

Design and Implementation of Router for NOC on FPGA

Gaurav Verma^{#1}, Harsh Agarwal^{*2}, Shreya Singh^{*3}, Shaheem Nighat Khanam^{*4},
Prateek Kumar Gupta^{*5} and Vishal Jain^{#6}

^{1,2,3,4,5}*Department of Electronics & Communication, Jaypee University, A-10,
Sector-62, Noida (U.P.)-India*

⁶*Bharati Vidyapeeth's, Institute of Computer Applications and Management
(BVICAM), New Delhi (INDIA)*

Abstract

In today's technological era, SOC has undergone rapid evolution and is still processing at a swift pace. But due to this explosive evolution of semiconductor industry, the devices are scaling down at a rapid rate and hence, SOC today have become communication-centric. However, the existing bus architectures comprising of wires for global interconnection in SOC design are undergoing design crises as they are not able to keep up with the rate of scaling down of devices. To overcome bottleneck of communication system, NOC is an upcoming archetype. In on-chip network, router is considered as an important component. This paper proposes router, its components and parameters which affects the entire design. Thus, to validate the functioning of NOC on hardware, router has been designed in VHDL and simulated in Xilinx ISE 14.1 targeting Xilinx XC5VLX30-3 FPGA.

Keywords: *Network-on-chip, System-on-chip, VHDL, Field programmable gate Array*

1. Introduction

NOC has become popular in domain of various communication infrastructures. Previously designers preferred various communication systems but as the core count has increased these communication architectures have proved to be bottleneck for future SOC (System-on-chip) as it limits scalability, reusability and complexity. In technological era, VLSI industry found a paradigm called SOC but after seeking major disadvantages in SOC like underutilisation of cores, poor reusability, high complexity, poor scalability, *etc.*, they found a new paradigm called NOC (Network-on-chip) to overcome the disadvantages of SOC. NOC is one of the efficient on-chip communication architecture for SOC. Concept of NOC was introduced by Ahmed Hemani *et al.* NOC functionality is explained in terms of network arrangement and data transportation from source to destination. The design of NOC in VHDL requires a lucid understanding of concepts like switching methodology, routing strategies, topology, control flow, *etc.* The future work involves the study and simulation of 4x4 mesh topology for NOC. Implementation of 4x4 mesh topology block diagram consists of various internal block architectures like Arbiter, Router, FIFO Buffer and Crosssbar. This paper focuses on study and simulation of router with its internal architecture. Block diagram of 4x4 mesh topology for NOC is represented in figure 1.

