A Distributed Adaptive Routing Algorithm for Regular Networks-on-Chip

Masoud Seyedmohammadzadeh

Abstract— In this paper, the problem of congestion in Networks-on-Chip (NoCs) is addressed. We employ a distributed approach by dividing NoC into several regions. Every region is controlled by a local agent. Each agent runs Dijkstra's shortest path computation procedure to find the best routing path and avoids highly congested paths. This leads to significant improvement in network latency and throughput. Simulation results which are based on Verilog implementation of the system shows that the proposed routing algorithm improves system performance significantly.

Index Terms— Networks on chip, Dynamic routing algorithm, Throughput, Latency, Dijkstra algorithm, Verilog, FPGA.

1 INTRODUCTION

Recent advances in design and fabrication of integrated circuits has pushed traditional bus based systems to the limit of their performance. Networks on Chip (NoC) are proposed as an alternative to bus based communication structures. Compared to traditional buses NoC is more predicatable, scalabile, and have higher bandwidth [1], [2], [3]. The specialized routers and the links that interconnect them are the building blocks of an NoC. Routers are connected to processing elements (PEs) and data/messages are exchanged between these PEs through NoC. Data on NoC can be transfered in two ways: By employing a deterministic or by employing dynamic routing algorithm.

In deterministic routing the routing path is pre-computed and is based on the transmitter and receivers locations (addresses). The main reason that makes deterministic routing attractive is its similicity and possibility of guaranteeing that deadlock and livelock will not happen in system. The main drawbacks of the deterministic routing algorithm are its inability to respond to network congestion and failure of network elements [4], [5]. On the other hand, dynamic routing algorithms can dynamically respond to changes in network traffic by computing the alternative paths in case some parts of the network are congested, and therefore improve system performance. Also if they are armed with fault tolerant mechanism, they can respond to interconnect failures and therefore recover system functionality even when some system elements are failed [6], [7]. The main drawbacks of the dynamic routing algorithms are area and power overheads caused by the extra hardware utilized to implement the fault detection mechanisms and the hardware required to compute new routing paths.

Adaptive routing in the context of NoC is a popular research topic and it has attracted a lot of researchers. [8] presents an adaptive routing algorithm that is based on a dynamic programming (DP) network and is capable of providing optimal path planning. A hybrid approach (DyAD routing algorithm) that switches between deterministic routing (in low traffic) and adaptive routing (when network is congested) is presented in [9]. The dynamic routing algorithm discussed in [10] uses information from all routers in the source-destination path to do the traffic routing. The source units utilize the information about network conditions to adjust the parameters that determine the path to the destination router. [11] presents a centralized monitoring system that can locate congested links and detour them. However, this method is not scalable to bigger networks where hundreds of PEs are integrated on a systems on chip (SoC). [12] discusses an adaptive routing algorithm that always tries to route packets to their destintion using a path that is the least congested as possible. This algorithm utilizes the cases of indecision occurrings when the routing function returns several admissible output channels. Authors in [13] proposed a dynamic routing scheme where intermediate routers make decisions locally based on the available bandwidth in each direction to the adjacent routers and also based on the distance between current and the target routers.

In this paper, we develop and implement a novel dynamic routing algorithm for NoC. The propsed algorithm is designed to address the congestion problem in NoC. The algorithm has a distributed structure, meaning that the routing descisions are made locally by local controllers. In this approach we divide NoC into several partitions. Each partition is controlled by a designated controller. Each local controller calculates the routing paths in its partition. The implementation cost of the proposed algorithm compared to previous work (like [11], [14]) is very low. This is because most of the previous works follow a centralized approach to implement their congestion aware routing strategy. Therefore, they need additional global (long) wires to transfer control signals to carry the information about the network traffic status. It is because of this big cost associate with their implementation that they face scalability problem when the size of network increases. The experimental results demonstrate how the proposed routing algorithm can considerably improve the network performance metrics: throughput and latency.

• *Masoud Seyedmohammadzadeh, received his Masters degree from Islamic Azad university, Dezful, Iran. He is currently a researcher at Rakhshan Dasht Nab, Urmia, Iran. Email: msm.zadeh@gmail.com*

2 Proposed Adaptive Routing

A regular homogeneous NoC is formed by routers and bidirectional links that interconnect them. As mentioned above we follow a distributed approach and employ local controllers (LCs) to compute routing paths. We divide NoC into several partitions and assign a local controller to each partition. Each local controller calculates routing paths in its designated partition. Because this method is a regional base, the cost associated with its implementation is minimal. Fig. 1 shows a 4x4 NoC devided into 4 partitions, and the local controllers designated to each region. Although in this figure the partitions have the same size, in practice they can have different sizes. In this figure each local controller is in charge of routing packets to routers within its designated partition and to the neighboring routers adjacent to its partition. For example in Fig.1, LC_1 is responsible for routing packets injected to the routers in its partition (routers {0; 1; 4; 5}) and the packets that arrive from adjacent routers {2; 6; 8; 9}. All tables and figures will be processed as images. You need to embed the images in the paper itself. Please don't send the images as separate files.

