

MODELLING AND SIMULATION OF 128-BIT CROSSBAR SWITCH FOR NETWORK -ON- CHIP

Mohammad Ayoub Khan¹ and Abdul Quaiyum Ansari²

¹Centre for Development of Advanced Computing,
Ministry of Communications and Information Technology, Govt. of India
B-30, Sector 62, NOIDA, UP, INDIA
ayoub@ieee.org

²Department of Electrical Engineering
Jamia Millia Islamia, New Delhi, India
aqansari@ieee.org

ABSTRACT

This is widely accepted that Network-on-Chip represents a promising solution for forthcoming complex embedded systems. The current SoC Solutions are built from heterogeneous hardware and Software components integrated around a complex communication infrastructure. The crossbar is a vital component of in any NoC router. In this work, we have designed a crossbar interconnect for serial bit data transfer and 128-parallel bit data transfer. We have shown comparison between power and delay for the serial bit and parallel bit data transfer through crossbar switch. The design is implemented in 0.180 micron TSMC technology. The bit rate achieved in serial transfer is slow as compared with parallel data transfer. The simulation results show that the critical path delay is less for parallel bit data transfer but power dissipation is high.

KEYWORDS

Network-on-Chip, routing, SoC, Crossbar

1. INTRODUCTION

Interconnection structure among the memories and processing elements determines the performance of the system. . There are three basic interconnection structures (a) Shared bus (b) Crossbar switch network (c) Shared (multiport) memories. Among available interconnection structures, shared-bus system is simple and easy to implement. But, at a time only one processing element can access a particular resource; otherwise, bus contention occurs. To avoid contention, a bus controller with an arbiter switch limits bus access to one processor at a time. The bus is not scalable and the system efficiency is low.

The crossbar switch is the interconnecting architecture for high performance systems. In crossbar m vertical processing elements are connected to n horizontal links, whereas n horizontal memories are connected to m vertical links. At each cross section, a switch connects the junctions with control signals. In this network, every processor can access a free memory or

resource independent of other processors. Also, several processors can have access to the memory or resource at the same time. If more than one processor tries to access the same memory or resources, the scheduler in the crossbar should determine which one to connect to. The drawback of the crossbar switch is the number of switches, in this case, $m \times n$. The multiport memory can be used as an interconnection network. All processors have a direct access path to every memory, and the controller inside the memory determines which processor to connect to memory. The complexity that is present in the crossbar is now shifted inside the memory. The realization of memory with such complex logic and multiport is very expensive, even impractical.

Network-on-Chip has a different outlook from conventional interconnection methods as not only it requires the interconnection technology but two more technologies (networking and packet switching fabric technologies) are required for NoC. This requires more advanced interconnection e.g., high-speed and low-power signaling, and on-chip serializer/deserializer. Switching fabric requires buffer and scheduler technologies. Networking technology includes network topology, routing algorithm, flow control and network performance analysis. In this paper, we have implemented 3×2 and 6×6 crossbar switch for serial data transfer and parallel bit data transfer. The crossbar switch is the heart of the router datapath. It switches bits from input ports to output ports. The crossbar switch is the interconnecting architecture for high performance. In this m inputs are connected to m horizontal links, whereas n outputs are connected to n vertical links. The crossbar switch is a fully connected network, where each input is connected to each output. Crossbar switch is of great interest in packet switch designs.

The paper is organized as follows: The section 2 discusses the basics and architecture of various crossbar switches (1-bit, 8-bit, 128-bits) and arbitration logic using DPA. We have also presented the schematic of all the architectures. The section 3 presents analysis on the power and delay for all the three architectures. Finally, a conclusion is presented in last section.