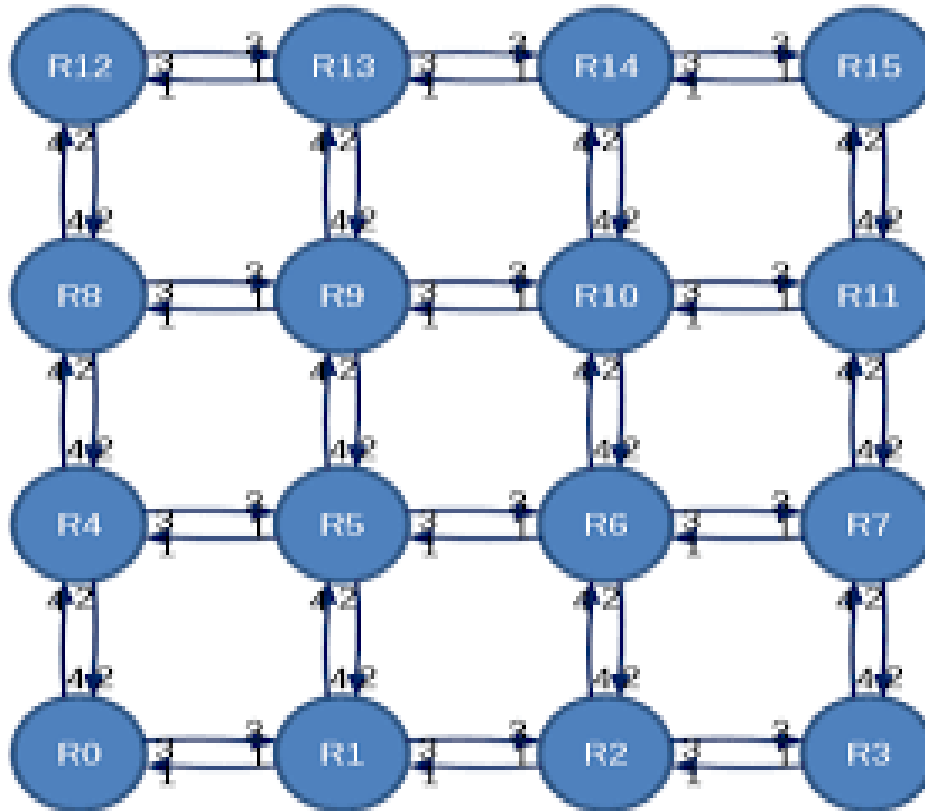


Figure 1. Block Diagram of the 4x4 Mesh Topology

2. Literature Survey

Ahmed Hemani *et al.* introduced NOC methodology as a solution to design productivity problem [1]. Adesh Kumar includes the study to develop large and complex systems on a single chip. This study allows implementation of 2D mesh topological structure supporting physical and architectural level design integration [2]. S. Kumar introduces topological structures for NOC. This paper proposes a packet switched platform for single chip system which scales to arbitrary no. of processors [3]. J.C. Hu and R. Marculescu, envisions a new routing technique called DyAD which combines the advantage of deterministic and adaptive routing schemes and judiciously switches between them on network congestion system [4]. R. Boppana and S. Chalasani present simple methods to enhance current minimal wormhole routing algorithms developed for high-radix, low dimensional mesh network for fault – tolerant routing [6]. S.Q. Zheng and M. Yang proposed PRRA and IPRA design based on simple binary search algorithm for performance of high speed router [7]. E.S. Shin, V.J. Mooney and G.F. Riley contributed automatic generation of round robin token passing bus arbiter to reduce time on design. Design and integration of fast distributed arbiter was also the contribution made by this paper [8]. H.J. Chao, C.H. Lam and X. Guo proposed FCRRRA to handle on-chip and off-chip buses for NOC applications [9]. J. Wang, Y.B. Li, Q.C. Peng and T.Q. tan proposed dynamic adaptive router based on round robin algorithm which detects and change priorities of input ports to elevate router performance [10]. J.M. Jou and Y.L. Lee proposed distinguish paths for data packets so as to avoid collision between them [11].

3. NOC Router

Designing of NOC router, requires vivid understanding of NOC parameters such as Topology, Flow control, Switching methodologies and Routing strategies. Topology is

defined as interconnection of nodes and channels within the network. Researchers proposed various topologies such as mesh, torus, star, spin, butterfly, *etc.* In our design the topology proposed is mesh as it is a regular fashioned topology with easy integration and designing. Messages in this are protected and served to their aligned addresses only. Ideal topology involves characteristics like low latency, high throughput, less power consumption, low cost, high performance, *etc.* It is impossible to meet *all* the advantages of ideal topology because of trade-off between the features. Switching techniques is defined as the way in which internal switches are set so as to connect input of router to output of router. Switching techniques are broadly classified as circuit switching and packet switching. In circuit switching, data packet is transmitted from source to destination only after path is established. In packet switching, there is no requirement of complete path for data to be transmitted from source to destination. Whenever data is available, it is transmitted. Routing strategy is defined as route or path for data packet to reach destination. In routing strategy there must be no occurrence of livelock, deadlock and starvation conditions. In our design, routing strategy involves is X-Y routing. Granting of resources to data packet is managed by flow control. Good flow control strategy is the one which avoids deadlock and bad flow control strategy is the one in which data packet has to wait.

