ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Index Based Round Robin Arbiter for NOC Routers

Zakkam Swetha Unmila
PG Scholar,
Sri Krishnadevaraya Engineering College,
Gooty, AP, India.

M Rajakiran
Assistant Professor,
Sri Krishnadevaraya Engineering College,
Gooty, AP, India.

K Geetha
Associate Professor,
Sri Krishnadevaraya Engineering College,
Gooty, AP, India.

Dr.R.Ramachandra
Professor,
Sri Krishnadevaraya Engineering College,
Gooty, AP, India.

Abstract:
In the NoC router micro-architecture design, arbiter has become increasingly important due to its significant impact on the performance and efficiency of NoC systems. In this paper, we propose an Index-based Round Robin (IRR) arbiter that functions on the index format of input ports of the router. The Round Robin Arbiter (RRA), a crucial building block for high-speed switches/routers, receives a new attention with the advent of the NoC. The micro architecture of IRR arbiter scales logarithmically with the number of input ports as compared to a conventional round robin arbiter that scales with its input ports. The behavior and architecture of our arbiter leads to lower power consumption and chip area as well as higher performance characteristics.

I. INTRODUCTION

In digital system design, arbiters are used to allocate and access shared resources. Whenever a resource, such as a buffer, channel or a switch-port is shared, an arbiter is required to assign the access to the resource at a particular time. The most common usage of arbiters is the shared-bus arbitration of a bus-based system where multiple master modules can initiate their transactions. The modules must be arbitrated for access to the bus before initiating a transaction. In this paper, we investigate the arbiters used in NoC systems.

An NoC system facilitates communication among IP cores of an SoC. It includes a network of switches (routers) that are interconnected by communication links as illustrated in Figure 1.a. Figure 1.b shows a typical NoC router that consists of some input and output ports, an arbiter and a crossbar switch.

The data path of a router is made of port buffers, crossbar switch and interconnection structure, while the control unit of a router is mainly consists of arbiters. The structure of arbiter becomes even more complex with the utilization of Virtual Channel (VC) mechanism. Figure 2 shows two 4-input arbiter that can arbitrate the use of resources. The arbiter accepts n requests (r0, r1,…r_{n-1}), arbitrates among the asserted request lines, and selects an ri for service, and then asserts the corresponding grant line, gi. For example, assume the arbitration for the output port of a crossbar switch among a set of requests from the VCs of some input ports. The input-port VCs that have flits will issue request signals for having access to one of the desired output-port. Assume, there are 5 VCs and VCs 0, 2, and 4 assert their request lines, r0, r2, and r4 respectively. The arbiter will then arbitrate and select one of these VCs for assigning the desired output-port. Assume the grant of VC2 (i.e. g2) is asserted. VCs 0 and 4 lose the arbitration and must hold their requests active until they receive the grant signal for their output-ports.

Arbiters can be categorized in terms of fairness (weak, strong or FIFO) arbiters. In a weak fairness arbiter, every request is eventually granted. The requests of a strong fairness arbiter will be granted equally often. The requests of FIFO fairness are granted in a first come first served basis. Moreover, arbiters in terms of priority can be grouped in twofixed and variable architectures. For a fixed priority arbiter, the priority of requests is established in a linear order. Figure 3a illustrates a 4-input fixed-priority arbiter where r0 has the highest and r3 has the least priority. The architecture can be expanded to n-input arbiter where for each middle request, there is an arbiter cell

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

consisting of two ANDs and an Inverter. The first and last arbiter cells can be simplified. For each request input ri, there is a carry input ci, a grant output gi, and a carry output, ci+1 where i {0,1, .. n-1}. Therefore, a low level ci indicates that at leastone of requests from r0 to ri-1 was has been asserted. More over, in case that the request ri and carry ci are high, the grant, gi is set, and all the following grants i.e. gi+1 to gn-1will become reset. It is obvious that the critical path of thecircuit is from the first request, r0 to the last grant, gn-1 due to propagation of carry from head to the tail of the arbiter.Fixed priority arbiters provide weak fairness arbitration because when a request is continuously asserted, none of itsfollowing requests will ever be served.