2. ARCHITECTURE OF CROSSBAR SWITCH

In crossbar switch packets are directed to their desired output port. The packets that have been granted passage on the crossbar are passed to the appropriate output channel. The grant is generated from the scheduler or the arbiter of crossbar switch. In virtual channel router has minimum flit size of 128 bits. Therefore, we have implemented 128-bit crossbar switch for virtual channel router to meet the standard. The crossbar switch act as switch traversal, once the grant issued from the scheduler. Scheduler used for crossbar switch is DPA [2]. In DPA request from the input ports arrive at the scheduler for destined output port. Upon, grant is issued from scheduler this will go to the switch fabric. The switch fabric consists of AND OR gates, which in turn passes the input port data to their destined output port. In this work, we have implemented 1-bit, 8-bit and 128-bit crossbar switch. Delay and power for serial bit data transfer and parallel data transfer through crossbar switch has been compared. Parallel bit data transfer provide high data rates at the cost of large chip area, routing difficulty, noise and power. Leakage power increases for parallel bit data transfer. In the following section we will discuss 1-bit, 8-bit, 128-bit architectures for crossbar switch.

A. Serial Bit Data Transfer

The single bit 3×2 switch consists of a crossbar scheduler or arbiter, and a crossbar fabric. Architecture of crossbar switch is given in figure 1. The overall functionality of the switch can be described as follows:

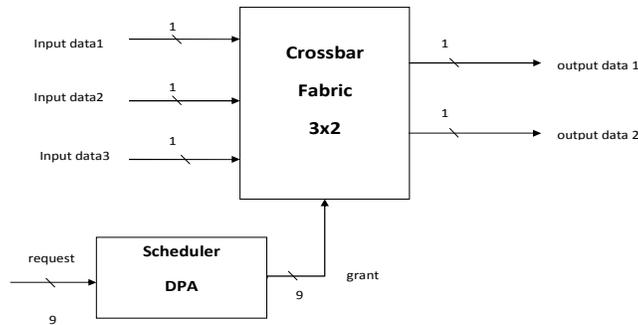


Figure 1: Architecture of 3x2 crossbar switch.

First request comes from the input ports to the crossbar scheduler of the switch for the destination output port. The scheduler grants a request based on a priority algorithm that ensures fair service to all the input ports. Once a grant is issued, the crossbar fabric is configured to map the granted input ports to their destination output ports.

DPA (Diagonal priority arbiter): Here in this crossbar switch we are implementing 3x3 DPA arbiters as delay is less and priority rotations are also possible. The DPA design is that there are some cells in the two dimensional propagation arbiters that are independent of one another, in the sense that granting one of them does not prevent granting the others. The cells that are independent of one another are put in diagonal rows, as shown in Figure 3. Internal structure of single arbiter cell is given below in figure 2.

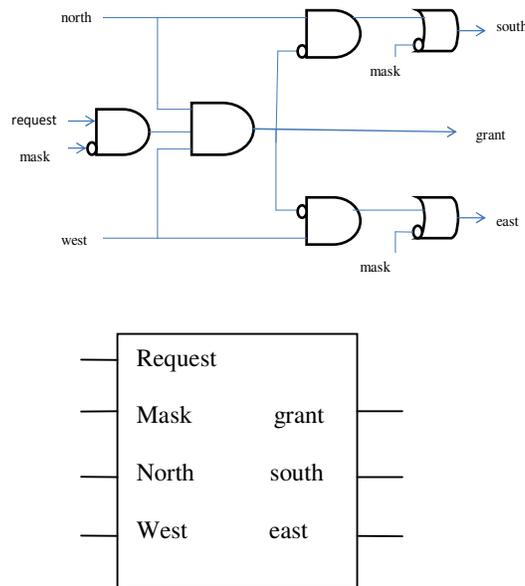


Figure 2: Single arbiter cell for DPA and its symbol

input				output		
Mask	Request	West	north	grant	south	east
0	0	1	1	0	1	1
0	1	1	1	0	1	1
1	0	1	1	0	1	1
1	1	1	1	1	0	0

Table 1: Single Bit-arbiter cell.

The algorithm for DPA is

1. The first (n-1) diagonals of an n x n DPA scheduler are repeated after the last row.
2. The W signals of the first column and the N signals of the first diagonal are assigned to logic one.
3. N² cells (marked by the n x n bold window) are active. We call the bold window “the active window” called MASK
4. The active window moves one step down in every time slot to rotate the priority. When the top most diagonal is diagonal n, the active window has traveled all the way through the DPA scheduler and, therefore, goes back to its starting position shown
5. To implement priority rotations in this design, vector P is introduced.