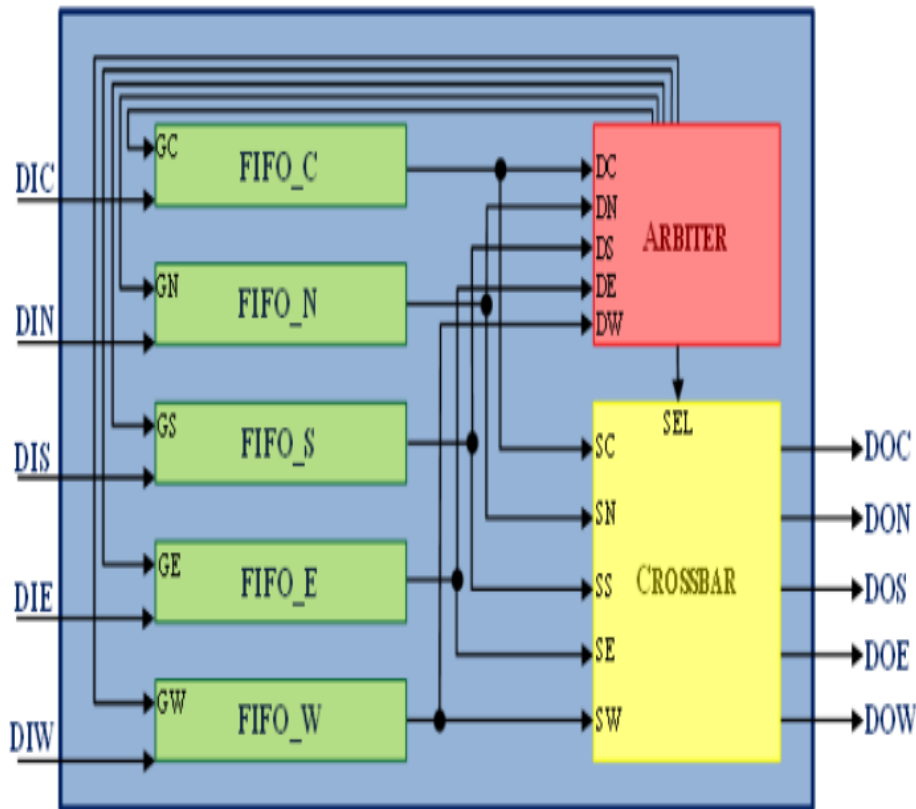


Figure 2. Block Diagram of Router

Router functionality can be estimated by validating the path traversed by data packet inside router. Router designing mainly contains three modules *i.e.*, FIFO Buffer, Arbiter and Crossbar. Block diagram of router is shown in figure 2. When data packet enters the router through one of the input ports of the FIFO Buffer, it sends the destination address of the packet to the arbiter to perform arbitration and generate grant signal of the corresponding FIFO. The arbitration result is send to crossbar module and grant signal is send to FIFO Buffer. FIFO reads the data packet and sends the packet to corresponding

input of crossbar. Then the data packet leaves the router through corresponding crossbar output. FIFO Buffer is a module used to organize and store data packets. As the name suggests FIFO means first input first output, *i.e.*, data packet entering the FIFO first will leave first and then so on. Role of FIFO is important in the era of NOC for packet storage. A FIFO has two data pointers, one for reading from RAM and second for writing into RAM. FIFO first checks for header bit in order to validate the presence of data. Read and Write address are updated using grant signal of a particular port. The data packet ejected from FIFO is send to corresponding input of crossbar and destination address of packet is send to arbiter to perform arbitration process.

