

An Efficient Design of Serial Communication Module UART Using Verilog HDL

Pogaku Indira

**M.Tech in VLSI and
Embedded Systems,
Siddhartha Institute of
Engineering and Technology.**

Dr.D.Subba Rao, M.Tech, Ph.D

**HOD,
Department of ECE,
Siddhartha Institute of
Engineering and Technology.**

Ashok Babu Amudapakula

**Assistant Professor,
Department of ECE,
Siddhartha Institute of
Engineering and Technology.**

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 microarchitecture 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. AnNoC 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 (r_0, r_1, \dots, r_{n-1}), arbitrates among the asserted request lines, and selects an r_i for service, and then asserts the corresponding grant line, g_i . 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, r_0, r_2 , and r_4 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. g_2) 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 two fixed 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 r_0 has the highest and r_3 has the least priority.

The architecture can be expanded to n-input arbiter where for each middle request, there is an arbiter cell consisting of two ANDs and an Inverter. The first and last arbiter cells can be simplified. For each request input r_i , there is a carry input c_i , a grant output g_i , and a carry output, c_{i+1} where $i \in \{0, 1, \dots, n-1\}$. Therefore, a low level c_i indicates that at least one of requests from r_0 to r_{i-1} has been asserted. Moreover, in case that the request r_i and carry c_i are high, the grant, g_i is set, and all the following grants i.e. g_{i+1} to g_{n-1} will become reset. It is obvious that the critical path of the circuit is from the first request, r_0 to the last grant, g_{n-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 its following requests will ever be served.

In order to have a fair iterative arbiter, we can use a variable priority arbiter as illustrated in Figure 2b. An OR gate and a priority input signal, p_i is added to each cell of the fixed priority arbiter shown in Figure. When p_1 is set, its corresponding request, r_1 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, g_i (where $i \in \{0, 1, \dots, n-1\}$) is connected to the next priority vector p_{i+1} , a Round Robin (RoR) arbiter is created. Figure 3 illustrates a 4-input RoR arbiter. If a grant, g_1 becomes high at the current cycle, it causes p_1 to be set high on the next clock cycle. This leads the request, r_2 to become the highest priority at the next cycle, where the request, r_1 becomes the lowest priority. For the sake of simplicity, we assume that the arbitration cycle takes one clock cycle in all the architectures describe in this paper.

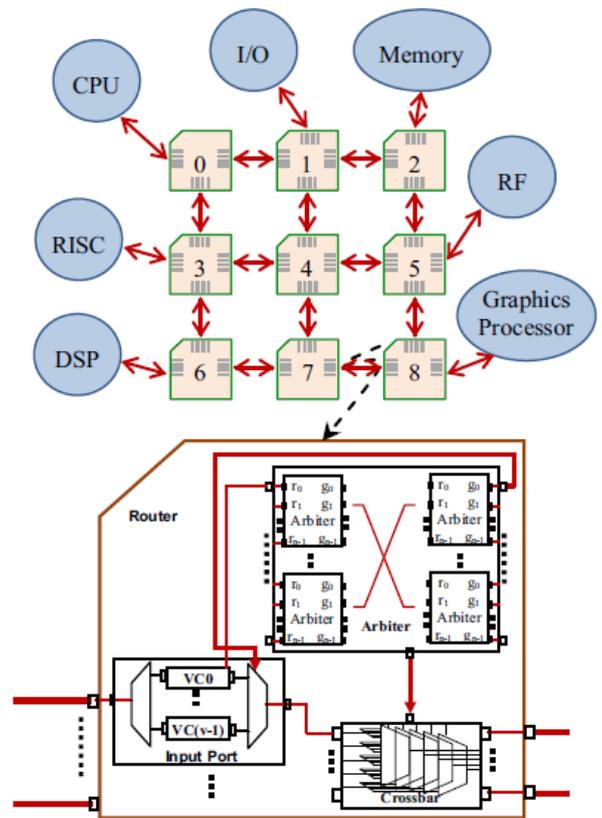


Figure. (a) 2D SoC mesh. (b) Wormhole NoC router.