In order to have a fair iterative arbiter, we can use avariable priority arbiter as illustrated in Figure 2b. An OR gate and a priority input signal, pi is added to each cell of the fixed priority arbiter shown in Figure. When p1 is set, its corresponding request, r1 has high priority and the priority decreases from that point cyclically around the circular carry chain. Now we can create a fair iterative arbiter by changing its priority from cycle to cycle. In an n-input arbiter, if the grant, gi (where i ={0, 1, ... n-1}) is connected to the next priority vector pi+1, a Round Robin (RoR) arbiter is created.Figure 3 illustrates a 4-input RoR arbiter. If a grant, g1 becomes high at the current cycle, it causes p1 to be set highon the next clock cycle. This leads the request, r2 to becomethe highest priority at the next cycle, where the request, r1becomes the lowest priority. For the sake of simplicity, we assume that the arbitration cycle takes one clock cycle in allthe architectures describe in this paper.

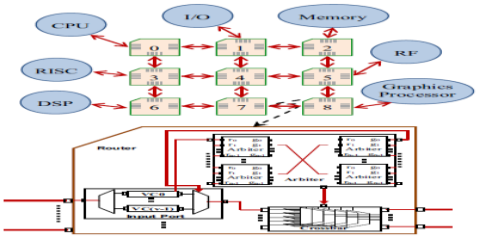

Figure . (a) 2D SoC mesh. (b) Wormhole NoC router.

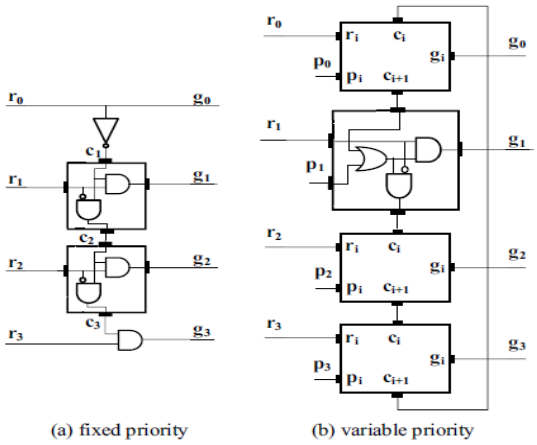

Figure .2 4-input arbiter architectures.

The functionality of a round-robin arbiter can be explained as a request that is just granted will have the lowest priority on the next arbitration cycle [1]. The round robin arbiters are simple, easy to implement, and starvation free. When the input requests are large in numbers, the structure of round robin arbiter grows that leads to large chip area, higher power consumption, and critical path delay. In an NoC design, the critical path delay of arbiter usually dominates among the critical path delays of input-port and crossbar switch due to the architectural complexity of arbiter as compared to those of port and crossbar switch. Therefore, the arbiter circuit determines the maximum frequency (or the speed), Fmax of an NoC router. The critical impact of arbiter on the performance of the NoC system and the characteristic behaviour of round robin architectures have created a lot of interest of NoC researchers.

II. RELATED WORK

The architecture of a popular Matrix round robin arbiteris presented by Dally and Towles. A 4-input Matrix arbiter architecture is shown in Figure. It implements aleast recently served priority scheme where a request, ri winsan arbitration. It resets the bits of row i and sets the bits of column i to make itself the lowest priority where I ϵ {0,..3}.The Matrix arbiter is claimed to be useful for small number of inputs as it is fast, economical, and performs strong fairness

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

arbitration. However, no evaluation is presented. Fuand Ling evaluated and compared the RoR and Matrix arbiters in terms of resource, performance and power consumption for an FPGA platform. They concluded that the Matrix arbiter consumes more resource, same power but can process data more quickly than the RoR arbiter.

Zheng and Yang proposed a Parallel Round Robin Arbiter (PRRA) based on a simple binary search algorithm as illustrated for a 4-input PRRA in Figure. They further proposed an Improved PRRA (IPRRA) design where the output signals, gL and gR of PRRA are disconnected and directly AND ed with grant signals to generate new grant signals as shown in Figure. The IPRRA reduces the timing of PRRA significantly. A High speed and Decentralized Round robin Arbiter (HDRA) has been presented by Lee etal., which is illustrated in Figure. Each circuit enclosed with dash circle represents a filter circuit whose maincomponents are a D flip-flop and a multiplexer.