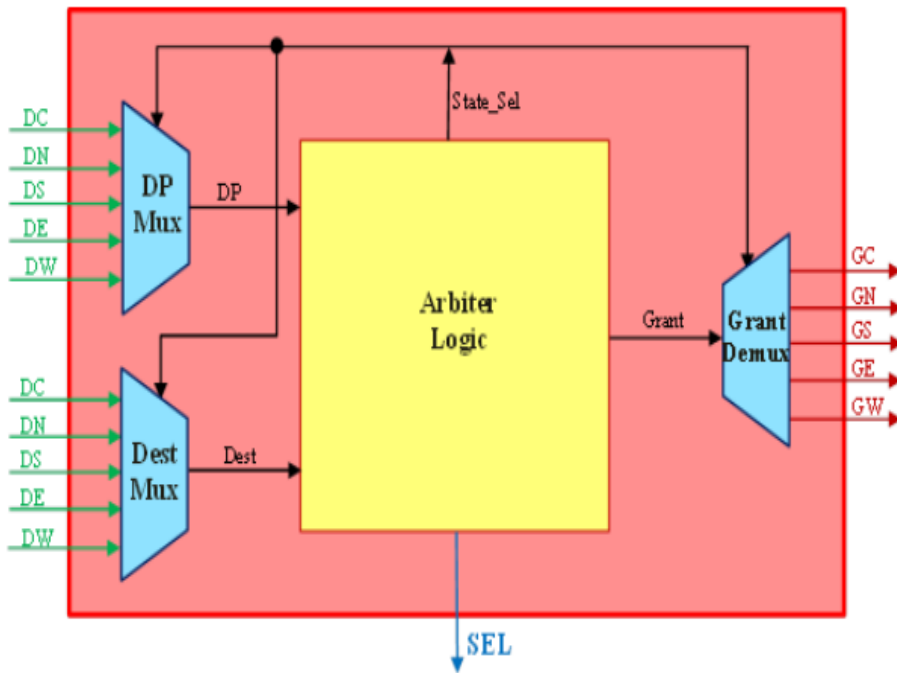


Figure 3. Block Diagram of Arbiter Module

The Arbiter is entitled as the control house of router. The arbiter is used to perform routing computation and round-robin arbitration for selection of five direction ports one at a time. The state diagram of round-robin arbitration is represented in figure 4. Signal 'init' represents that arbiter initially starts with state sc and checks for DP flags. These flags are muxing output of DP lines of five FIFO buffers *i.e.*, dc_f, dn_f, ds_f, de_f and dw_f. On obtaining logic high DP flag, state changes to sc_g, generating logic high grant signal. Then the same state_sel signal and logic high grant signal is demuxed, there by sending a corresponding grant signal to every FIFO buffer *i.e.*, gc, gn, gs, ge and gw. Afterwards, state changes to sc_d, making logic low grant signal and generating select signal for connection between crossbar module.

