International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 01 | Jan 2020 www.irjet.net p-ISSN: 2395-0072

Design of a Round Robin Bus Arbiter using System Verilog

Ansuman Mishra

M.Tech, ECE Dept., Manipal Institute of Technology, Manipal, India

---

Abstract – *In SoC's (System on Chip), the number of buses and their interconnection with different subsystems and functional blocks for communication have risen exponentially in the present generation designs. Bus arbiters are highly efficient in handling master bus requests and slave bus responses. The bus arbiter is usually coded according to an algorithm which dictates its operation and efficiency. A Round robin arbiter scheme is based upon the concept of fixed slot per requestor. The most common way of behavioral modelling a round robin arbiter is by using "nested-case" statement in Verilog. However, this coding style is not at all efficient and is lengthy, when the number of requestors are high. This paper describes how to code the arbiter efficiently and in a compact manner using System Verilog language.*

Key Words: Round Robin Arbiter, Bus arbitration, System Verilog.

1.INTRODUCTION

Modern generation SoC (System on Chip) design increases the complexity of on-chip bus-based communication architecture. Communication architecture generally deals with interaction of control and data signals amongst the various functional blocks[1]. Bus based architectures are usually preferred in SoC designs as they are power efficient and provide the framework for complex interconnections. An arbiter is a crucial component in shared bus architecture[2].

Most of the arbiters are modelled based on an algorithm which governs its overall operation and performance. Some of the most commonly used algorithms are:

1. Fixed Priority Algorithm.

2. Round Robin Algorithm.

The Round Robin Algorithm allows each requestor to be served and once served, the requestor is moved to the end of the queue. i.e. lowest priority is assigned to it the maximum wait period time interval a requestor has to wait before the re-service of its request is dependent on the total number of requestors in the queue[3].

Round Robin Bus Arbiter is widely used for shared bus arbitration, queueing & work load balancing[4].

1.1 Round Robin Algorithm

Round Robin is a CPU scheduling algorithm. A fixed time slot is assigned to each process in the queue. It is quite simple in implementation and avoids process starvation. The short coming of Fixed Priority Algorithm is that the time interval before the grant is provided to the lowest priority process has no limit [5]. In Round Robin Algorithm, every process request is allowed a grant in a cyclic order. Here, a pointer is designated, and it shifts to next process request. Hence, the maximum wait period before a request is re-serviced again is dependent on the number of requestors in the queue.

2. DESIGN METHODOLOGY

The Round Robin Arbiter can be modelled in system Verilog using nested case statements. However, this style of coding being simple, and primitive is highly inefficient when the number of requestors is more. The style of coding also affects the synthesis output. The proposed way of coding involves the request process which is pointed by a pointer is assigned the top most priority and remaining all other requests are rotated behind it. This rotated request vector is then sent through a simple priority arbiter. The grant vector from the priority arbiter is then "unrotated" to come up with the Round-Robin Arbiter's final grant signal. The Fig.1 depicts the operation.

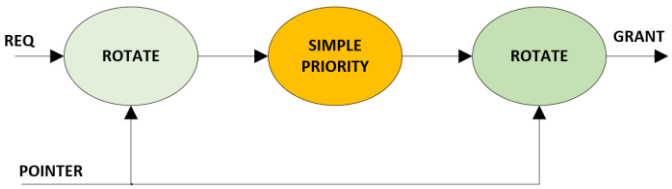

Fig -1: Operation methodology for Round Robin Arbiter.

For this paper, a 4-bus round robin arbiter is considered. The block diagram of the Round Robin Arbiter modelled using System Verilog HDL is shown in Fig 2.

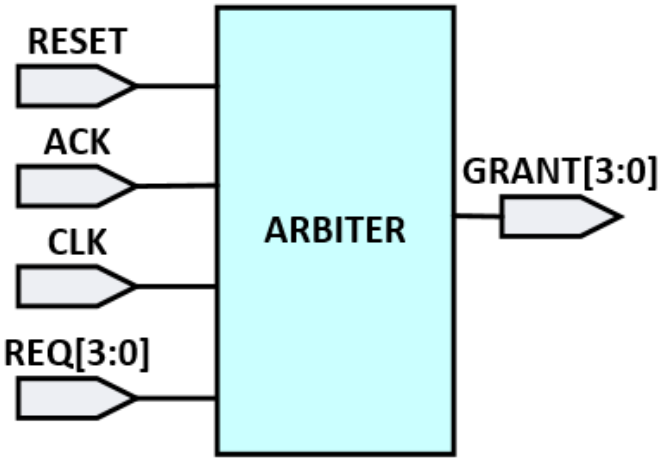

Fig -2: Top Level block diagram for Round Robin Arbiter.

