Hierarchical Graph: A New Cost Effective Architecture for Network on Chip

Alireza Vahdatpour[1], Ahmadreza Tavakoli[2], and Mohammad Hossein Falaki[1]

[1] Computer Engineering Department,
Sharif University of Technology
{vahdatpour, falaki}@ce.sharif.edu
[2] Electrical Engineering Department,
Sharif University of Technology
artavakoli@ee.sharif.edu

Abstract. We purposed a new Network on Chip (NoC) architecture called Hierarchical Graph. The most interesting feature of this novel architecture is its simple implementation process. Furthermore, the flexible structure of this topology makes it suitable for use in application specified chips. To benchmark the suggested architecture with existing ones, basic models of physical implementation have been extracted and simulated using NS-2. The results compared with the common used architecture Mesh show that HG has better performance, especially in local traffics and high loads.

1 Introduction

As predicted in [1], we will enter the *Ultra Large Scale Integration* (ULSI) design world with 45nm technology in 2010. Integration of multibillion transistors will be possible and device delays will get smaller. The dominant issue in design would be the interconnection between transistors.

The closer we get to large chip integration, the more we need smart design plans. Currently one of the most accepted plans is *System on a Chip* (SoC). In SoC methodology a number of IP-cores (FPGAs, mixed signal blocks, processors, memory blocks etc.) are placed together on a single chip, with proper systematic interconnections. Today the most common method of interconnection is BUS. While the number and complexity of cores are increasing in *Ultra Deep Submicron*, interconnections become more challenging.

Computer networks have been successful in handling the problems of managing large number of resources. This has led to a new paradigm in the design of SoC systems, called *Network on Chip* [2] which is compatible with *Globally Asynchronous Locally Synchronous* (GALS) design method. NoC design uses protocols similar to OSI reference model of networking [3], as the interconnection facility between IP cores. Currently, the most common network architectures for NoC systems are Mesh and Fat-Tree [4].

In this paper we have introduced a new architecture for NoC systems named *Hierarchical Graph* (HG). Some related work is addressed in the next section and then various aspects of the proposed architecture are introduced in section 3. In section 4 some physical properties of the implementation of HG are discussed and then the performance of the new architecture is compared with Mesh through the simulation. Finally the results of simulations are presented in section 5.

2 Related Work

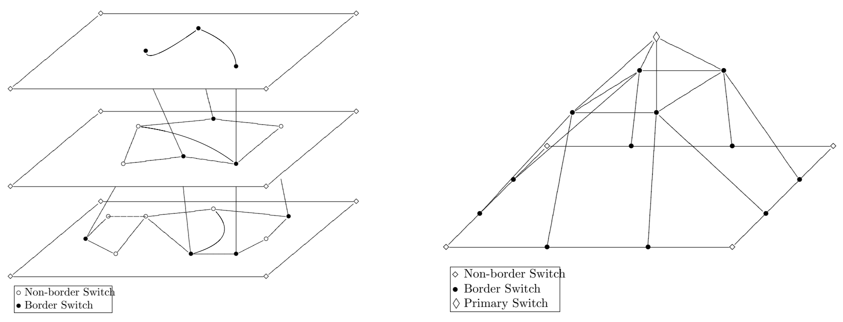
There are many research groups working on SoC systems from architectural point of view. In [4] Shashi Kumar et al. introduced a new methodology for designing mesh architecture for NoC. Lacking specific tools for evaluating networks on chip, the performance of Mesh and Fat-Tree were compared using Network Simulator (NS-2) in [5] and [6]. Some useful strategies were introduced by Daniel Wiklund et al. in [7] for benchmarking on chip networks.

3 Architecture

Hierarchical Graph (HG) network topology is designed specially for NoC systems to be compatible with Ultra Deep Submicron fabrication technologies.

Fat-Tree as a network topology has good performance, but its irregular and intensive interconnection lines make it impractical to be implemented on a chip. Although the throughput of HG is not as good as Fat-Tree, it can be implemented with a single metal layer because of its simple interconnection rules. Decreased ratio of switch per resource and simpler switch implementations decreases the implementation cost.