The algorithm for priority rotations is:

```

set P = "11100".
if P = "00111" then
set P = "11100"
else

```

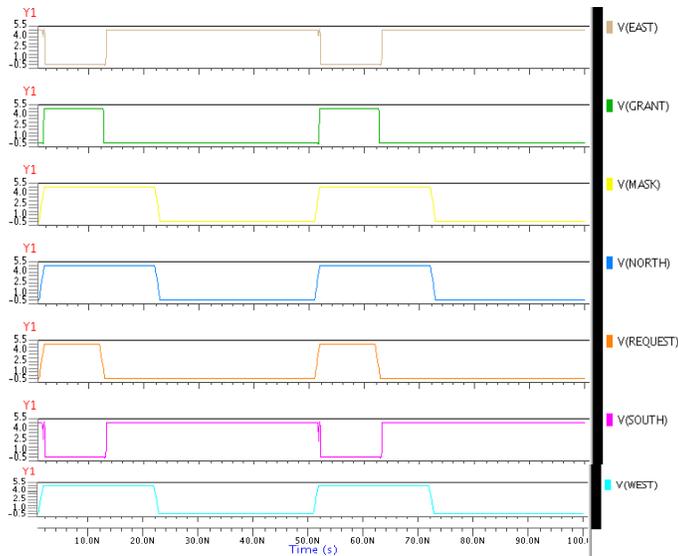


Figure 3: waveform for single arbiter cell

For example, cells (1,1),(3,2), (3,1) ,(1,3) and (2,2) are requesting for respective outputs. Only (1,1),(3,2) have given the respective grant as mask is 11100. In the next cycle, mask is 01110, grant is given to (1,3),(2,2),(3,1).

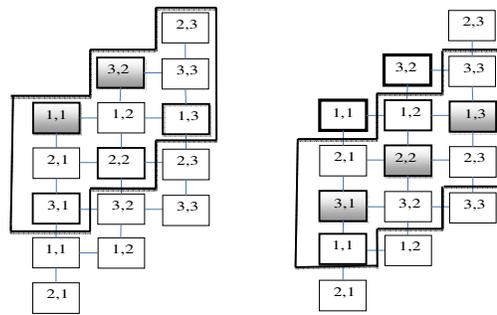


Figure 4:3x3 DPA

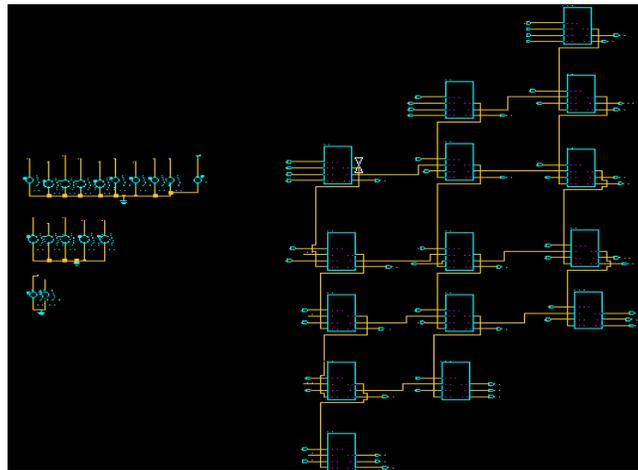


Figure 5: schematic of 3x3 DPA

Input request	Mask	Output grant
R0=1	M1=1	G0=1
R1=1	M2=1	G1=0
R2=1	M3=1	G2=0
R3=1	M4=0	G3=0
R4=1	M4=0	G4=0
R5=1		G5=1
R6=1		G6=0
R7=1		G7=1
R8=1		G8=0

Table 2: 3x3 DPA

Schematic of 3x3 DPA is given in figure 5. In this when all the requests are high and mask is 11100. Therefore, first diagonal has higher priority so grant bit 1, 7, 5 are high. Cells (1, 1), (2, 3), (3, 2) are active. Table 2 explains the working of 3x3 DPA when mask is 11100.