The filter circuit filters out the input without request or the one with request that has already been granted at that arbitration cycle. The un-filtered inputs with their requests participate inthe arbitration again next cycle by setting its corresponding D-type flip-flops to 0 that are done by enabling the ack signals from higher lower level. The HDRA arbiter will reset itself asynchronously by the input self_rst from the root. Thesys_rst indicates the system reset signal and is used initially before each arbitration cycle for all requests. A 4-input HDRA arbiter has a simpler circuit than a higher input HDRA architecture because the act, rnext and self-rst are connected together.

III.INDEX-BASED ROUND ROBIN ARBITER

In this paper, we present a new arbiter design called Index-based Round Robin (IRR) arbiter that employs a least recently served priority scheme and achieve strong fairness arbitration. The proposed arbiter has smaller arbitration delay, lower chip area and it also consumes less power as compared to theafore mentioned arbiters. Before describing the IRR arbiter architecture, we introduce an in separable and critical output in the arbiter design.

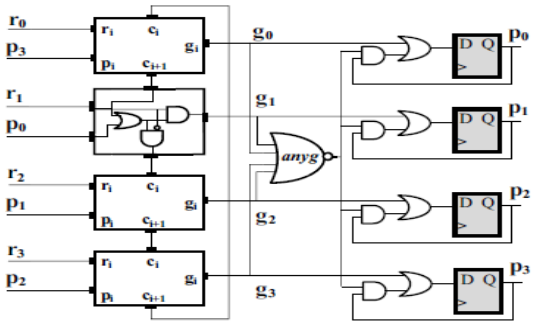

Figure: 3 4-input RoR arbiter architecture.

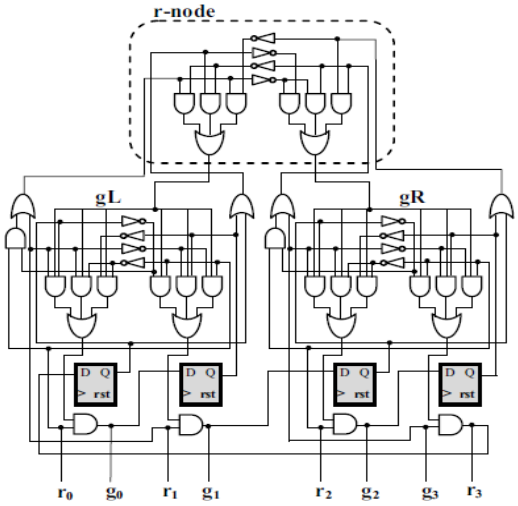

Figure:4 4-input PRRA architecture.

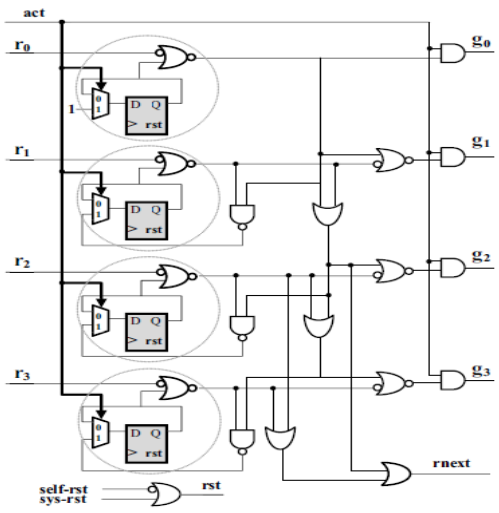

Figure: 5-input HDRA architecture.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A. Grant Index

All the arbiters have an output array, grant whose widthis the same as that of input width. However, in practical designs, the index of grant signals, g_id is also generated that is used to address the granted request in some other components, such as control tables, multiplexers and memories used in NoC routers. When a router crossbar is made of multiplexers, the g_id can be connected to selection port of multiplexer to switch the granted input tothe requested output port. The width ofg_id is the log2 of the width of grant. We used the g_id asthe first output of our proposed design and due to lower width of g_id, our arbiter design is smaller and faster as compared to other arbiters. Due to the critical use of g_id in NoC design, we consider all the arbiters covered in this paper to generate both grant and g_id as outputs.

