

Analyzing the Performance of Mesh and Fat-Tree topologies for Network on Chip design

Vu-Duc Ngo, Huy-Nam Nguyen, and Hae-Wook Choi

System VLSI Lab Laboratory, SITI Research Center, School of Engineering
Information and Communications University (ICU)
Yusong P.O. Box 77, Taejon 305-714, Korea
{duc75,huynam,hwchoi}@icu.ac.kr

Abstract. The demand of integration of many heterogeneous semiconductor intellectual property (IP) blocks has been introducing a new chip design paradigm so called on chip network. This paradigm promisingly offers a packet switched network among IPs to reduce the main problems in the very deep sub micron technologies that arise from non-scalable global wire delay, failure to achieve global synchronization, errors due to the signal integrity, non-scalable bus based functional interconnection, etc. This paper introduces interconnected or switched network topologies and also analyze their performances in terms of communication protocol related to the issues such as routing strategy, buffer size, routing algorithm, etc. The above mentioned evaluations are done by utilizing the tool that has been widely used in the research domain of computer network design, so called NS-2.

1 Introduction

The continuation of scaling down of transistor size is leading to the fact that a large number of IPs can be integrated on a single chip. This integration, according to [1], gives rise to new challenges. The most prominent issue allied with the mass integration design comes up with non-scalable global wire delay [2]. Though sub-micron technology creates the good chance to reduce the gate delays, global wire delays significantly increase or remain constant by including repeaters. This matter also leads to the high consumption of silicon area and power. The other critical concerns related to forthcoming technology and current NOC design methodology are global synchronization and non-scalable bus based functional interconnection (especially when the number of integrated IPs is numerous). To overcome the above mentioned problems, we can propose the use of switching based topologies that have been extensively used for computer network to integrate IPs. Another word, the on-chip topology resembles the interconnect architecture of high performance parallel computing systems [3]. By using this design approach, the communications between IPs can be the packet based. We also realize that the needs of global wire can be omitted. The wires are now used only as the connections between switches. Also, we do not any longer worry about global synchronization due to decoupling of the processing

nodes. To get over the induced noise caused by unreliable physical channels, the packet based error control mechanisms can be applied. The two promising on chip network topologies have been mostly proposing are Mesh and Fat-Tree, respectively. The comparison of performance evaluation of the two topologies with respected to physical constraints was made by Petrini and et al. [4]. The comparison showed clearly that the Fat-Tree topology proposed firstly by Leiserson [5], with several merits such as:

- *Scalability; Reliability; Regular and simple topology; High throughput; Good performance,*

has superior performance to that of the Mesh topology. To verify these above mentioned merits in terms of on chip network topology design approach, we need to carry out the high level simulation Fat-Tree topology in the scenario of on chip network implementation. However, the high level simulation of on chip network is still in its infancy. Choosing simulation tool and defining the physical constraints for it are now at the beginning stage. There is no gate level simulation tool, which is known as the best method to evaluate the physical behavior of SOC design, can be developed to adapt to on chip network design and verification. To conquer this limitation we choose network simulation tool to simulate the on chip network with respect to high level approach. Among network simulation tools, NS2 [6] emerges to be the best solution to simulate the high level of on chip network architecture. The NS2 is widely used in the research for design and evaluation of computer network domain. We take the advantage of NS2 in terms of communication protocol (OSI protocol) to simulate and verify the performance of our proposed interconnection architecture for on chip network's application, including topologies selection, the design of router, and routing schemes, as well as queuing schemes. In [7] and [8], the authors proposed the Mesh and Fat-Tree architectures for on chip network topology design, respectively. However, the performances of these architectures are not yet mentioned in terms of high level of topology evaluation. In this paper, we report several experiment results of general NoC and H.264 video decoder design based on the simulation of on chip network using NS2 for Mesh and Fat-Tree architectures with the relevant routing and queuing schemes. By using these simulation results with certain comparisons, we propose to use Fat-Tree as on chip network interconnection topology. Our paper is organized as follows: The interconnection architectures and routing schemes are introduced and compared in section II. The NS2 and its mapping to on chip network is conducted in section III. The simulation results and their comparisons are analyzed in section IV. Finally the conclusion and future work will be given in section V.