Single Bit Fabric: The fabric connects an input port to an output ports. This is the second module of the given crossbar switch ,which is used for connecting input and its corresponding output depending on the grants issued by the scheduler. Schematic and symbol for single bit fabric is

given in figure 6. In every crossbar the cross points are controlled by the grant input of the fabric module. Each bit of the grant input corresponds to one of the cross points of the crossbar. If a certain grant bit is logic high, then the corresponding cross point is closed. Fabric is establishing a physical path between input and output. Like if grant bit 0, 7, 5 are high, then port 1 input data goes to output port 1 and input data at port 3 goes to port 2.

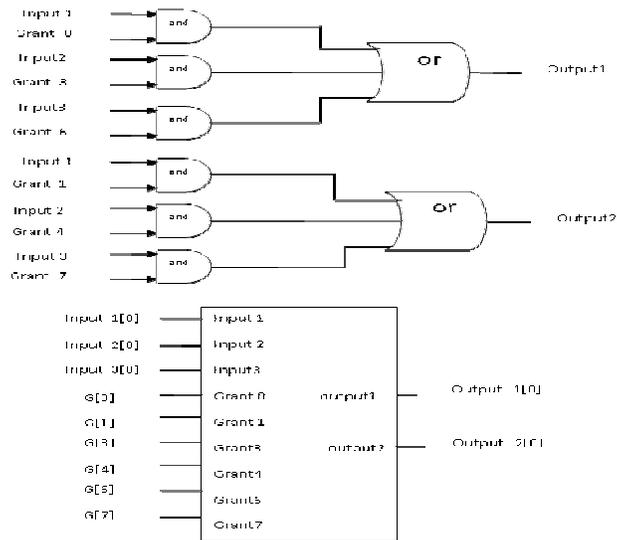


Figure 6: single bit fabric and its symbol

Schematic of 1-Bit Crossbar Switch: In this crossbar switch, input request is of 9 bits for three input ports as each input port is connected to each output port and mask is of 5 bits. If request is 111111111, and mask is of 5 bits. In the first cycle mask is 111000, so cells in the first diagonal has higher priority. Therefore, grant for cells (1,1), (3,2), (2,3) are active so for 3x2 fabric input data from port 1 goes to output port 1 and input data from input port 3 goes to output port 2. Similarly, this process will be repeated for 2nd cycle, now mask is rotated by 1 position. Now, mask is 01110, second diagonal has higher priority, grant for cells (1, 2), (2, 1), (3, 3) are active. Therefore, input data from port 1 goes to output port 2 and input data from port 2 goes to port 1. For 3rd cycle, mask is 00111, grant for the cells in last diagonal are active i.e. (1, 3), (2, 2), (3, 1). Therefore, input data from port 2 goes to output port 2 and input data from port 3 goes to port 1. Waveform for 1bit switch is given in figure 9 for 3rd cycle when mask is 00111.

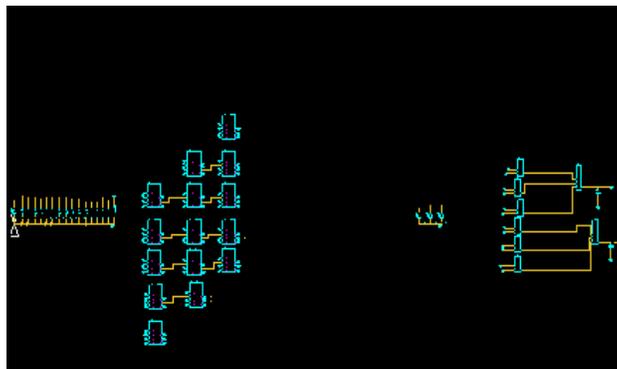


Figure 7: 1bit crossbar switch

B. 8-Bit and 128-Bit Crossbar Switch

In this data is transferred in parallel. According to virtual channel router flit size is 128 bits. Therefore we have modified the architecture of crossbar switch. In this we have used 8 fabric modules for 8bit parallel data transfer and 128 fabric modules in parallel for 128 bit switch. Here, we have shown the results for 8bit crossbar switch in table 3. For 128 bit crossbar switch, waveform will be the same, only input and output data size is 128 bits.

