systems but it is often embedded within other schemes.

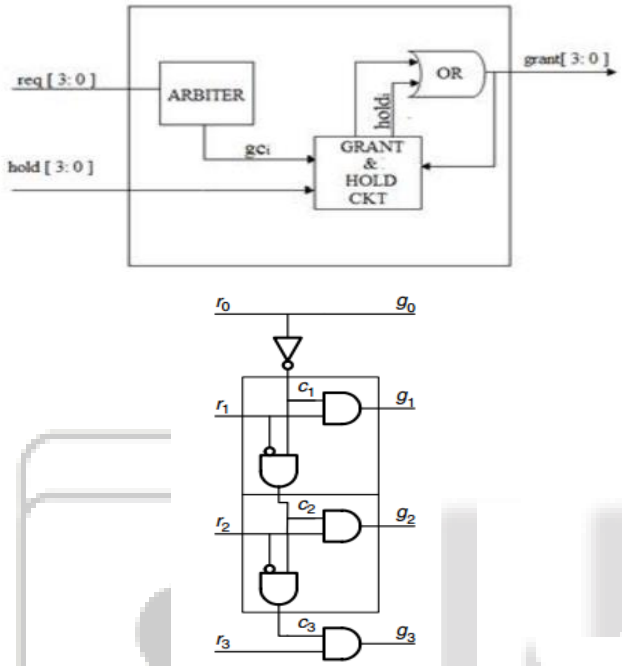

Fig. 6: A 4-bit fixed priority arbiter [6]

E. Fixed Priority Arbiter

Fixed priority arbiter is simplest arbiter. As shown in figure 5 signals r_i, c_i, g_i and c_{i+1} (where i= 0 to 3) indicate request input, carry input, grant output and carry output respectively.

The carry input c_i indicates that the resource has not been granted to a higher priority request and, hence, is available for this bit cell. If the current request is true and the carry is true, the grant line is asserted and the carry output is deasserted, signaling that the resource has been granted and is no longer available. This technique is not useful in practice because it is completely unfair [6].

IV. Results

Any arbitration technique has major issue of fairness which means any requestor should be less suffer from starvation. Fixed Priority arbiter has weak fairness and not used in practice. Lottery Bus algorithm uses random generator technique for selection of requestor which has quite weak fairness [7]. In FCFS and Token pass technique requestor will not release the bus untill it get served. In Round Robin technique each requestor is eventually served.

Below table shows design summary of some of the techniques for 4 requests which are implemented in Xilinx ISE.

NO.	Techniques	Parameters			
		T_{min} (ns)	Fmax (MHz)	Filp Flop	IO Buffers
1	Token Pass	0.833	1200.408	1	7
2	FCFS	1.818	550.191	11	6
3	RR	1.522	656.836	3	9