B. Fixed Priority Arbiter

Our fixed priority arbiter is simpler and economical as illustrated in Figure. The priority of requests is linear andin the ascending order where r0 has the highest priority. The index of first asserted request is switched to the output asthe index of grant, g_id. Then the g_id is decoded to create the grant signals. The last request, rn-1 has a simplified circuit where instead of being multiplexed like other requests, it is ANDed by gn-1. Therefore, in case that only rn-1 is high, then only gn-1 becomes high.

C. Variable Priority Arbiter

If the g_id output of the fixed priority arbiter of Figure 6is connected to the last multiplexer, each request behaves as it has the highest priority through ascending order of the loop. For example, for four requests (r0, r1, r2 and r3) the output of multiplexer, M1 generates an index where r1 has the highest priority then r2, r3, and r0. Therefore, by further multiplexing of these outputs, we can choose an input as the highest priority request as shown in Figure 7. For example,when P=1, the output of multiplexer M1 is selected and the request, r1 has the highest priority. In the case of no request asserted, or r0 is asserted, the g_id issue the same value (i.e.zero). In order to separate these two conditions, ORing of requests, any_r is ANDed with the g0 so that when all the requests are zero, all the grants will become zero.

D. IRR Arbiter

If the next index of granted request is chosen for thenext priority selection, the current granted request receives the least priority, and its next request receives the highest priority among all the requests. To accomplish it, the g_id array is stored in a register, and the output of the register is incremented and connected to the selection port of multiplexer, MP as shown in Figure8.

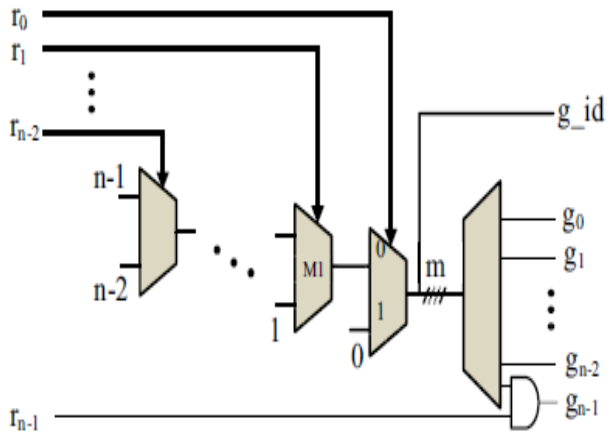

Figure.6: n-input fixed priority arbiter, where m =log2 (n)

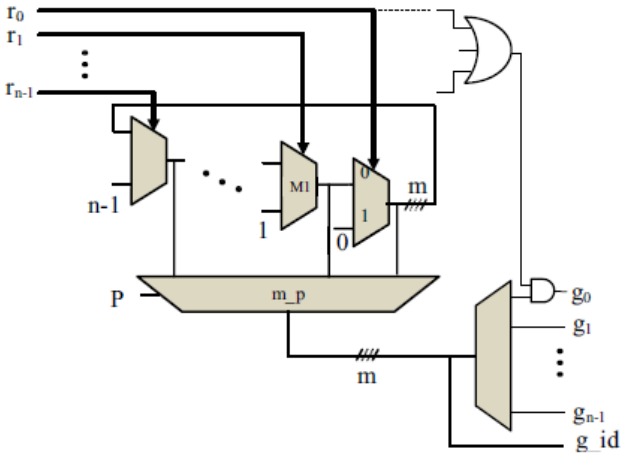

Figure.7: n-input variable priority arbiter.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

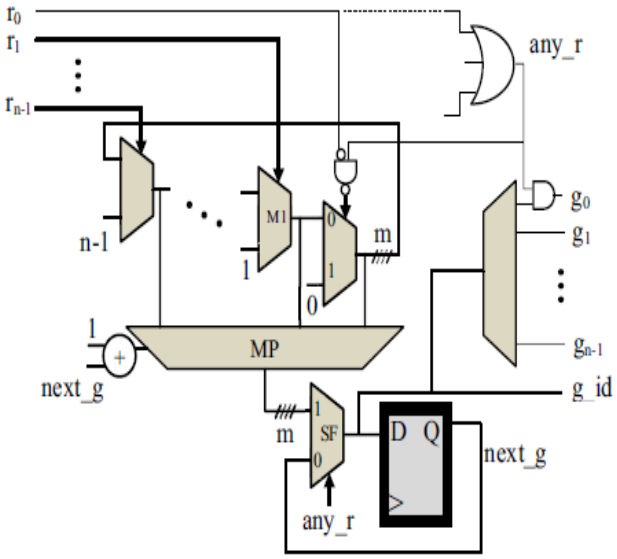

Figure.8: n-input IRR arbiter, where m=log2 (n).

IV.SIMULATION RESULTS

We have coded the all Round Robin techniques in Verilog HDL. All the designs are synthesized in the Xilinx Synthesis Tool and Simulated using Xilinx ISE simulator. The synthesis and simulation results are as shown below figures.