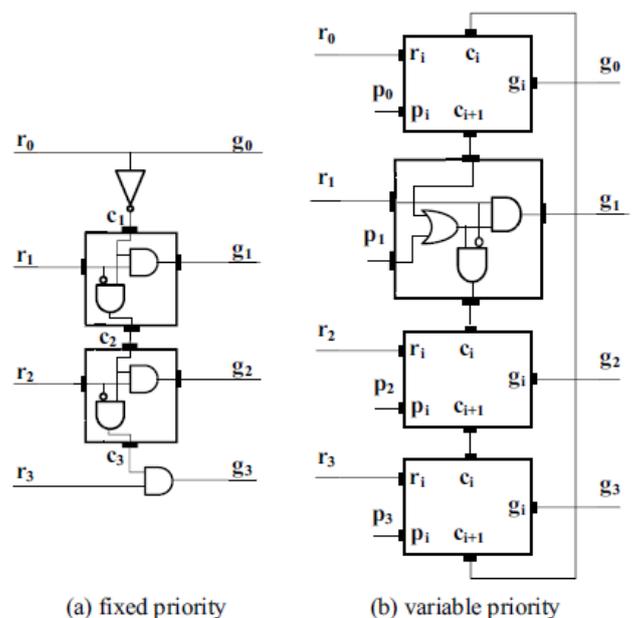


Figure .2 4-input arbiter architectures.

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 arbiters compared to those of port and crossbar switch. Therefore, the arbiter circuit determines the maximum frequency (or the speed), F_{max} 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 arbiters presented by Dally and Towles. A 4-input Matrix arbiter architecture is shown in Figure. It implements least recently served priority scheme where a request, r_i wins an arbitration. It resets the bits of row i and sets the bits of column i to make itself the lowest priority where $i \in \{0,..3\}$. The Matrix arbiter is claimed to be useful for small number of inputs as it is fast, economical, and performs strong fairness arbitration. However, no evaluation is presented. Fu and 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, g_L and g_R of PRRA are disconnected and directly ANDed 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 et al., which is illustrated in Figure. Each circuit enclosed with dash circle represents a filter circuit whose main components 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 in the arbitration again next cycle by setting its corresponding D-type flip-flops to 0 that are done by enabling the a ck signals from higher lower level. The HDRA arbiter will reset itself asynchronously by the input $self_rst$ from the root. The sys_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 , $rmxst$ 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 the aforementioned arbiters. Before describing the IRR arbiter architecture, we introduce an inseparable and critical output in the arbiter design.

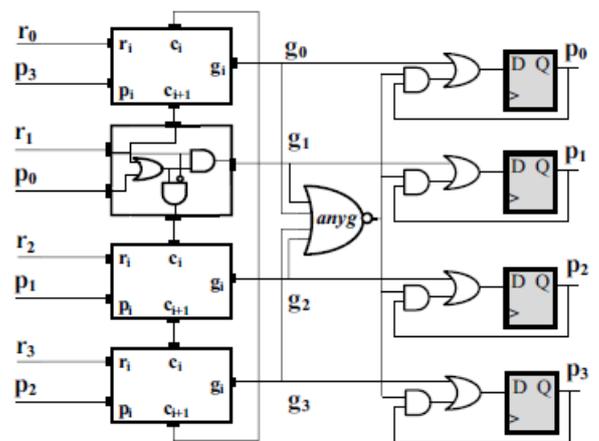


Figure: 3 4-input RoR arbiter architecture.