2 Interconnection architectures and Routing schemes

2.1 Interconnection architecture

Mesh architecture: The k -ary d -dimensional Mesh architecture is built by its dimension d and radix k . This follows that the total number of switches is

k^d . The k^d switches are organized in an $d - dimensional$ grid such that there are k switches located in each dimension and wrap-around connections. Since the number of IPs that can be connected to one switch is $d - 1$, it follows that the total number of mounted IPs is clearly calculated by

$$N_{Mesh} = k^d(d - 1). \quad (1)$$

We denote b as the unidirectional bandwidth, the total bandwidth of the network is obtain by

$$B_{Mesh} = 2k^d b. \quad (2)$$

Moreover, this architecture offers a simple connection scheme. Therefore, the Shortest Path routing is mostly applied on it. Also the number of the mounted IPs is relatively high compared to the total bandwidth. So that the performance of Mesh architecture is not high.

The Fig.1 shows an example of $4 - ary 2 - dimensional$ Mesh architecture. We can see in this figure the number of switches and the number of the mounted IPs are both equal 16.

Fat-Tree architecture: The Fat-Tree is an indirect interconnection network based on a complete binary tree, which was proposed by Leiserson [5]. There is a significant difference from the traditional tree in computer science, Fat-Tree copies image from real tree that it gets as thicker as it approaches the root. A set of processors are located at the leaves of the Fat-Tree and each edge of the underlying tree corresponds to a bi-directional channel between parent and child. The bandwidth of the Fat-Tree increases as it goes closer to the root. This feature makes Fat-Tree to be sufficiently applicable with advanced routing algorithm as the Distance Vector instead of Shortest Path. Furthermore, since the depth of the tree is calculated by the dimensional metric plus 1, it follows that the size or the number of IPs that can be mounted on one certain Fat-Tree topology depends on the depth of the tree. As the depth of the tree increases, the number of IPs as well as the total capacity of the network gets bigger. These two significant characteristics lead Fat-Tree having better performance in comparison with Mesh architecture. More analytically, we introduce some calculations in $k - ary d - dimensional$ Fat-Tree architecture, which is shown in Fig.2 (case of $k = 4, d = 2$).

As can be seen in the Fig.2 the fat nodes are the switches that are located in the higher degrees of the tree and they can be considered as the combination of $2^{(t_d-1)}$ single nodes, where t_d presents the node's degree on the tree. The number of mounted IPs $N_{Fat-Tree}$ and the number of switches $S_{Fat-Tree}$ are straightforwardly calculated as follows:

$$N_{Fat-Tree} = k^d, \quad (3)$$

$$S_{Fat-Tree} = k^{d-1}d, \quad (4)$$

respectively. Therefore with the unidirectional bandwidth defined by b , the total bandwidth is presented by

$$B_{Fat-Tree} = 2k^d db. \quad (5)$$

We carry out the comparison of this above total bandwidth with the case of the k -ary d -dimensional Mesh's total bandwidth, which is shown in (1). With $d \geq 2$, the total bandwidth of the k -ary d -dimensional Fat-Tree architecture is equal d times of the total bandwidth of k -ary d -dimensional Mesh architecture. Since more available total bandwidth is offered, it follows that the more propagation delay can be decreased. This fact is very good factor to consider Fat-Tree as a appropriate candidate for on chip network topology. However, the designers have to accept the higher network cost or complexity.

2.2 Routing schemes

Shortest Path: The distributed Shortest Path routing scheme [12] is an iterative procedure to find out the routing table which contains the shortest paths from all the nodes of the network to each common destination which also belongs to the network. Let one of the common nodes be defined as x . In order to keep track of the shortest paths, we label each non-destination node i with a pair (η, d_i^x) where d_i^x represents the current iteration for the shortest distance from the node to x and η is the number of next nodes along the currently computed shortest path. The iterative algorithm is presented as the following 4 steps:

- Initialization: Set the initial value for the destination node x :

$$d_x^x = 0, \quad (6)$$

and label all other non-destination nodes $(\bullet, +\infty)$.

- Updating: Update d_i^x for each non-destination node i by checking the current value of d_j^x for each neighbor node j and carrying out the operation:

$$d_i^x \leftarrow \min_j [d_j^x + l(i, j)], \quad (7)$$

where $l(i, j)$ denotes the length of the link from node i to node j .

- Node j 's label is completely updated by replacing the metric η by the number of the neighbor nodes which minimize the (7).
- Step 2 is repeated at each node until no further changes occur.

This above algorithm is terminated with the output of routing table. By this table the data packet with a destination node identifier in its header can reach it's desired destination node with a certain route that contains specifically preferred neighbors assigned by predetermined routing table. This fact significantly affects network performance due to its static characteristic. We will discuss this issue in the IV section. However, this routing algorithm is still widely applied over others for implementation within operational networks due to its simplicity.