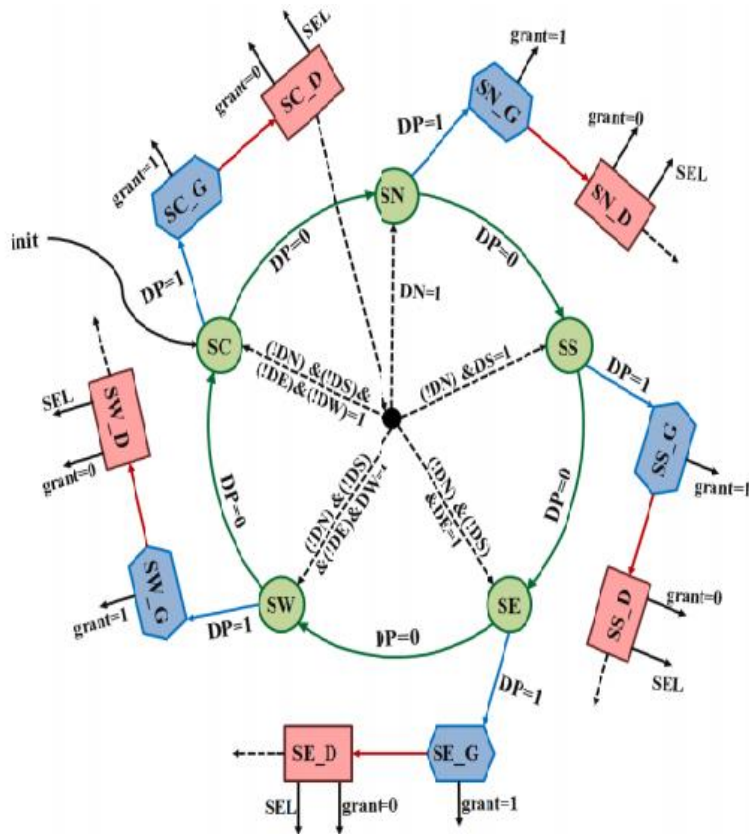


Figure 4. State Diagram of Round-robin Arbitration

After a clock cycle delay, DP flags are checked from each FIFO in round-robin mechanism. For instance, considering sc_d state, in which DN flag is firstly checked, if found logic high, then state changes to sn, otherwise checks for DS flag and so on. If all the flags dn_f, ds_f, de_f and dw_f are logic low, then it again jumps to state sc. Because of this logic implementation, arbitration speed increases when one or more FIFOs do not have data in them. The proposed arbiter block diagram is shown in figure 3. As seen from the block diagram, the proposed module consists of ten input ports, five data present flags and five destination address signal, and six output ports, five grant flags and one select signal. Role of arbiter logic block is played in execution of state diagram. Presence of DP flags is checked by arbiter logic block. After a clock cycle delay, destination address of the packet is read by arbiter logic block. In this implementation, routing strategy inferred is XY routing scheme. Destination ID of the packet is compared with the source address of the packet, and an output port value is obtained which sends corresponding select signal to crossbar to generate connection between present FIFO buffer and output port value. Crossbar is defined as a module containing combination of muxes and demuxes. In our design, as shown in figure 5, crossbar provides 5x4 connections. Connection is made between input port and an output port. This design does not contain feedback. Crossbar can establish only single link at a particular time instant.

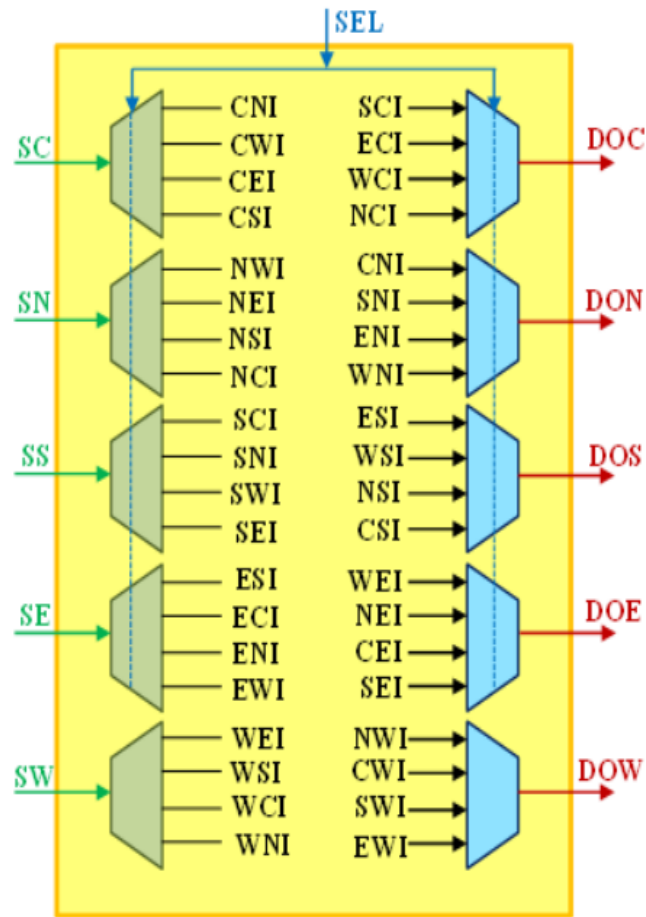


Figure 5. Block Diagram of the Crossbar Architecture

The input digits in binary are Cin, Ein, Nin, Sin, Win and the outputs are Cout, Eout, Nout, Sout, Wout. Each input and output is of 8-bit. This design contains select line down to two lines making 4 possible combinations in binary.

When select = "00",
 Nout<=Cin
 Wout<=Nin
 Cout<=Sin
 Sout<=Ein
 Eout<=Win

When select = "01",
 Wout<=Cin
 Eout<=Nin
 Nout<=Sin
 Cout<=Ein
 Sout<=Win

The remaining process is carried out similarly through different combination as shown in table 1.