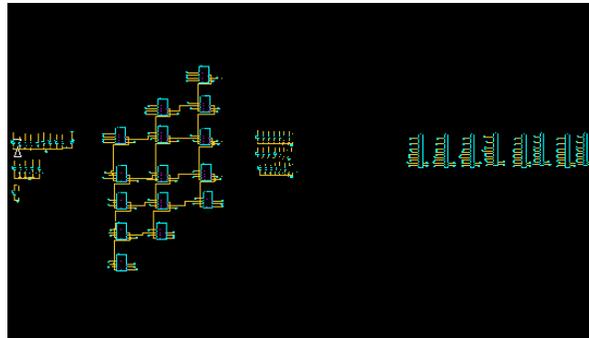


Figure 8: Schematic of 8 bit fabric

Input request	Mask	grant	Active cells
R0=1	M1=0	G0=0	(1,3), (2,2),(3,1)
R1=1	M2=0	G1=0	
R2=1	M3=1	G2=1	
R3=1	M4=1	G3=0	
R4=1	M4=1	G4=1	
R5=1		G5=0	
R6=1		G6=1	
R7=1		G7=0	
R8=1		G8=0	

Input data bit	Cntrl			Output data bit	Input-output ports
	C0=0	C3=0	C6=1		
10000001	C0=0	C3=0	C6=1	11001100	(3,1)
10101010	C1=0	C4=1	C7=0	10101010	(2,2)
11001100	C2=1	C5=0	C8=0	0	0

Table 3: results for 8 bit crossbar switch

For 128 crossbar switch instead of 8 fabric modules we use 128 fabric modules. Similarly, we have implemented 6x6 crossbar switch for 1bit, 8 bit, 128 bit.

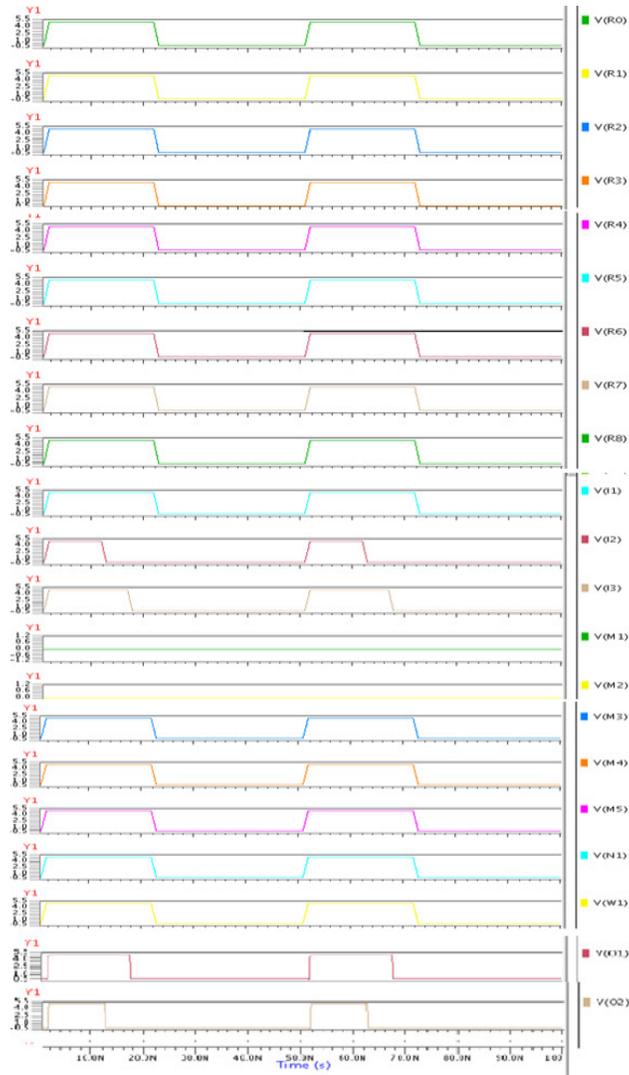


Figure 9: waveform for single bit crossbar switch

3. ANALYSIS AND DISCUSSION

In this section, we have discussed power and critical path delay analysis of 1 bit, 8 bit, 128 bit 3x2 and 6x6 switches. From the above architectures for crossbar switch of serial bit data transfer and parallel bit data transfer we conclude that power dissipation increases for 8 bit and 128 bit in comparison to serial bit data transfer. But data transfer rate increases. At the same time 8-bit and 128-bit of data is available at the same time. Graphs for 1 bit, 8-bit, 128-bit 3x2 and 6x6 crossbar switch is given below in figure 10 and 11. Parallel bit data transfer provide high data rates at the cost of large chip area, routing difficulty, noise and power. Leakage power increases for parallel bit data transfer.