Distance Vector: In this context we reintroduce the Distance Vector routing algorithm [12] that was proposed relying on Distributed Bellman-Ford algorithm. In this algorithm, each node i maintains a set of distances d_{ij}^x where j varies over the neighbors of i . Moreover, node i treats its neighbor k as the

next intermediate node that the data packet visits before reaching the given destination x if:

$$d_{ik}^x = \min_j \{d_{ij}^x\}. \quad (8)$$

The succession of next intermediate nodes selected in this above method leads to x along the shortest path. To keep track of the distance estimation up-to-date, each node monitors the length metrics of its available outgoing links. It periodically broadcasts its current estimate of the shortest distance to every other node inside the network to each of its neighbors. By checking out the available outgoing links, this algorithm outperforms to that of Shortest Path algorithm. The comparisons will be performed by simulation in section IV.

3 NS2 and its on chip network mapping

The network simulator, NS2, with its capabilities of describing network topologies, network protocols, routing algorithms, packet scheduling schemes, as well as traffic generation methods, has been broadly used in the field of computer network design and simulation. Referring to the article [9], the authors used NS2 to simulate the Mesh architecture with dimension of 5x5 for on chip network topology. However, only the simplest cases of routing algorithm, routing strategy, queuing algorithm, transmission protocol are defined as simulation constraints. In this paper, we not only carry out the simulations for 2 – *dimensional* Mesh architecture with the advanced issues of those mentioned constraints but also for 2 – *dimensional* Fat-Tree architecture to make a comparison and choose a good candidate for on chip network topology.

For both of Mesh and Fat-Tree topologies, one IP, called resource, is always connected to one switch, called router. The interface between IP and switch is called Resource Network Interface (RNI). The functions of RNI cover 4 upper layers of the internet 5 layers model. The physical layer is functioned by the linkages between resources and routers (Rs2Rt) or between routers and routers (Rt2Rt). In NS2 environment the transmission protocol can be defined as UDP or TCP. The exponential traffic generator model is applied. The routing strategies can be: Shortest path, Dynamic or Distance Vector (DV), according to the choice of two architectures (Mesh or Fat-Tree). Since IPs (resources) of on chip network platform can be CPU, RAM, DSP, FPGA, USB, etc, the demand of communication loads originated from each IP are different. This leads us use the fair queuing scheme like Stochastic Fairness Queuing (SFQ) instead of Drop-Tail introduced in [9]. The uses of FQ and SFQ for our mentioned architectures achieve overperformance to that of the DropTail in terms of drop probability, delay, as well as throughput. The Table. 1 shows the summary of mapped NS2 constraints for our experiments on the two mentioned architectures.

4 Simulation results and discussion

In this paper, we firstly simulate several cases with different constraints of routing and queuing algorithms as well as buffer sizes for the 4x4 Mesh and 4 – *ary*

Table 1. Application of On chip network constraints to NS2.

On chip network model	Applied NS2 constraints
Connections	Rs2Rt and Rt2Rs
Transmission protocols	UDP and TCP
Packet size	8 bytes
Buffer size	8, 16, 32 (packets)
Queuing schemes	DropTail, DRR, RED, SFQ
Routing schemes	Shortest Path, Distance Vector
Traffic generator	Exponential
Bisection bandwidth (Max)	8 Gbits (expected)

2 – *dimensional* Fat-Tree architectures. In detail, we simulate these two architectures with Shortest Path and Distance Vector routing algorithms, also we attach the congestion control or Queuing algorithms such as DropTail, RED (Random Early Detection), DRR (Deficit Round Robin), and SFQ (Stochastic Fairness Queuing) to the switches of both architectures. Finally, we compare the simulation results of all simulated cases to choose the best combination of embedded routing and queuing algorithms for each topology. We also compare the performances offer by two architectures in the cases of using the similar options of routing, queuing, as well as the buffer sizes. We already defined the on chip network model and it mapped constraints onto NS2 simulator in the Table. 1. Due to the limitation of the computer capability, we just limited the data rate of each resource up to 200Mbps. The resources are activated to generate data packets randomly. The Fig. 3 and Fig. 4 depict the aggregative throughput of 4x4 Mesh topology vs. different options of applied queuing schemes and buffer sizes, respectively. As can be seen in the Fig. 3, with the same buffer size as well as routing scheme (Shortest Path), the use of SFQ queuing scheme offers us the best performance with respect to total throughput on 4x4 Mesh topology. The Fig.4 shows that the bigger buffer size is applied, the higher throughput is achieved. The throughputs of Fat-Tree topology with queuing schemes of SFQ and DropTail are depicted in Fig. 5 and 6, respectively. We realize that by applying Distance Vector routing algorithm to Fat-Tree topology, with its own merits as pointed out in previous sections, the system throughput can be increasingly achieved. In the last figure, Fig. 7, we compare Fat-Tree and Mesh topologies with the same buffer size of 8 packets in terms of system throughput related to different choices of routing and queuing schemes. We can see that the Fat-Tree topology with the embedded routing algorithm of Distance Vector and queuing algorithm of SFQ gains highest throughput. In the stable period, this design achieves approximately 250Mbps higher to that of Mesh topology using Shortest Path and DropTail schemes [9].