Table 1. Combination Table of Crossbar Module

INPORT	SELECT	OUTPORT
C	00	N
C	01	W
C	10	E
C	11	S
N	00	W
N	01	E
N	10	S
N	11	C
S	00	C
S	01	N
S	10	W
S	11	E
E	00	S
E	01	C
E	10	N
E	11	W
W	00	E
W	01	S
W	10	C
W	11	N

4. Results and Discussions

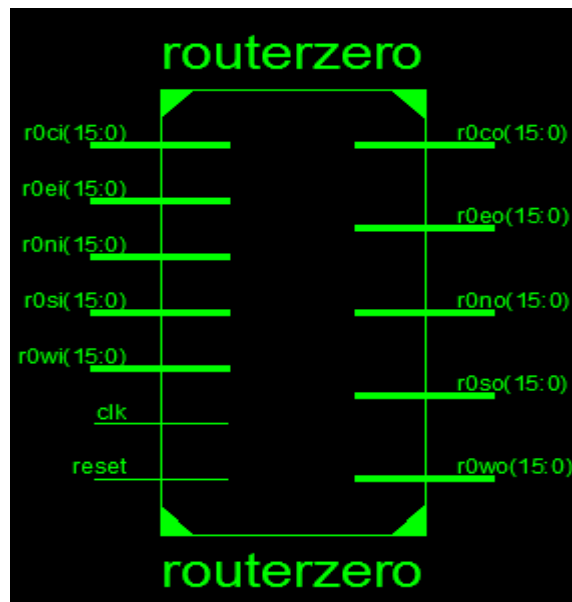


Figure 6. RTL Schematic of Router

RTL Schematic of Router is shown in figure 6. It represents combination of five FIFO Buffers, Arbiter module and Crossbar module. Simulation of router is shown in figure 7. Router architecture proposed in our design utilizes power of 0.620W. The minimum latency shown for data packet to traverse from source to destination is 150ms. Device utilisation summary of router designed is represented in table-2.

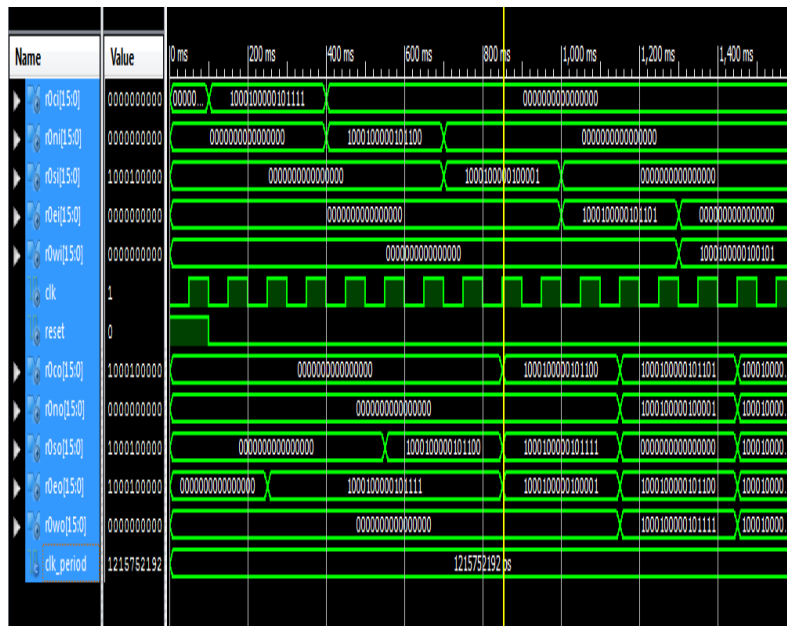


Figure 7. Simulation Result of Router

Table 2. Device Utilisation Summary of Router

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	238	17,344	1%	
Number used as Flip Flops	235			
Number used as Latches	3			
Number of 4 input LUTs	500	17,344	2%	
Number of occupied Slices	298	8,672	3%	
Number of Slices containing only related logic	298	298	100%	
Number of Slices containing unrelated logic	0	298	0%	
Total Number of 4 input LUTs	532	17,344	3%	
Number used as logic	340			
Number used as a route-thru	32			
Number used for Dual Port RAMs	160			
Number of bonded IOBs	162	190	85%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	4.33			