The case statement used for Round Robin arbitration is as follows:

```
case (rotate_ptr[1:0])
    2'b00: shift_request[3:0] = request[3:0];
    2'b01: shift_request[3:0] = {request[0],request[3:1]};
    2'b10: shift_request[3:0] = {request[1:0],request[3:2]};
    2'b11: shift_request[3:0] = {request[2:0],request[3]};
endcase
```

The Round Robin Arbitration request and grant methodology is depicted in the Fig-3.1-3.5

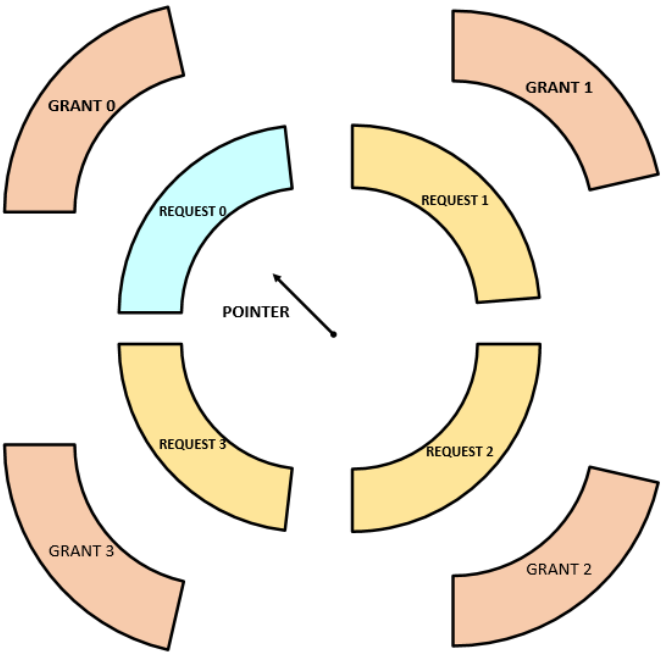

Fig-3.1: Request 0 is asserted

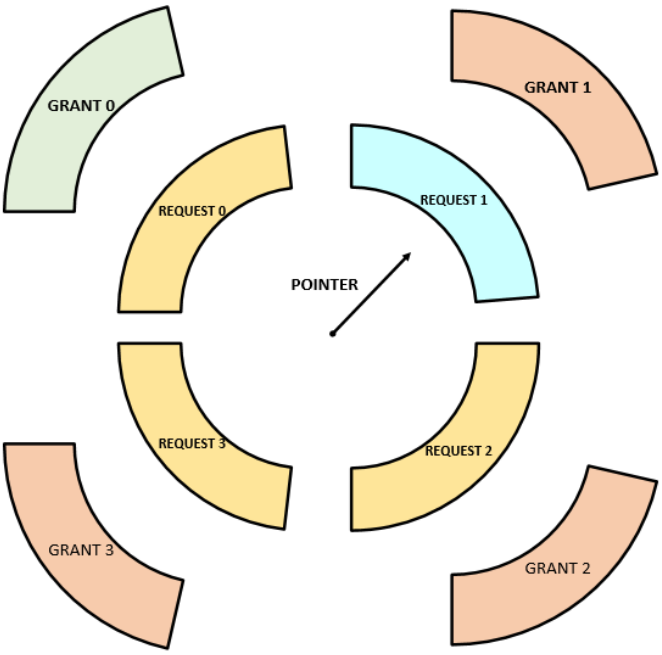

Fig-3.2: Request 1 is asserted, Grant 0 asserted.

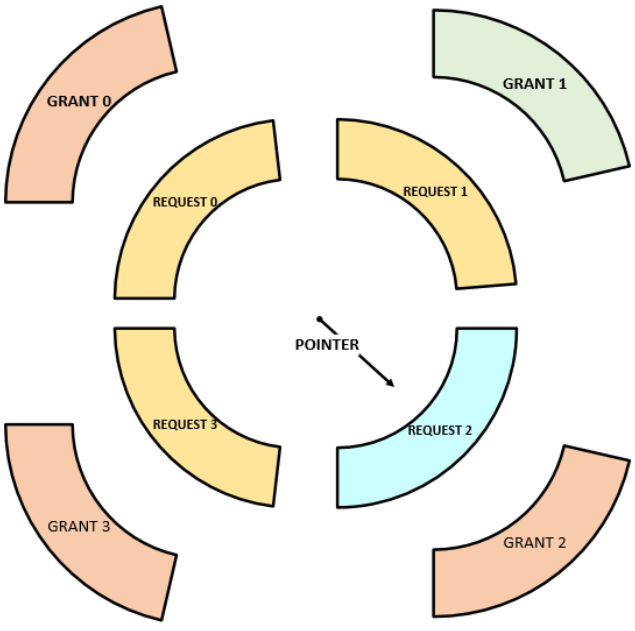

Fig-3.3: Request 2 is asserted, Grant 1 asserted.