More specifically, we implement the H.264 video decoder on those mentioned NoC architectures. The IPs and their connections represented by the directional data rates are defined by the table in Fig. 8. There are 11 IPs that are allo-

cated onto 11 corresponding switches. To maximize the throughput transactions among the IPs, we adopt the Branch and Bound technique presented in [13] for the cost function of data rates. The results of optimum mapping shows that the IPs which are connected by the connections of high data rates should be allocated next to each others. This mapping reduces the drop packets that is very high rate if we allocate the mentioned IPs far away since huge number of packets from the original IP need to go through a lot of intermediate switches before reach the destination. In order to make this argument clearly, two mapping schemes of IPs onto the 4x4 Mesh architecture are simulated as depicted in Fig. 9. As seen in this figure, the throughputs of IPs of random mapping is shown by Fig. 9a are almost similar to that of the throughputs of IPs of the optimum mapping shown by Fig. 9b, but the corresponding bisectional bandwidth required for random case of 30Mbps is much more higher to that of 15Mbps for the optimum case. These two simulations are done with SFQ queuing scheme and DV routing scheme. The last experiment is carried out to show the superior performance of using Fat-Tree compared to Mesh architecture. As seen in Fig. 10, with bisectional bandwidth of 20Mbps, not only the system requirement of throughput is met but also the throughputs of IPs are almost similar to the case of random mapping of Mesh architecture with the bisectional bandwidth of 30Mbps.

In conclusion, we propose to use Fat-Tree topology with embedded Distance Vector routing and SFQ queuing schemes as on chip network switching core topology.

5 Conclusion

In this paper, we compare the performance of Fat-Tree and Mesh architectures in the sense of on chip network design methodology. We also carry out the high level simulation of on chip network based on Mesh and Fat-Tree architectures using NS2 to verify our analysis. The simulations of the Mesh and Fat-Tree architectures are done with advanced routing algorithms and router queueing mechanisms for general NoC design as well as H.264 video decoder. Finally, by numerical and simulation results we can conclude that the Fat-Tree architecture is suitable for on chip network switching core. By utilizing Fat-Tree architecture and Distance Vector routing algorithm as well as Stochastic Fairness Queue algorithm, we can achieve the best combination for our design in terms of choosing on chip network topology. It has superior performance to that of Mesh architecture, in which Shortest Path routing algorithm is applied, with respected the system throughput.

References

1. L. Benini and G. DeMicheli, "Networks On Chips: A new SoC paradigm," IEEE computer, Jan, 2002.

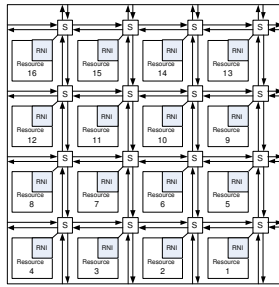


Fig. 1. 4-ary 2-dimensional Mesh topology.

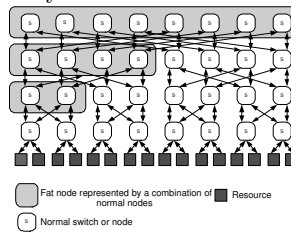


Fig. 2. 4-ary 2-dimensional Fat-Tree topology.

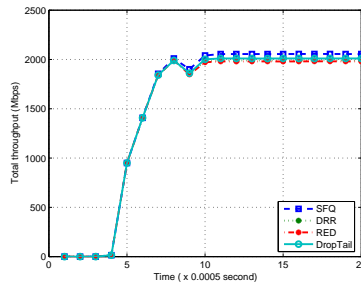


Fig. 3. Throughput of Mesh (Shortest Path) vs. Queuing Schemes.