2.1 Shortest Path Computation Procedure

In order to be able to compute the shortest path for the packets traversing through NoC we should first construct and associate a directed graph to the NoC. To do so, we will consider the NoC routers as the nodes in the graph and the links as graph edges. The edge weights of the graph are computed by utilizing buffer occupancy. That is, the weight is calculated as the sum of the numbers of memory slots oocupied in the output buffers of the router that the link originates from. Fig.2 depicts an example of such a graph. The edge weights wij , i = 1, …, V, j = 1, …, V (where V is the total number of nodes in the graph) can be calculated as follows:

$$w_{ij} = \begin{cases} Used\ memory\ slots & \text{if } (v_i, v_j) \in E, \forall v_i, v_j \in V \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

The following equation describes the objective in the procedure of calculating the shortst path. The goal is to minimize the path cost:

$$Min: \quad \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} x_{ij} \times w_{ij} \quad (2)$$

where (v_i, v_j) is used or not as a part of the path. In the following, we explain the main steps in the proposed dynamic routing algorithm:

1) Initially (when we power up the system) we start with XY routing algorithm. The routing table of each router is initialized in advance to support XY routing. Therefore when the system is powered up the first packets are routed using XY routing algorithm.

2) When links become congested, each router informs its corresponding controller (using the dedicated wires) about the change in network condition (traffic).

3) After receiving the information about the traffic condition in their designated partition, local controllers calculate the new optimal and alternate routing paths for all of the packets by running the Dijkstra's shortest path algorithm.

4) Once the new routing paths are calculated and identified, the local controller will update the routing tables in the routers accordingly.

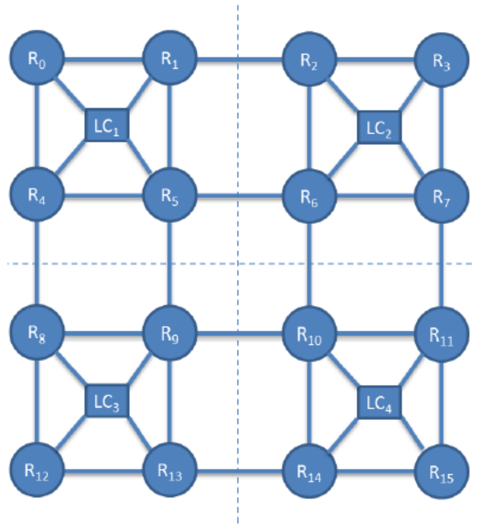

Fig. 1. A 4x4 regular NoC that is divided to four equally sized partitions, each partition is controlled by a local controller (LC).

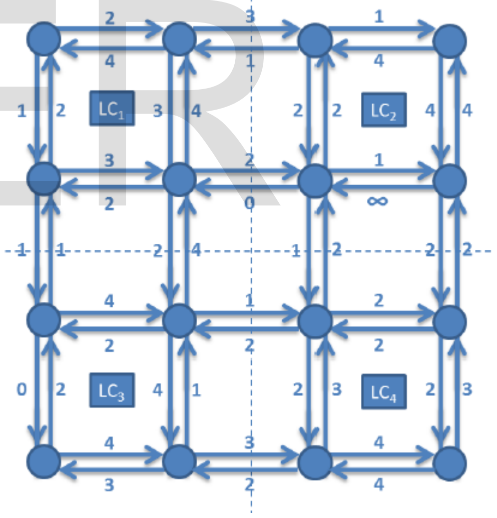

Fig. 2. Graph representtion of a 4x4 regular NoC.

3 Router Architecture

In order to implement the proposed routing algorithm, the router architecture should be modified. We design a new router to provide hardware support for our proposed routing algorithm. As mentioned above, we use output port buffers occupancies to calculate edge weights. Therefore we modify the output port buffers to provide the information about the port occupancy to the corresponding local controller (Fig.3).

Whenever there is an empty buffer in the output buffers, the arbiter in Fig.3 will be notified. Arbiter continuously monitors the output ports, and as soon as an empty buffer is available it will first look at input ports that are requesting access to that output port. Then it will choose and grant access to an input port that has the most occupied buffers. Since it is possible that one of the input ports would be dealling with high traffic and because of that it will keep winning the competition over the access to a specific output port and the input ports that have lower traffic would never win access to that output port, the arbiter sets a threshold for the number of time an input port can win access to a specific output port. Once that threshold is met it will give chance to the other ports. The output interface unit shown in Fig.3 manages the handshaking process with the adajacent router.

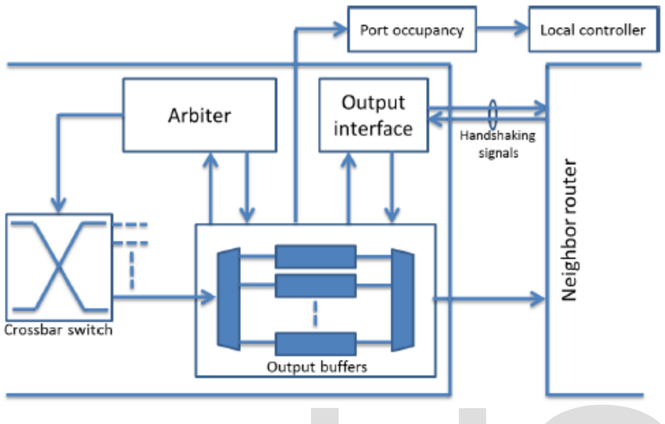

Fig. 3. Block digram of the output port of the router.