As the name suggests, HG consists of a hierarchical structure of plain graphs (see Fig.1(a)). In each level, there is a graph of interconnected switches, some of which named *Border Switches* are responsible for connecting the graph of this



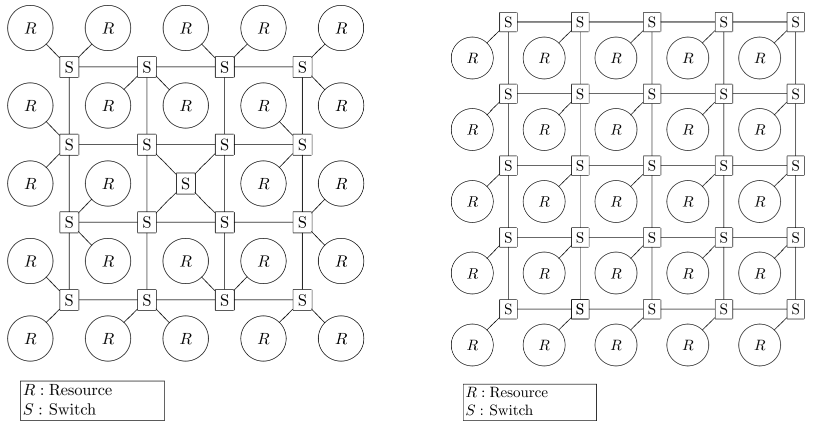
(a) General Hierarchical structure of HG

(b) Structure of proposed Square HG(SHG.)

Fig. 1.

level to the upper level graph. To facilitate physical implementation of interconnections, the graph in each level can have a regular shape e.g. square, which leads to *Square Hierarchical Graph* (SHG). The main focus in this paper is on SHG. The number and position of the border switches are flexible parameters of this architecture. A good choice of border switches in a SHG network would avoid diagonal links as shown in Fig.2(a). In the highest hierarchy, there is always a single powerful switch called *primary switch*. In HG the cores can be placed in different levels of the hierarchy according to their specific communication needs. For example the cores with higher traffic requirements such as central processing units or shared memories can be placed in higher levels to reduce latency and increase the overall performance of the chip.

To analyze performance parameters of HG we used a specific version of SHG. For N^2 general cores, such a SHG network uses $(N-1)^2$ switches with strict connection lines to make a symmetric graph with maximum performance and minimum latency. When N is odd, the central core is substituted with a high performance switch named primary to make the graph more symmetric. The network in Fig.1(b) which is a version of the SHG just mentioned consists of 17 switches, one powerful switch in the highest level, 4 ones in the second level and 12 switches in the third level. Four of the switches in the lowest level are not border switches and the rest are all border switches.



(a) Overview of SHG architecture (b) Overview of Mesh architecture

Fig. 2.

Simulations in section 4 demonstrate that, this type of SHG outperforms Mesh in many benchmarks while its number of switches is less than the equivalent Mesh (see Fig.2(b)). It is predicted that the complexity of switches in NoC systems would increase dramatically as the need for implementation of heavy weighted routing algorithms, large queues and queue management disciplines increases. Thus the fewer number of switches would reduce the chip area and overall cost. According to [8], switches with less than five inputs and output virtual channels have simple implementation. So HG switches are not more complex in comparison to common switches used in Mesh.

4 Simulation

Simulations were used to compare the new architecture with existing ones from the performance point of view. Our intention was to prove the suggested architecture has no or little performance penalty while reduces costs. Among the two other architectures , Mesh and Fat-Tree, only Mesh is compared to HG. This is because of the implementation issues of Fat-Tree, discussed in 4.1. In the section 5 the results are compared and different aspects of performance issues of the new architecture are discussed.

Simulations were done using two different traffic models: random traffic and local intensive traffic. In each simulation scenario we used a Mesh of 25 cores which requires 25 switches and a SHG with 24 cores in which 17 switches are involved. The two measured performance indicators were "drop probability" and "overall packet delay". Drop probability is the ratio of the number of dropped packets to the total number of generated packets. For the overall packet delay, end to end delay of all delivered packets was measured and averaged.

The underlying transmission protocol used was UDP. UDP offers no congestion control and reliability. While these two issues are not covered well in NoC researches, TCP is not a good choice because of its high overhead, non predictable delay and other implementation issues. The use of an application specific congestion control can increase the overall performance. There should also be methods to detect and recover delivery failures either in application layer or network layer. These issues are still open to research. Since we are just interested in the architectural performance evaluation, we can safely use UDP for all architectures.

The two traffic models discussed before differ in the choice of sender-receiver pairs. In the random traffic model, each node sends data to another random node. Since each node has no more than one traffic sink and source, this is equivalent to a uniform distribution of sink and source pairs among the cores. While this traffic model can indicate some performance features of any architecture, in practice the locality of traffic is important. For the local traffic model we chose source and sink pairs in a way that 75% of traffic is local. The rest 25% was distributed randomly among the remaining nodes.