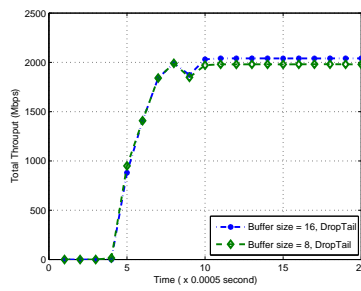


Fig. 4. Throughput of Mesh (Shortest Path) vs. Buffer sizes.

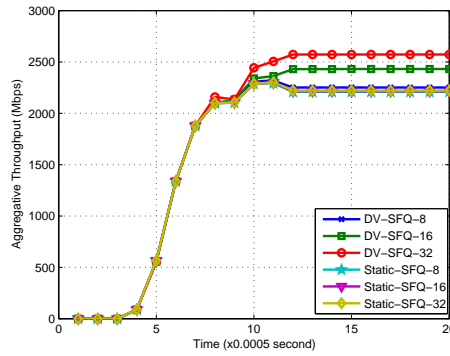


Fig. 5. Throughput of Fat-Tree (SFQ queuing schemes) vs. Buffer sizes and routing schemes.

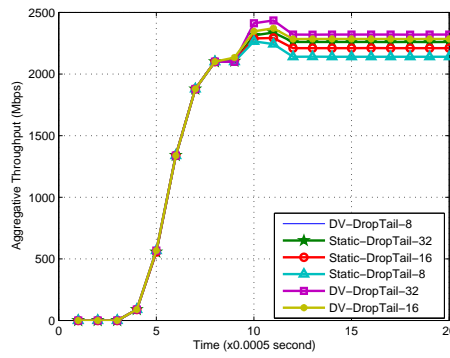


Fig. 6. Throughput of Fat-Tree (DropTail queuing schemes) vs. Buffer sizes and routing schemes.

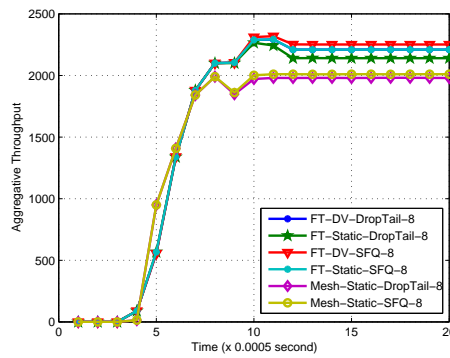
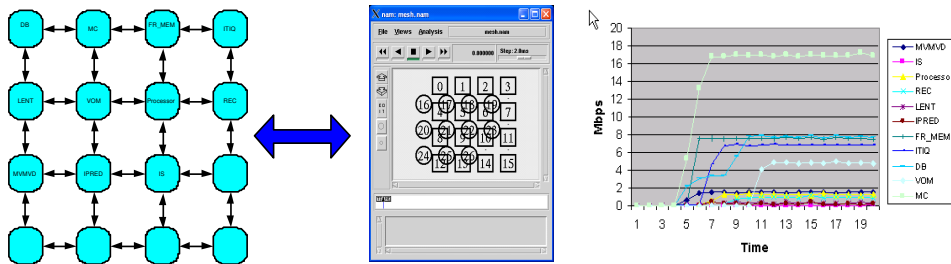


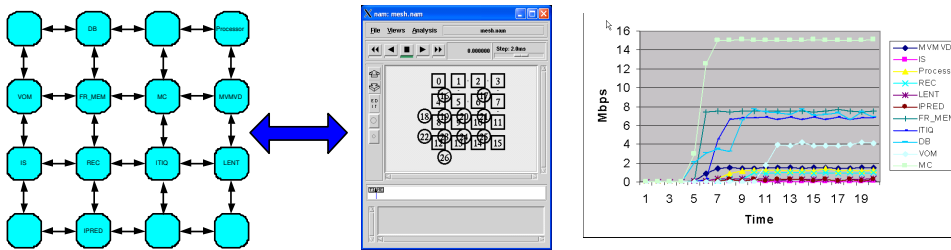
Fig. 7. Throughput Comparison of Fat-Tree (FT) and Mesh topologies.

<i>FromTo</i>	DB	MC	FR_MEM	ITIQ	LENT	VOM	Processor	REC	MVMVD	IPRED	IS
DB	.	.	7114
MC	4475	.	345
FR_MEM	2650	10652	.	.	63	4802	.	40	.	.	.
ITIQ	.	5949	642	.	.	.
LENT	.	.	.	6592	.	.	408	.	496	.	.
VOM
Processor	326	322	.	186	232	12	.	7	922	59	12
REC	321	13	27	145	.
MVMVD	↔	↔	↔	↔	↔	↔	828	.	↔	↔	↔
IPRED	↔	↔	↔	↔	↔	↔	.	321	↔	↔	↔
IS	↔	↔	63	↔	↔	↔	↔	↔	↔	↔	↔

Fig. 8. IPs and their connections (in kbps) of H.264 video decoder.