5. Conclusion

This paper initiates router using wormhole concept of switching. FPGA platform is served for router implementation using round-robin routing scheme. In every clock cycle, the proposed router encounters the status of FIFO of input ports and each input port priority is calibrated dynamically. The above designed architecture ensures that all input ports are served with justice. The proposed architecture is designed in VHDL using Xilinx

tools. Its implementation has been done in ISE design suit 14.1 and the functionality of the design has been verified.

6. Future Work

With devices scaling swiftly, SOC designing has inculcated countless obstacles to researchers. A great contribution is made in the field of NOC. An attempt will be bestowed in NOC era to implement 4x4 mesh topology using VHDL for improving network latency for communication.

References

- [1] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oberg, M. Millberg and D. Lindqvist, "Network On Chip: Architecture for billion transistor era", In Proceeding of the IEEE NorChip Conference, (2000).
- [2] A. Kumar, S. Singhal and P. Kuchhal, "Network on Chip for 3D Mesh Structure with Enhanced Security Algorithm in HDL Environment", International Journal of Computer Applications (IJCA), USA, (ISBN:973-93-80871-97-9), vol. 59, no. 17.
- [3] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja and A. Hemani, "A network on chip architecture and design methodology", in: Proceedings of the IEEE Computer Society Annual Symposium on VLSI, (2002), (ISVLSI.02).
- [4] J. C. Hu and R. Marculescu, "DyAD - smart routing for networks-on-chip", In Proc. Design Automation Conference, (2004), pp. 260-263.
- [5] L. Benini and G. D. Micheli, "Networks on chips: a new SOC paradigm", IEEE computer, vol. 35, (2002), pp. 70-78.
- [6] R. Boppana and S. Chalasani, "Fault-tolerant wormhole routing algorithms for mesh networks", Computers, IEEE Transactions on, vol. 44, no. 7, (1995), pp. 848-864.
- [7] S. Q. Zhengy and M. Yang, "Algorithm-hardware co-design of fast parallel round-robin arbiters", IEEE Transactions on Parallel and Distributed Systems, vol. 18, (2007), pp. 84-95.
- [8] E. S. Shin, V. J. Mooney and G. F. Riley, "Round-robin arbiter design and generation", in Proceedings of the 15th International Symposium on System Synthesis, (2002), pp. 243-248.
- [9] H. J. Chao, C. H. Lam and X. Guo, "A fast arbitration scheme for terabit packet switches," in Proceedings of IEEE Global Telecommunications Conference, (1999), pp. 1236-1243.
- [10] J. Wang, Y. B. Li, Q. C. Peng and T. Q. Tan, "A dynamic Priority arbiter for Network-On-Chip", IEEE International symposium on Industrial Embedded Systems, (2009), pp. 252-256.
- [11] J. M. Jou and Y. L. Lee, "An Optimal Round-Robin Arbiter Design for NoC", Journal of Information Science and Engi, vol. 26, (2010), pp. 2047-2058.
- [12] G. Verma, V. Verma, D. Sharma, A. Kumar, H. Verma and K. Kalia, "Design Goal Based Implementation of Energy Efficient Greek Unicode Reader for Natural Language Processing", International Journal of Smart Home, vol. 10, no. 3, (2016), pp. 181-190.
- [13] G. Verma, "Power Consumption Analysis of BCD Adder using XPower Analyzer on VIRTEX FPGA", Indian Journal of Science and Technology", vol 8, iss.17, (2015), IPL160.
- [14] G. Verma, "Low Power Techniques for Digital System Design", Indian Journal of Science and Technology", vol 8, iss. 17, (2015), IPL063.