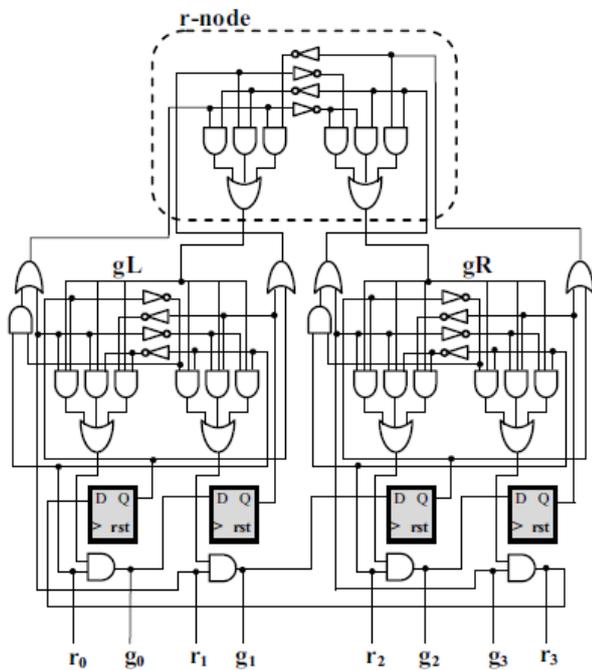


Figure:4 4-input PRRA architecture.

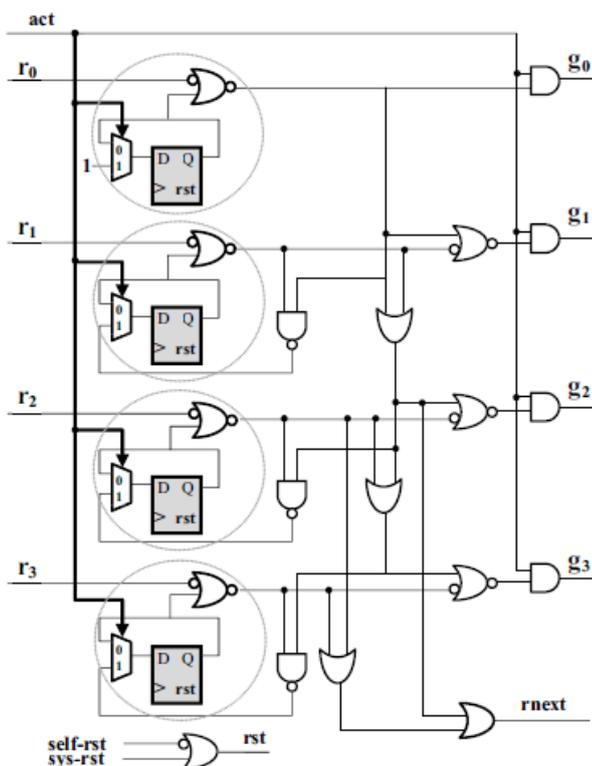


Figure: 5-input HDRA architecture.

A. Grant Index:

All the arbiters have an output array, grant whose width is 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 to the requested output port. The width of g_id is the \log_2 of the width of grant. We used the g_id as the 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 and in the ascending order where r_0 has the highest priority. The index of first asserted request is switched to the output as the index of grant, g_id . Then the g_id is decoded to create the grant signals. The last request, r_{n-1} has a simplified circuit where instead of being multiplexed like other requests, it is ANDed by g_{n-1} . Therefore, in case that only r_{n-1} is high, then only g_{n-1} becomes high.

C. Variable Priority Arbiter:

If the g_id output of the fixed priority arbiter of Figure 6 is 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 (r_0, r_1, r_2 and r_3) the output of multiplexer, M1 generates an index where r_1 has the highest priority then r_2, r_3 , and r_0 . 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, r_1 has the highest priority. In the case of no request asserted, or r_0 is asserted, the g_id issue the same value (i.e.zero). In order to separate these two

conditions, O Ring 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 the next 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_idarray 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.

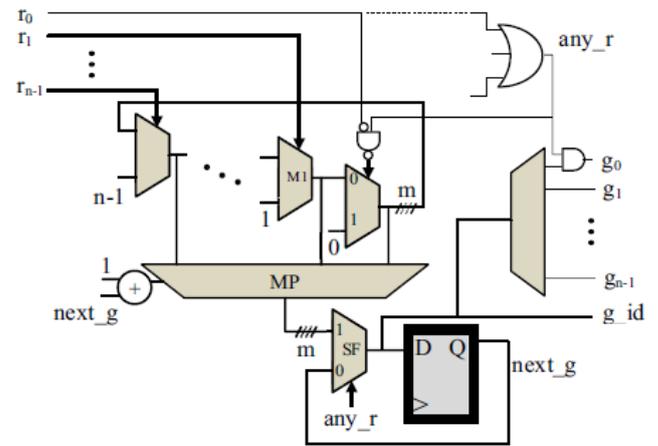


Figure.8: n-input IRR arbiter, where $m = \log_2(n)$.