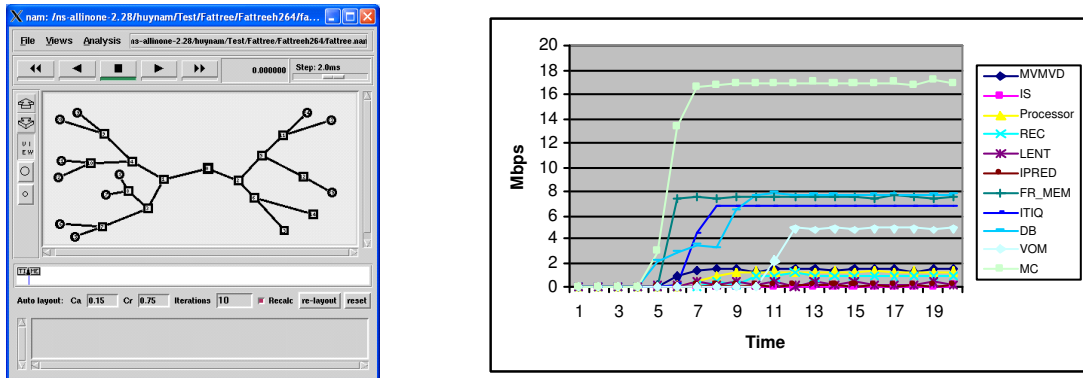


(a). Bisected bandwidth of 30Mbps, SFQ, DV



(b). Bisected bandwidth of 15Mbps, SFQ, DV

Fig. 9. IPs allocation schemes their throughputs (in Mbps) of H.264 video decoder with Mesh architecture: (a) random allocation, (b) optimum allocation.



Bisectional bandwidth of 20Mbps, SFQ, DV

Fig. 10. IPs allocation scheme their throughputs (in Mbps) of H.264 video decoder with Fat-Tree architecture.

2. M. A. Horowitz et al., "The future of wires," *Proceeding of IEEE*, Vol. 89, Issue. 4, pp. 490-504, Apr 2001
3. P. Guerrier, A. Grenier, "A generic architecture for on-chip packet-switched interconnectio," *Design automation and test in Europe conference*, pp. 250-256, Aug 2000.
4. F. Petrini and M. Vanneschi, "Network performance under Physical Constrains," *Proceedings of International Conference on Parallel Processing*, pp.34 - 43, Aug 1997.
5. C. E. Leiserson, "Fat Trees: Universal networks for hardware efficient supercomputing," *IEEE Transactions on Computer*, C-34, pp. 892-90,1 Oct 1985.
6. Ns2: www.isi.edu/nsnam/ns/.
7. H. Kariniemi, J. Nurmi, "New adaptive routing algorithm for extended generalized fat trees on-chip," *Proceedings of International Symposium on System-on-Chip*, pp. 113 - 118, Nov 2003.
8. P. P. Pande, C. Grecu, A. Ivanov, R. Saleh, "Design of a switch for network on chip applications," *Proceedings of the 2003 International Symposium on Circuits and Systems*, pp. V-217 - V-220 vol. 5, May 2003.
9. Yi-Ran Sun, S. Kumar, A. Jantsch, "Simulation and Evaluation for a Network on Chip Architecture Using Ns-2," *Proceeding of 20th IEEE Norchip Conference*, Nov 2002.
10. D. Berozzi, L. Benini and G. DeMicheli, "Low-Power Error Resilient Encoding for On-chip Data Buses," *DATE-International Conference on Design and Test Europe*, Jun 2002.
11. J. Muttersbach et al., "Globally-asynchronous locally synchronous architectures to simplify the design of on-chip systems," *Proc. 12th Annual IEEE International ASIC/SoC Conference*, pp. 317-321, Sep 1999.
12. D. Bertsekas and R. Gallager, "Data Networks," Chapter 5., Second Edition, Prentice-Hall, Inc., 1992.
13. S.Murali and G.De Micheli, "Bandwidth-Constrained Mapping of Cores pnto NoC Architectures," *DATE, International Conference on Design and Test Europe*, 2004, pp. 896-901.