The traffic source behavior is also of special importance, while the particular traffic behavior can greatly depend on the specific application, exponential traffic seems most suitable for the simulation purpose. All simulations were also repeated with randomized Constant Bit Rate (CBR) to cover a wider range of traffic behaviors in real applications.

4.1 Modeling Simulation Parameters

For simulation of any network architecture for NoC, we need system level modeling of different chip implementation parameters such as switch delays, link maximum bandwidth and delay of links.

Our modeling parameters are based on the ITRS 2004 data for 45 nm technology predicted to come to market in 2010 [1]. For modeling purposes we will use the top level global interconnection parameters. As predicted in [1], these interconnections will have RC delay about $143ps/mm$ to $189ps/mm$ (the exact amount of delay depends on scattering effects). Also the intrinsic delay of each NMOS is considered to be about $0.39ps$ to $0.98ps$, (the exact amount of delay depends on performance and power efficiency of the device). With these delay values for NMOS the delay of a NAND gate would be about $9.88ps$ to $24.8ps$. The number of maximum parallel lines that can be used to increase bandwidth of a link is also important. This depends on global links wiring pitch size which is predicted to be about $205nm$.

In Fig.2(a) there are 24 cores, one primary switch and 16 ordinary switches. If we assume the cores and primary switch to be about $2.5 \times 2.5mm^2$ and the ordinary switches about $0.5\times0.5mm^2$, in a chip area of $1.5\times1.5cm^2$, the length of the links between ordinary switches would be $2.5mm$ and the length of the links between cores or primary switch and ordinary switches is $0.1mm$. The overall length of the links is smaller than $10cm$, so using the given pitch size there could be more than 5000 parallel lines in each link. For simplification of modeling only 1024 parallel lines have been used. Using $\sim 200ps/mm$ delay for links, delay of a link between two switches will be about $500ps$ and the links from a switch to a core would have about $20ps$ delay. The switch implementation is based on [8]. The delay of this switch could be modeled with 5 series of NAND gates estimated about $50ps$. The delay of each *Resource Network Interface* (RNI) is about half of the delay of an ordinary switch and thus about $25ps$.

For a signal transmitted from a RNI and received by a switch, the overall delay consists of RNI delay, link delay and half of a switch delay which is totally about $70ps$. This means $550ps$ switch to switch delay. For calculating switch to link bandwidth, $550ns$ delay is considered leading to $2GHz$ bandwidth for each line. So 1024 parallel lines bundled in a link would have $2048GHz$. Using this method the bandwidth of the link between a switch and a core is $51.2GHz$.

In Mesh topology (Fig.2(b)) core to switch links and switch to switch links are the same as in SHG. Also RNI and Switches have approximately the same structure as in SHG. So the same model as SHG is used for Mesh.

In Fat-Tree the connections cannot be implemented on a single metal layer and its long link distances in real chips make the bandwidth of links very poor in comparison to the two other topologies, Mesh and HG.

4.2 Simulation Experiment

All simulations were done using the publicly available Network Simulator (NS-2) [9]. Both a traffic sink and a traffic generator were attached to each node. In other words each node could both send and receive data which is near to what is expected in real applications. Since the selected Mesh network had one core more then SHG, the central core of Mesh was not involved in simulations.

To reduce high variations in random traffic model we had to repeat each simulation and use the average result.

Since NS-2 is designed for computer networks, the bandwidth and delays proposed for a NoC system are not suitable for simulation. We scaled down all the estimated quantities by dividing all bandwidth values by a factor of 10240 and multiply delay values by a factor of 10240, so all simulation results are in terms of scaled quantities.

5 Results

As mentioned before the results of simulations are average packet delay and drop ratio for Mesh and SHG, using two traffic models and two traffic source behaviors. The results are presented here according to the traffic model. In the following graphs delay and drop probability are shown versus traffic load generated by a single traffic source.

5.1 Random Traffic Model

Although the random traffic model is not what is expected to happen in real applications it can provide a good measure for comparing the two architectures.