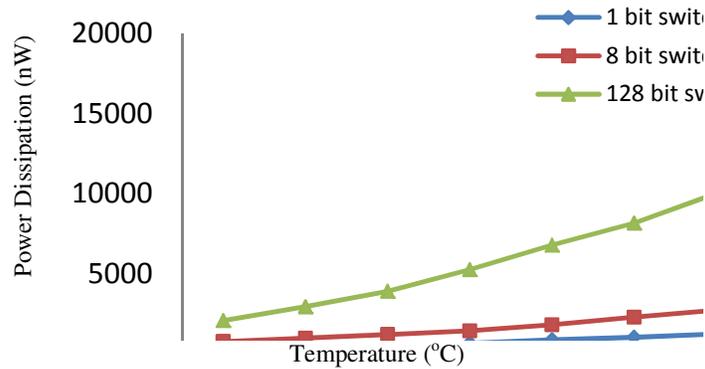


Figure 10: Power Vs temperature graph for 1-bit, 8-bit, 128-bit 3x2 crossbar switch.

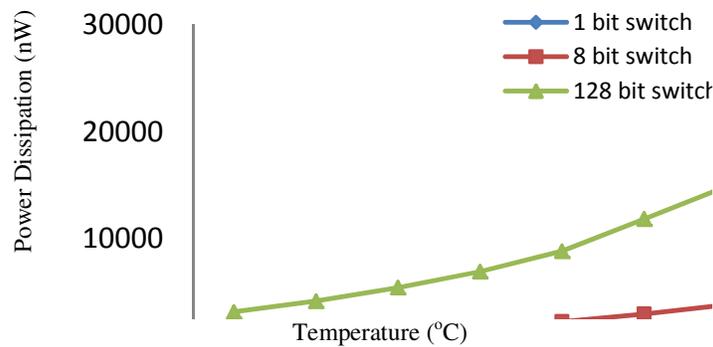


Figure 11: Power Vs temperature graph for 1-bit, 8-bit, 128-bit 6x6 crossbar switch.

Critical path delay is 4.34ns for 1 bit, 8bit and 128 bit 3x2 switch. For 6x6 switch critical path delay is 12.89ns. Therefore at the same time 128 bit data is available at the cost of increase in power dissipation.

Parameters	3x2 switch			6x6 switch		
	1 bit	8 bit	128bit	1 bit	8 bit	128 bit
Power (nW)	281.18	455.055	1300.0	387.98	789.34	2000.97
Delay(ns)	4.34	4.34	4.34	12.89	12.89	12.89

Table 4 : Delay and power analysis of 3x2 and 6x6 switch

4. CONCLUSION

We have presented three architectures of crossbar switch for Network-on-Chip (NoC). This crossbar is targeted for embedded applications. The presented design has an advantage to rotate the priority. This provides fairness in the on-chip network communication. This high performance crossbar is coined with Diagonal Propagation Arbiter. We have concluded that for parallel bit data transfer a higher data rates are achieved at the cost of increase in power and area. The critical path delay obtained is 4.34 ns for 1 bit, 8 bit and 128 bit 3x2 crossbar switches.

ACKNOWLEDGMENT

The authors wish to acknowledge the financial support received from University Grants Commission, Ministry of Human Resource Development, Govt. of India, during the course of this project under the Grant F. No. 39-895/2010(SR) to Department of Electrical Engineering, Jamia Millia Islamia, New Delhi, India.