Table 1: Synthesis Results

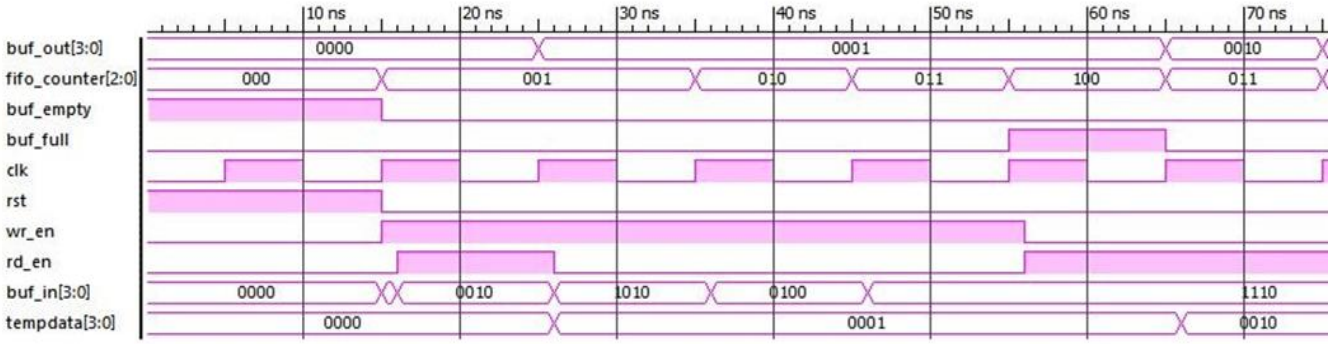

Fig. 7: FCFS for N=4

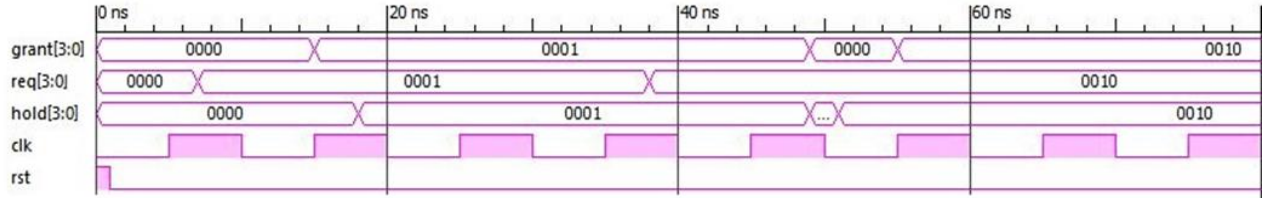

Fig. 8: Token Pass for N=4

V. Conclusion

From the result, it is clear that hardware complexity of FCFS is higher than all the techniques and Token pass technique has least complexity. Average waiting time for process to be excuted is higher in RR compare to FCFS and Token Pass.

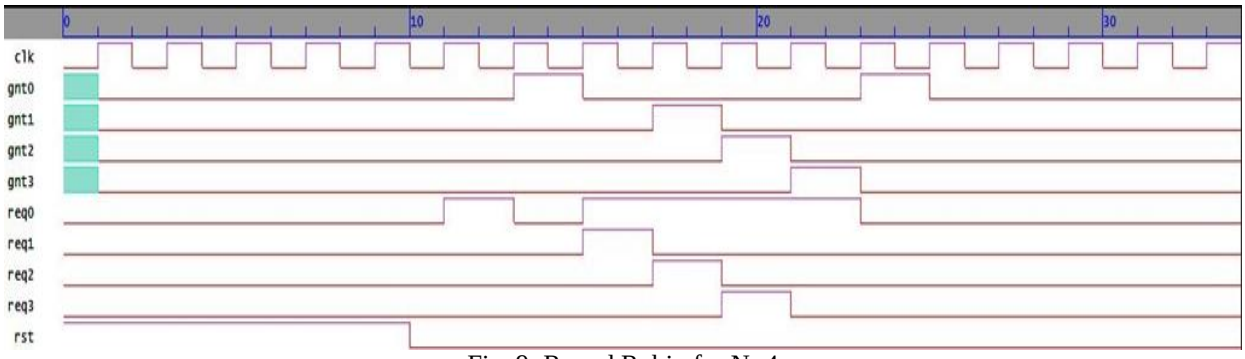

Fig. 9: Round Robin for N=4

REFERENCES

[1] Xufan Wu, Jun Yang, Longxing Shi, "Bus Buffer Evaluation of Different Arbitration Algorithms",SOC Conference, 2005. Proceedings. IEEE International
[2] E. S. Shin, V. J. Mooney and G. F. Riley, "Round-robin Arbiter Design and Generation," 15th International Symposium on System Synthesis, pp. 243 – 248, Oct. 2002.
[3] K. Lahiri, A. Raghunathan, and G. Lakshiminarayan, "LOTTERY BUS: New HighPerformance Communication Architecture for System-on-Chip Designs" ,Design Automation Conference, pp. 15-20, 2001
[4] Matt Weber, 'Arbiter: Design Idea and Coding style', Silicon Logic Engineering, Inc,SNUG boston,200
[5] T.K.Gosh, Computer Organization & Architecture, 3rd edition, McGraw Hill Education, March 2011,
[6] William J. Dally, Brian Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann Publishers 500 Sansome Street, Suite 400, San Francisco,2004
[7] Rachid Benlamri(Ed.), Network Digital Technologies(Part I), Springer, 4th International Conference, NDT 2012, Dubai, UAE, April 2012, p.332
[8] Khanam R., Sharma H., Gaur S., "Design a low latency arbiter for on chip communication architecture", International Conference on Computing, Communication & Automation (ICCCA),Noida, pp.1421-1426, May 2015
[9] Gupta J., Goel N., "Efficient bus arbitration protocol for SoC design", International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM),Chennai, pp.396-400,2015
[10] https://rtlery.com/articles/how-design-round-robin-arbiter.
[11] http://www.os-concepts.thiyagaraaj.com/scheduling-algorithms/first-come-first-served--fcfs--scheduling.