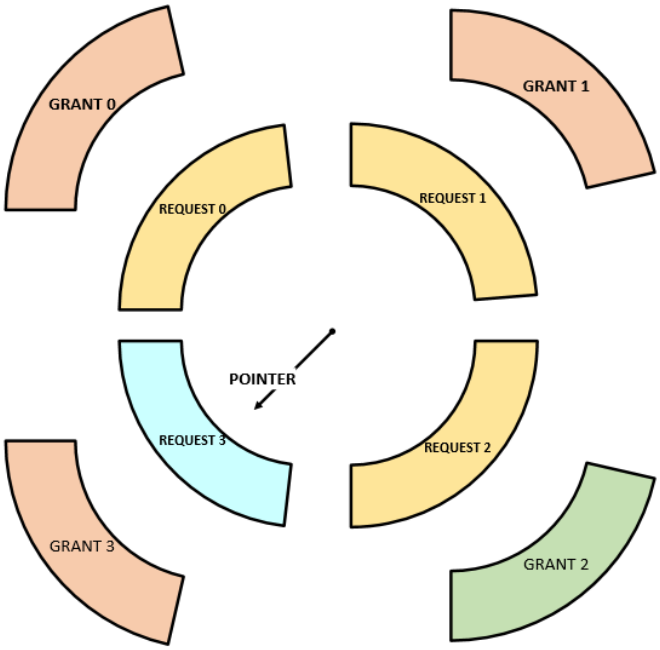

Fig-3.4: Request 3 is asserted, Grant 1 asserted.

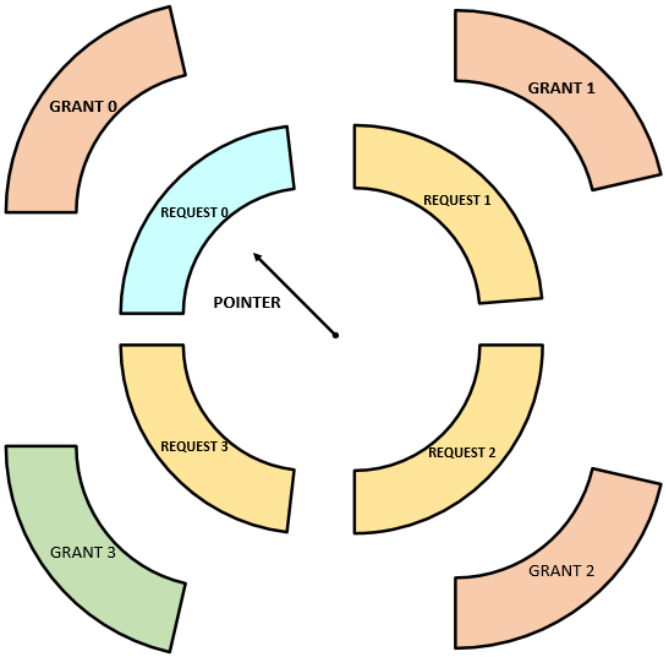

Fig-3.5: Request 0 is asserted, Grant 3 asserted.

3. RESULTS

The main design and testbench are written using System Verilog HDL and compiled using Synopsys Verdi. The waveform viewer is used to display the output. Fig-4 shows the output case when requests are synchronized. Fig-5 shows the case when request 0 is set first priority and it is kept high for rest of the cycles.

Fig-4: Output when the requests are synchronized.

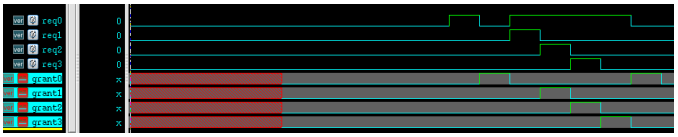

Fig-5: Output when the request 0 is set high priority and again set high for rest of the cycles

4. CONCLUSIONS

The Round Robin Arbiter is modelled correctly using System Verilog HDL and compiled. The operation is found to be correct as in Fig-5 where request 0 once serviced is assigned lowest priority and it is re-serviced again at the end of grant 3. The use of vector arrays and concatenation operator significantly reduces the code density and helps in better RTL synthesis.

REFERENCES

[1] Z. Fu and X. Ling, "The design and implementation of arbiters for Network-on-chips," in 2010 2nd International Conference on Industrial and Information Systems, vol. 1, July 2010, pp. 292–295.

[2] E. S. Shin, V. J. Mooney, and G. F. Riley, "Round-robin Arbiter Design and Generation," in 15th International Symposium on System Synthesis, 2002., Oct 2002, pp. 243–248.

[3] M. Oveis-Gharan and G. N. Khan, "Index-Based Round-Robin Arbiter for NoC Routers," in 2015 IEEE Computer Society Annual Symposium on VLSI, July 2015, pp. 62–67.

[4] M. Oveis-Gharan and G. N. Khan, "Index-Based Round-Robin Arbiter for NoC Routers," in 2015 IEEE Computer Society Annual Symposium on VLSI, July 2015, pp. 62–67.

[5] S. Q. Zheng and M. Yang, "Algorithm-Hardware Codesign of Fast Parallel Round Robin Arbiters," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 1, pp. 84–95, Jan 2007.