REFERENCES

- [1] P. Alfke, C. Fewer, S. McMillan, B. Blodget, D. Levi, S. Young, "A high I/O reconfigurable crossbar switch," in *Field-Programmable Custom Computing Machines*, 2003, pp. 3-10.
- [2] Y. Tamir, H.C. Chi, "Symmetric crossbar arbiters for VLSI communication switches", *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 1, pp.13-27, 1993.
- [3] C. Fewer, "Cross Bar Switch Implemented in FPGA," Xilinx White Paper WP166, September 2002.
- [4] J. Hurt, A. May, X. Zhu, and B. Lin, "Design and implementation of high-speed symmetric crossbar schedulers," *Proc. IEEE International Conference on Communications (ICC'99)*, Vancouver, Canada, June 1999, pp. 253-258.
- [5] High-Speed Buffered Crossbar Switch Design Using Virtex-EM Devices, Vinita Singhal and Robert Le, Xilinx paper XAPP240 (v1.0) March 14, 2000, pp:1-7.
- [6] "FPGA Crossbar Switch Architecture for Partially Reconfigurable Systems", by Till Fischer, Karlsruhe Institute of Technology, 7-05-2010, pp:1-88.
- [7] A Parameterized Model of a Crossbar Switch In Bluespec System Verilog(TM) June 30, 2005 © Copyright Bluespec, Inc., 2005.
- [8] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proc. Int. Symp. Comput. Architecture*, Cambridge, UK, Jun. 2004, pp. 188–197.
- [9] Sudeep Pasricha, Nikil Dutt, "On-Chip Communication Architectures". Morgan Kaufmann Publications, U.S., 2008.
- [10] W. J. Dally, "Virtual-channel flow control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, no. 2, pp. 194–205, Mar. 1992.
- [11] Kumar, L. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2007, pp. 150–161.
- [12] Nicopoulos et al., "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," in *Proc. Int. Symp. Microarchitecture*, Dec. 2006, pp. 333–346.
- [13] N. Enright-Jerger, L.-S. Peh, and M. Lipasti, "Virtual circuit tree multicasting: A case for on-chip hardware multicast support," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2008, pp. 229–240.
- [14] Nick McKeown, Martin Izzard, Adisak Mekkittikul, William Ellersick, and Mark Horowitz, "Tiny Tera: a packet switch core," *IEEE Micro*, Jan/Feb 1997, pp. 26-33.
- [15] William James Dally & Brian Towles, "Principles and Practices of Interconnection Networks", Morgan Kaufmann Publishers, U.S., 2004.
- [16] H.Eggers, P.lysaght, H.Dick, and G.McGrefor, Fast reconfigurable crossbar switching in FPGA's. In R.W. Hartenstein and M.Glesner, editors, *Field-Programmable Logic*, pages 297-306, Darmstadt, Germany, September 1996, Springer-Verlag.

Authors

M Ayoub khan is working with Centre for Development of Advanced Computing (Ministry of Communication and IT), Govt. of India as a Scientist, with interests in radio frequency identification, electromagnetic engineering, microcircuit design, and signal processing, NFC, front end VLSI(Electronic Design Automation, Circuit optimization, Timing Analysis), Placement and Routing in Network-on-Chip etc. He has more than six years experience in his research area. He is contributing to the research community by various volunteer activities. He has served as Conference chair in various reputed international conferences like International Conference on Recent Trends in Information, Telecommunications and Computing 2009, Kerala, INDIA, ICMLC 2010, ICSEM 2010, International Conference on Recent Trends in Business Administration and Information Processing, 2010, Trivandrum, Kerala, India, ICIII 2010, to name a few. He is member of professional bodies of IEEE, ISTE, IACSIT, ACEE and IAENG. He may be reached at ayoub@ieee.org



Prof A. Q. Ansari is a Ph.D (Hierarchical Fuzzy Systems) from Jamia Millia Islamia, New Delhi (2000), M. Tech (Integrated Electronics and Circuits) from I.I.T. Delhi (1991), and B.Tech. (Low Current Electrical Engineering) from AMU, Aligarh (1984). Prof. Ansari is a C. Eng. and Fellow, Institution of Engineers (India); C. Eng. and Fellow, Institution of Electronics and Telecommunication Engineers (IETE), India; C. Eng. and Member, IET, U.K. (formerly IEE, U.K.); Fellow, National Telematics Forum, India; Sr. Member, IEEE, U.S.A.; Sr. Member, Computer Society of India (CSI), Life Member, Indian Society for Technical Education (ISTE), Life Member, Indian Science Congress Association and Life Member, National Association of Computer Educators and Trainers (NACET), India. He may be reached at aqansari@ieee.org