The use of random CBR traffic generators in the random traffic model for SHG and Mesh resulted in Fig.3 and Fig.4. The graph in Fig.3 shows average packet delivery delay versus traffic load of any traffic source and the graph in Fig.4 depicts packet drop ratio vs. traffic load. While in SHG, packet drop begins near $85Mb/s$ for a source, in Mesh packet drop begins from $90Mb/s$. No NoC

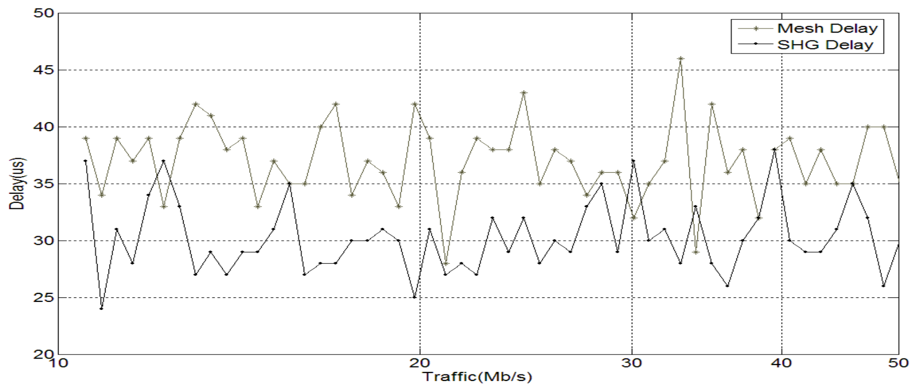


Fig. 3. Packet delay vs. traffic load of a random CBR source in a random traffic network (random traffic model)

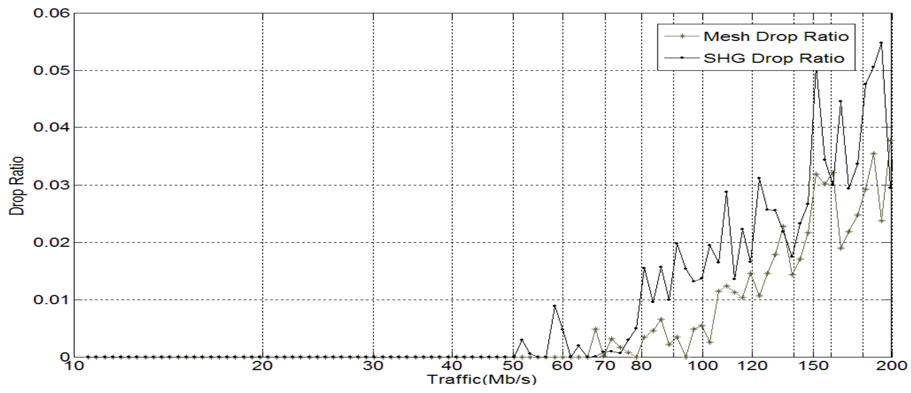


Fig. 4. Drop probability vs. traffic load of a random CBR source in a random traffic network (random traffic model)

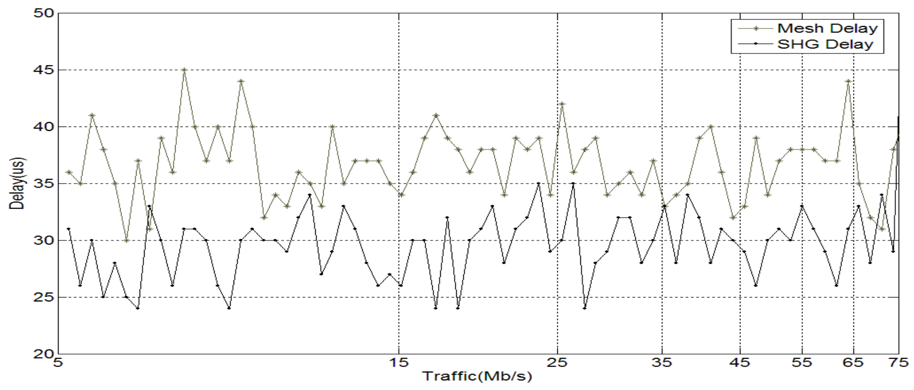


Fig. 5. Packet delay vs. traffic load of an exponential source in a random traffic network (random traffic model)

system can be used in traffic loads near these values but for comparison purposes we continued the simulations beyond these points.

Fig.3 shows that the overall packet delay in SHG is less than Mesh by a constant factor. On the other hand packet drop ratio is less in Mesh.

The exponential traffic source simulations lead to similar results in Fig.5 and Fig.6.

The exponential traffic source is expected to be more realistic as a traffic source. As the graphs show, the results for this type of traffic source have very little difference from the outcome of random CBR traffic source simulations.