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.

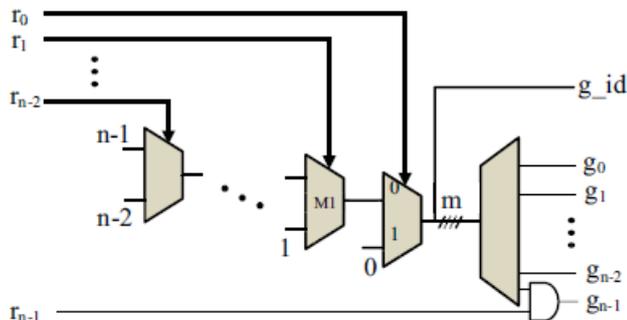


Figure.6: n-input fixed priority arbiter, where $m = \log_2(n)$

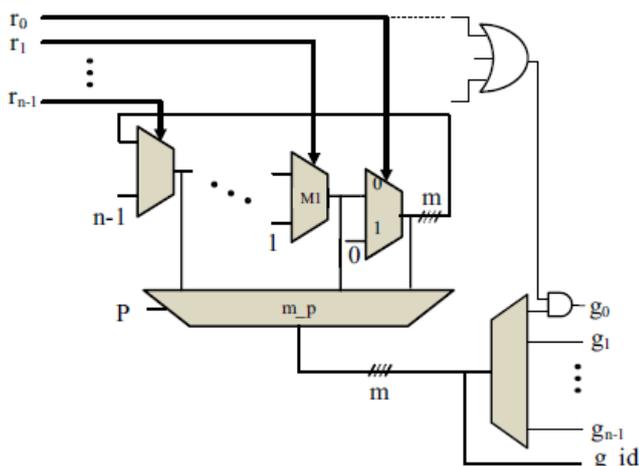


Figure.7: n-input variable priority arbiter.