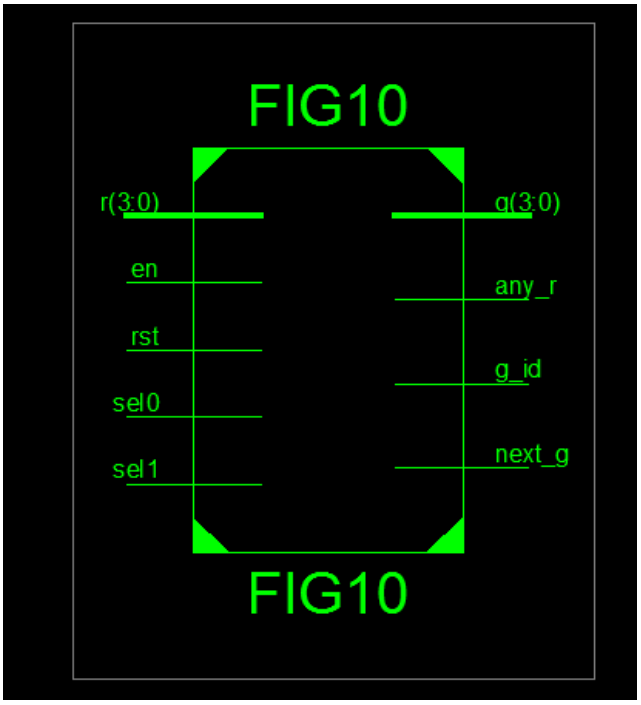

Fig8: Block diagram of an Index-based Round Robin (IRR) arbiter

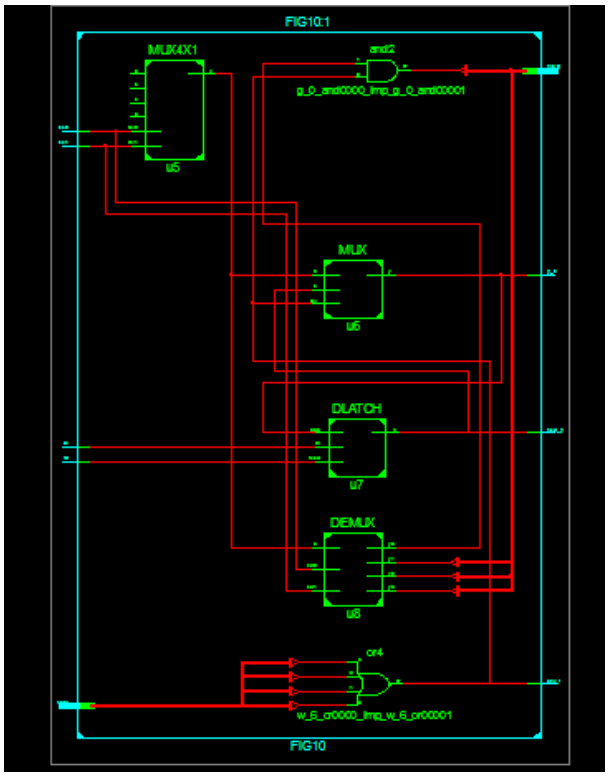

Fig9: RTL Schematic of an Index-based Round Robin (IRR) arbiter

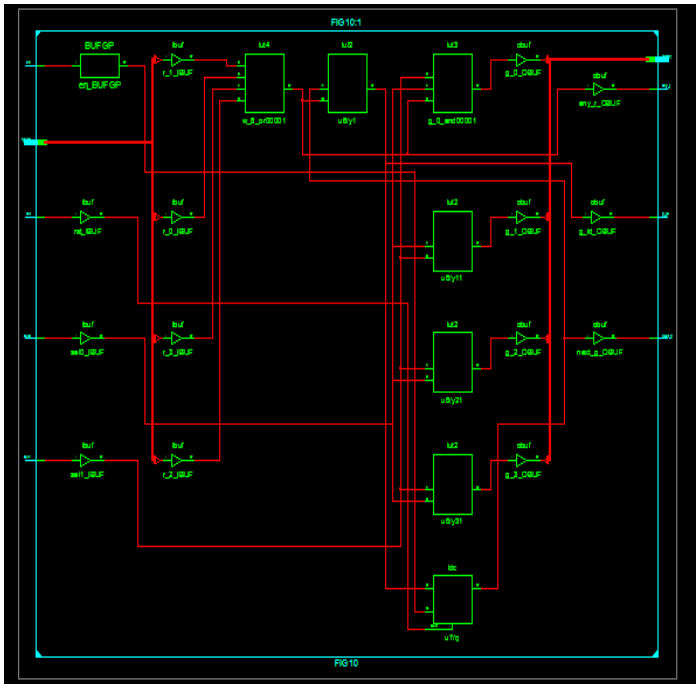

Fig9: Technology Schematic of an Index-based Round Robin (IRR) arbiter.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

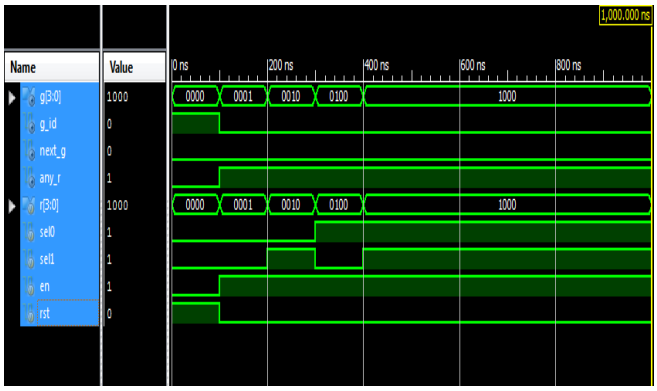

Fig10: Simulation output waveform of an Index-based Round Robin (IRR) arbiter.

CONCLUSION:

We have presented a strong fairness round robin arbiter design, IRR. We proved that our design achieve a strong fairness arbitration for all input patterns, which is not guaranteed by some other previous designs such as HDRA [3], PRRA and IPRRA [4]. The key difference of our approach is its operation on the basis of index format of the input ports. This index based arbitration is simple, fast and with small hardware over head. The distinctive feature of our IRR arbiter design is its lower power consumption as compared to other arbiters due to its usage of fewer registers.

REFERENCES

[1] W. J. Dally and B. Towles. “Arbitration,” In: Principles andPractices of Interconnection Networks. Morgan KaufmannPublishers, 2004, pp. 349-362.

[2] Z. Fu and Xiang Ling, "The design and implementation ofarbiters for Network-on-chips," Proc. 2nd Int. Conf. Industrialand Information Systems, Dalian, 2010 pp. 292-295.

[3] S. Q. Zheng and Mei Yang, "Algorithm-Hardware Codesign ofFast Parallel Round-Robin Arbiters", IEEE Trans Parallel andDistributed Systems, vol. 18, January 2007, pp. 84-95

[4] Yun-Lung Lee, Jer Min Jou, and Yen-Yu Chen, "A High-Speed and Decentralized Arbiter Design for NoC," Proc.IEEE/ACS Int. Conf. on Computer Systems and Applications,Rabat, 2009 pp. 350-353.