The results of random traffic model indicate that SHG performs better when considering overall packet delay. On the other hand SHG is defeated by Mesh when considering drop. The point where considerable packet drop begins is the same in both topologies and after that the drop rate is higher for SHG. After the

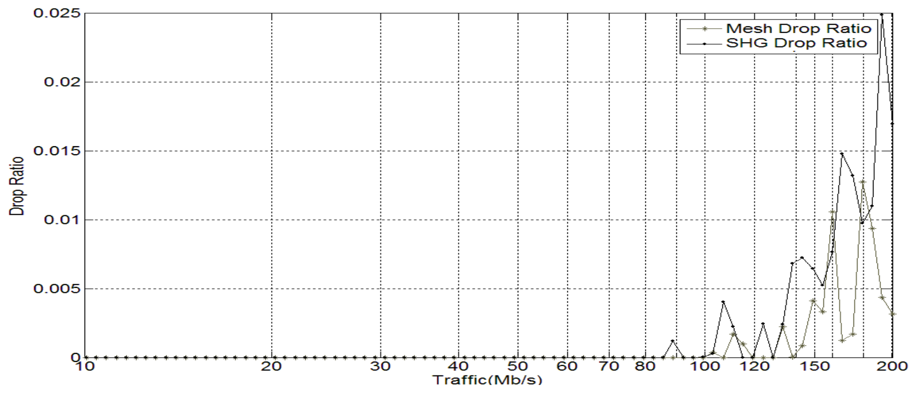


Fig. 6. Drop probability vs. traffic load of an exponential source in a random traffic network (random traffic model)

packet drop starts the network is considered unusable. Thus the most important section of all simulations is where the network is working without any drop.

5.2 Local Traffic Model

The local traffic model resembles real applications to a greater extent than random traffic. Thus the results of this section are of more interest and importance. Besides the more interesting nature of local traffic model, the absence of random variations makes the graphs more readable especially for the exponential simulation.

Fig.7 shows the results of simulating random CBR sources in SHG and Mesh in a local traffic model.

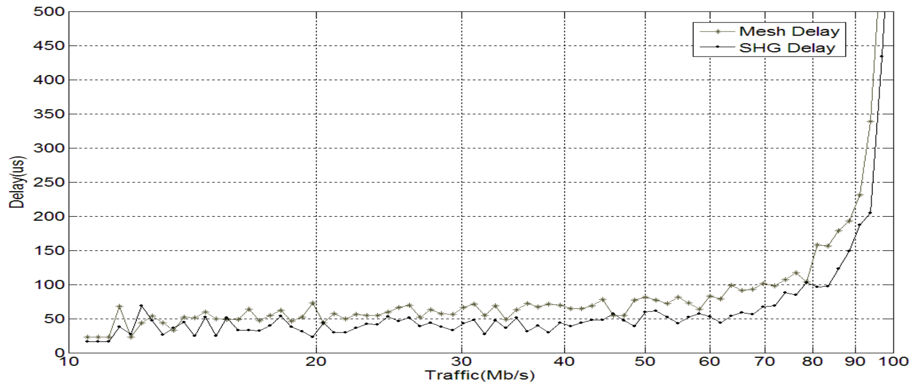


Fig. 7. Packet delay vs. traffic load of a random CBR source in a local traffic network (local traffic model)

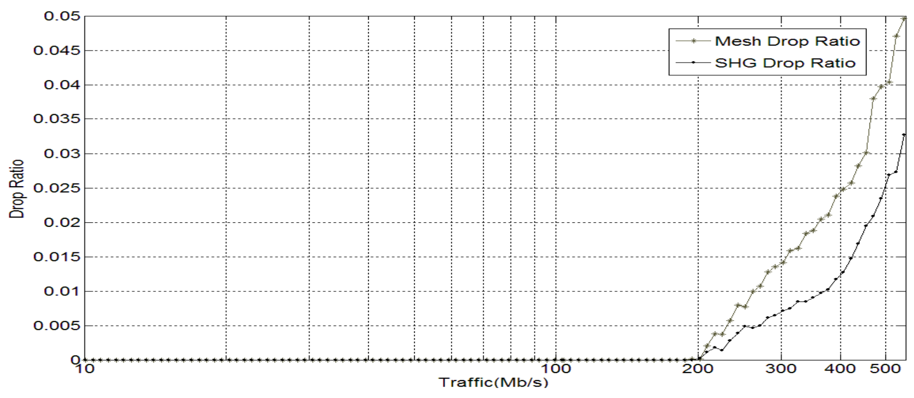


Fig. 8. Drop ratio vs. traffic load of exponential sources in a local traffic network (local traffic model)

Unlike the random traffic model, SHG shows better results for drop ratio in local traffic simulations. The reason is that in SHG there exist neighbor nodes with only one switch between but in Mesh the minimum number of switches between two resources is two. In other words SHG shows better locality than Mesh. Exactly the same reason stands for the bigger gap in delay of SHG and Mesh in higher loads.