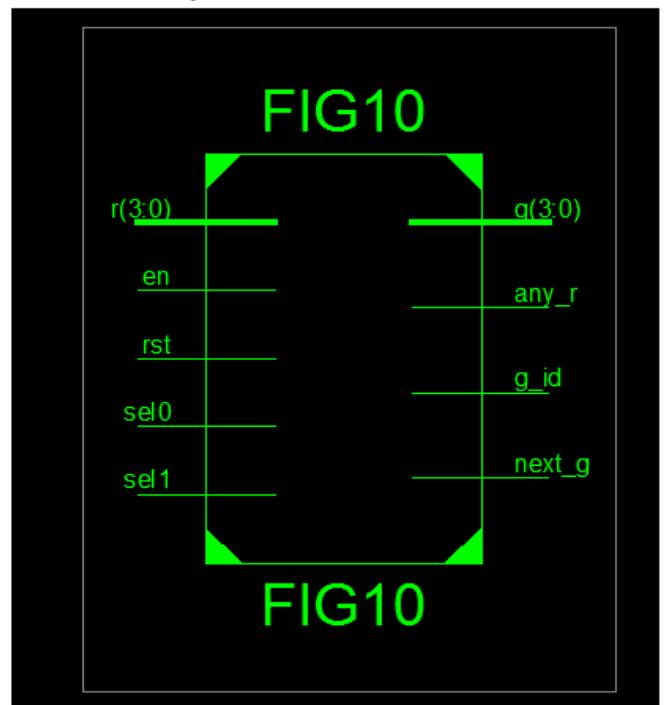


Fig8: Block diagram of an Index-based Round Robin (IRR) arbiter

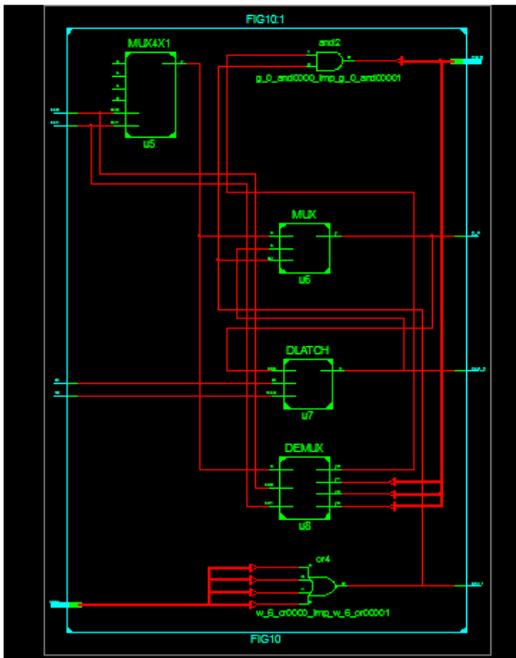


Fig9: RTL Schematic of an Index-based Round Robin (IRR) arbiter

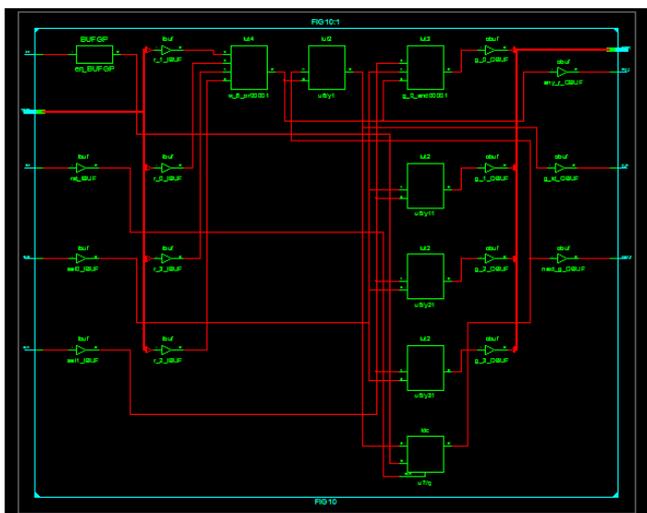


Fig9: Technology Schematic of an Index-based Round Robin (IRR) arbiter.



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 IPRA [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 overhead. 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 and Practices of Interconnection Networks. Morgan Kaufmann Publishers, 2004, pp. 349-362.

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

[3]S. Q. Zheng and Mei Yang, "Algorithm-Hardware Codesign of Fast Parallel Round-Robin Arbiters", IEEE Trans Parallel and Distributed 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.

Author's Details:



Pogaku Indra

She received her B. Tech in ECE from CVSR College of Engineering Hyderabad affiliated to JNTU, Hyderabad ,A.P, India, and pursuing M. Tech in VLSI and Embedded Systems at Siddhartha Institute of

Engineering and Technology, Ibrahimpatnam,
Hyderabad affiliated to JNTU, Hyderabad ,A.P, India.



Dr.D.Subba Rao

Is a proficient Ph.D person in the research area of Image Processing from Vel-Tech University, Chennai along with initial degrees of Bachelor of Technology in Electronics and Communication Engineering (ECE) from Dr. S G I E T, Markapur and Master of Technology in Embedded Systems from SRM University, Chennai. He has 13 years of teaching experience and has published 12 Papers in International Journals, 2 Papers in National Journals and has been noted under 4 International Conferences. He has a fellowship of The Institution of Electronics and Telecommunication Engineers (IETE) along with a Life time membership of Indian Society for Technical Education (ISTE). He is currently bounded as an Associate Professor and is being chaired as Head of the Department for Electronics and Communication Engineering discipline at Siddhartha Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad.



Ashok Babu Amudapakula

He received his B. Tech in ECE from Rao&Naidu Engineering College of Technology, ongole, affiliated to JNTU, Kakinada, A.P, India, and M. Tech in DECS at QIS College of Engineering & Technology ongole, affiliated to JNTU, Kakinada, A.P, India. His area of interest are Digital VLSI Design & VLSI Image Processing. He is presently working as Asst prof He is currently bounded as an Assistant Professor for Electronics and Communication Engineering discipline at Siddhartha Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad.