An Innovational Intermittent Algorithm in Networks-On-Chip (NOC)

Ahmad M. Shafiee, Mehrdad Montazeri, and Mahdi Nikdast

Abstract—Every day human life experiences new equipments more automatic and with more abilities. So the need for faster processors doesn't seem to finish. Despite new architectures and higher frequencies, a single processor is not adequate for many applications. Parallel processing and networks are previous solutions for this problem. The new solution to put a network of resources on a chip is called NOC (network on a chip). The more usual topology for NOC is mesh topology. There are several routing algorithms suitable for this topology such as XY, fully adaptive, etc. In this paper we have suggested a new algorithm named Intermittent X, Y (IX/Y). We have developed the new algorithm in simulation environment to compare delay and power consumption with elders' algorithms.

Keywords—Computer architecture, parallel computing, NOC, routing algorithm.

I. INTRODUCTION

HOWEVER an ENIAC computer had 18000 lamps and 30 Tons weight but we expect future computers have just 1000 lamp and 1.5 Tons of weight [1]. This sentence was a wish for 1950 peoples. The dream of making faster and smaller computers never ends with transistors; millions and billions of transistors are spending to make a chip for processors now. Foremost, the need of super computers is sensitive more and more, so this make human build systems-on-chips (SoC). A study on SoC reveals a poor scalability beyond a certain number of partners, for instance, the researches from the chip design have explored a very large scale of area for units such as boards, cooling systems, network cards and Etc. Also traditional computer networks which have already dealt are difficult to setting. These all findings has yielded a novel interconnect architecture called Networks-On-Chip (NOC). [2], [3].

Current algorithm on chip and system on chip design methodologies need billion transistors Area. The possible solutions must be searched from platform based design and computer system design, which rely on the reuse of components, architectures, applications and implementations. The essential issue is the trade-off between generality and performance. Generality provides reusability of hardware; operating systems and development practices, while performance (Delay, Cost, Power, Etc.), is achieved by using application specific structures. [4].

The main purpose of NOC is to provide communication infrastructure for the Resources. To survive in the billion transistor era a methodology should facilitate the developing of resources independently as standalone blocks and connect them as elements in the network. This scalable and configurable network is a flexible platform that can be adapted to the needs of different workloads, while maintaining the generality of application development methods and practices. Providing this intra chip communication based on switches will cause little or no overhead in the billion transistor generation. [2], [3], [4].

Even though many topologies fulfill the above requirements, a simple Mesh interconnection topology can be taken as a basic topology. NOC consists of resources and switches that are connected by channels. [5].

The Fig. 1 illustrates a 4*4 mesh topology and how resources are connected to switches. Each resource has one switch that has routes to four other switches. Obviously, these switches do the routings and transmitting data between resources.
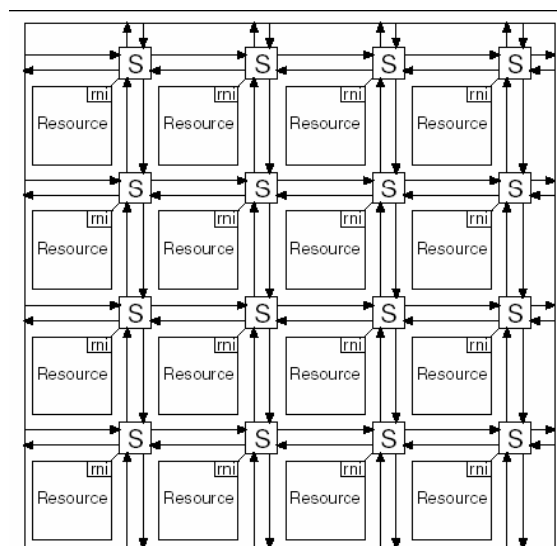
Ahamd M. Shafiee is now with the Computer department, Islamic Azad University (Najafabad Branch), Isfahan, Iran (fax: +983312291008; e-mail: Shafiee@ ce.sharif.edu).

Mehrdad Montazeri is with the Islamic Azad University (Najafabad Branch) Isfahan, Iran (e-mail: MMontazeri@iaun.ac.ir).

Mahdi Nikdast is with the Islamic Azad University (Najafabad Branch) Isfahan, Iran (e-mail: MNikdast@acm.org).

Fig. 1 A 4*4 simple Mesh

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:9, 2008

The paper is organized as follows. In section II, we review the existing adaptive routings. Section III, introduces our intermittent algorithm. In section IV, we clarify the simulation environment and the simulation results. Finally, we summarize our work in section V.

II. RELATED WORKS

Scholars have done significant researches to improve the routing efficiency of NOC. In [4], a First-Come-First-Served (FCFS) input selection algorithm is mentioned and the developed version of this routing algorithm named XY+FCFS is introduced. This improved algorithm combines the advantages of both deterministic and adaptive routing by choosing the appropriate routing algorithm under different traffic conditions.

In [8, 9], other approaches are proposed. There are several additional works that present adaptive and dynamic routing, like DyAd [5], and Odd-Even routing by Chiu [6]. These solutions do not perform offline optimizations at configuration time; instead, they dynamically change routing policies on the fly. The motivation of this paper is to investigate the routing techniques and use solutions to improve NOC efficiency and performance.

III. INTERMITTENT ROUTING ALGORITHM

Routing is an act of moving information from a source to destination following the shortest possible path available. Finding this path requires an algorithm which provides the routing method. In deterministic routing algorithms, transfer path is completely determined by the source and the destination address. All of the switches check the destination address of the packet and demand a link to send it to a neighbor.

XY routing algorithm first routes packets horizontally, towards their X coordinates, and then vertically towards their Y coordinate. XY is commonly used in NOCs thanks to its simplicity and inherent deadlock-freedom. However, it induces unbalanced link loads [7]. YX routing is the same as XY but first routes packets vertically towards Y direction and then horizontally towards X direction.

In this paper we intermittently use XY and YX algorithms to decide the next hop. That means one time the X coordinate will be checked and the packet will be passed through the horizontal path, unless the destination is in the same column. If so the packet will rout vertically. In the next routing the Y coordinate check will have higher priority and in the situation with same destination row, the X coordinate would be checked.

In Fig. 2 we show two packets have routed by IX/Y. Data would be transferred from node 0 to node 15; first packet's Boolean variable is set to 0 and is routed by XY, then next packet's Boolean variable is set to 1 and is routed by YX, again 3rd one is set to 0 and the routing method is XY, and so on. This continuous process from routers in network layer will help to have fewer collisions.
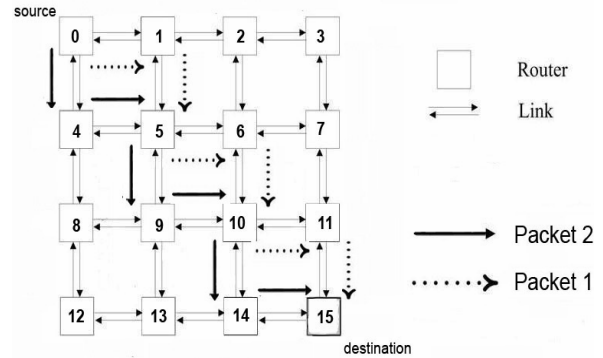

Fig. 2 Intermittent X, Y

IV. SIMULATION RESULTS

To check the delay and energy conception, we have used an open source program called NOXIM. We added our routing algorithm into its routing types and compared IX/Y results with other algorithms.

In this section we present the results. To have more reliable comparison, we ran the simulation and extracted each value for 5 times and made an average value.
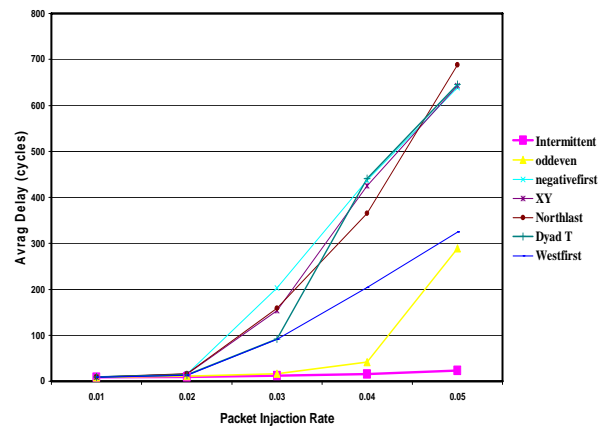

Fig. 3 Comparing average delay

Fig. 3 illustrates how delay in XY, Dyad T, Negativefirst, and Northlast algorithms will increase dramatically from 0.02 of injection rates onwards. The westfirst algorithm climbed slightly to reach 324 cycles in packet injection rate of 0.05. However, oddeven algorithm stabilized with the injection of 0.04 but it rocketed to 289 in 0.05 injection rate. It is clear that our innovational intermittent algorithm is roughly remained stable below 30 cycles at all. We also educated that this statistics is approximately the same in a 10*10 mesh-based and is an enduring algorithm in higher injection rates.

On the energy consumption facet, Fig. 4 gives that all algorithms had rose sharply and the highest results was for oddeven in comparison with the lowest Northlast and our Intermittent is closely following the Northlast.
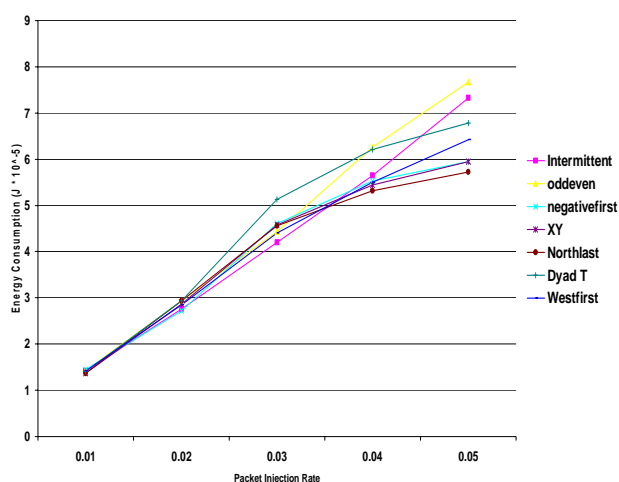
World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:9, 2008

Fig. 4 Comparing energy consumption

V. CONCLUSION

To sum up we experienced a novel algorithm for NOC that have a noticeable improvement in the amount of average delay. Also on the power consumption facet we could compete with the best algorithm.

Actually there is insignificant difference in low packet injection rates. It can be easily concluded that low packet injections makes less collisions, so delay may have no differences. Increasing injection rate reveals some distinction between IX/Y and the others. The most significant reason for this result is that intermittent algorithm can disperse the packets into the paths better than other algorithms.

REFERENCES

[1] Andrew Hamilton. Brains that click. *Popular mechanic, 91*(3): 162-167, 256, 258, march 1949.
[2] Luca Benini and Giovanni De Micheli, Networks on Chips: A New SoC Paradigm., *DATE, Design, Automation and Test in Europe, Paris, March 3 - 7, 2002.*
[3] P.Guerrier and A.Greiner .A generic architecture for on-chip packet switched interconnections, *In Proc. Of DATE 2000*, March 2000.
[4] Shashi Kumar, Axel Jantsch, Juha-Pekka Soininen, Martti Forsell. A Network on Chip Architecture and Design Methodology. Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI.02) 0-7695-1486-3/02.
[5] Jingcao Hu, Radu Marculescu, "DyAD - Smart Routing for Networks-on-Chip", DAC 2004.
[6] Chiu Ge-Ming. "The Odd-Even Turn Model for Adaptive Routing", IEEE Transactions on Parallel and Distributed Systems, Vol. 11. July 2000, pp. 729-738. 2000.
[7] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny. "Efficient Link Capacity and QoS Design for Network-on-Chip".In DATE 2006.
[8] E. Nilsson, M. Millberg, J. Oberg and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," DATE, Germany, 2003, pp. 1126-1127.
[9] J. Hu, R. Marculescu, "DyAD - Smart routing for networks-on-chip," *DAC*, USA, 2004, pp. 260-263.