In Fig.8 the drop ratio of SHG is much less than Mesh and the difference becomes more as the traffic load increases.

Fig.9 shows the delay of delivered packets for a local traffic model using exponential traffic sources.

These graphs shows that in lower traffic loads SHG performs better by a constant factor, and as the load increases the difference becomes more.

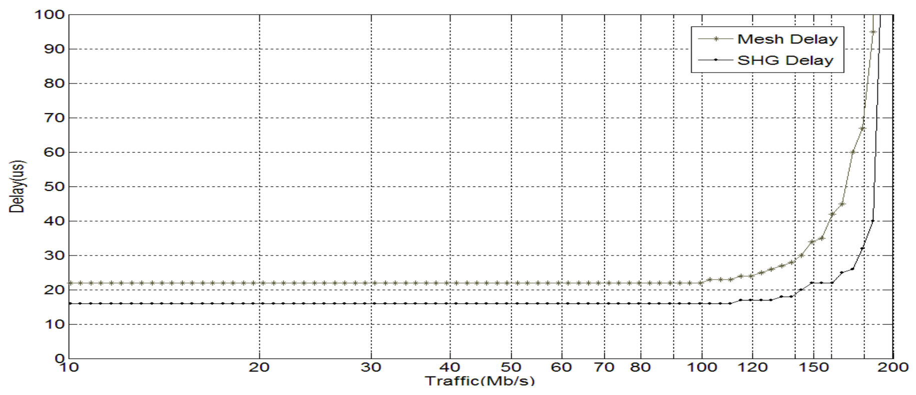


Fig. 9. Packet delay vs. traffic load of exponential sources in a local traffic network (random traffic model.)

6 Conclusion

This paper introduced HG as a flexible topology, specially designed for on chip networks. The flexibility of HG makes it suitable for various applications of SoC. HG provides a network infrastructure that can be easily altered to suite different applications. As an example SHG was introduced and benchmarked with a common used architecture (Mesh). The simulations proved that while SHG is easier to implement it outperforms Mesh in many aspects.

7 Future Work

The performance of HG depends on the appropriate choice of different topological parameters and many other factors such as switch buffer size, bandwidth and etc. The best approach is to find suitable values for these parameters for specific applications. These problems can be addressed in future work.

Furthermore, there are uncovered issues such as congestion and reliability control protocols, routing protocols and many other issues that are common among a very wide range of applications.

References

1. ITRS. International technology roadmap for semiconductors - 2004 edition, http://public.itrs.net/
2. L. Benini and G. De Micheli, Networks on Chip: A New SoC paradigm, IEEE Computer, January 2002(Vol. 35, No. 1), pp. 70-78.
3. JD Day, H Zimmermann, "The OSI reference model", Proceedings IEEE, 1334-1340, December 1983.
4. Shashi Kumar, et al., "A Network on Chip Architecture and Design Methodology", IEEE Computer Society Annual Symposium on VLSI, Pittsburgh, Pennsylvania, USA, April 2002.
5. Yi-Ran Sun, Shashi Kumar, Axel Jantsch, "Simulation and Evaluation for a Network on Chip Architecture Using Ns-2", Proceedings of 20th NORCHIP conference, Copenhagen, November 2002.
6. Vu-Duc Ngo and Hae-Wook Choi "On Chip Network: Topology design and evaluation using NS2".
7. Daniel Wiklund, Sumant Sathe, and Dake Liu, "Network on chip simulations for benchmarking", Proceedings of the International workshop on SoC for real-time applications, Banff, Canada, July 2004.
8. Nikolay Kavaldjiev, Gerard J. M. Smit, Pierre G. Jansen "A Virtual Channel Router for On-chip Networks", Proceedings of IEEE International SOC Conference, Santa ClaraSep, California, 2004
9. The network simulator - NS-2, http://www.isi.edu/nsnam/ns/