4 Experimental Results

In order to test and validate the performance of our proposed dynamic routing algorithm, we implemented and prototyprd a 4x4 NoC using Verilog. We used Xilinx ISE compiler [15] to simulate and synthesize our NoC structure. We compared our routing algorithm with DyXY [5] and the adaptice routing algorithm presented in [11]. Note that the hardware implementation of most of the dynamic routing algorithms is not available. Therefore, we restrict our experiments to comparisons with the aforementioned algorithms. Fig.4 shows the multimedia task graph we adopted from [16] and used in our simulations. We report our results for two multimedia benchmarks. The communication task graph (CTG) and optimized mapping of the first benchmark are from [16] and are shown in Fig.4. The injected traffic at all sources of the CTG is generated by local generators. The average number of injected packets at each source is proportional to the communication volume of each source-destination pair shown in Fig.4.a. The avarage latencies calculated utilizing the proposed routing algorithm, and DyXY [5], and the algorithm presented in [11] is demonstrated in Fig.5. This fiqure depicts how our proposed adaptive routing algorithm improves the network latency and throughput.

Table I depicts the improvement in network throughput and the hrdware cost associated with these algorithms compared to traditional XY routing algorithm.

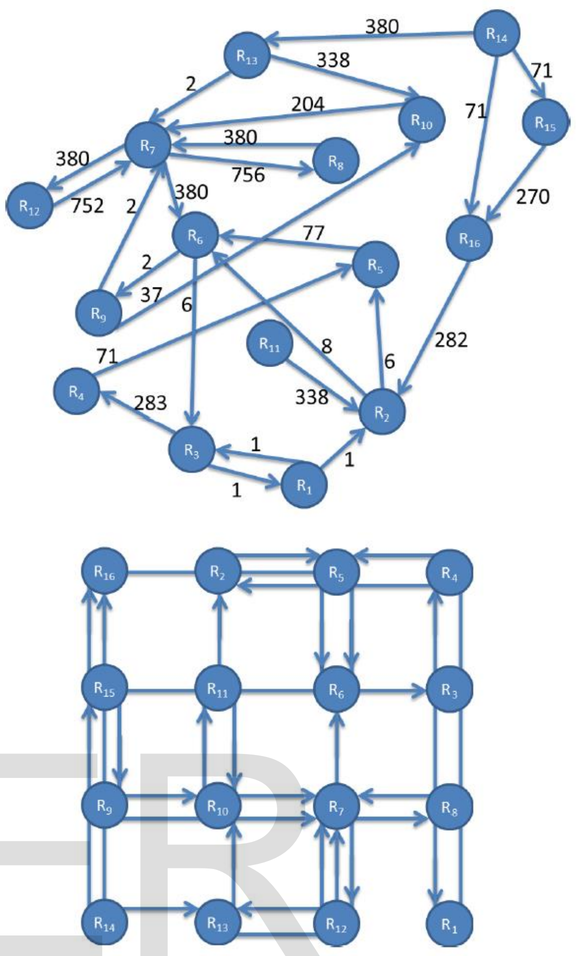

Fig. 4. Characteristic Task Graph (CTG) of the multimedia testebech and its optimized mapping on a regular 4x4 NoC.

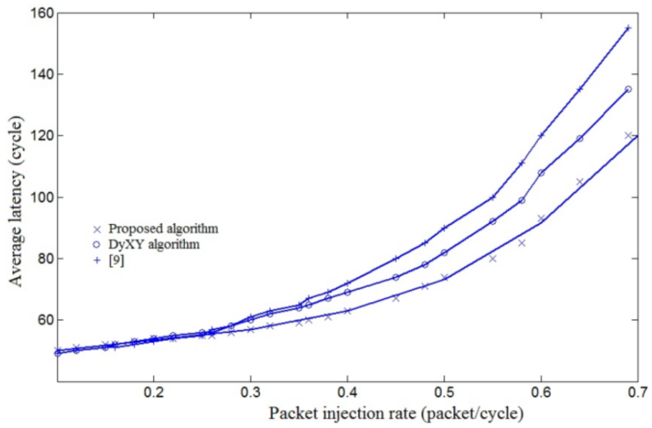

Fig. 5. Comparison of the average latency achieved by the proposed adaptive routing, DyXY [5] and the adaptive algorithm presented in [11] for the multimedia benchmark shown in Fig.4.

International Journal of Scientific & Engineering Research Volume 5, Issue 6, June-2014
ISSN 2229-5518

TABLE 1
COMPARISON AGAINST XY ROUTING ALGORITHM

Routing algorithm	*Area cost*	*Improvement in throughput*
Proposed algorithm	10 %	22%
DyXY algorithm	12 %	18%
[11]	15 %	17%

5 CONCLUSION

We proposed a new congestion-aware dynamic routing algorithm for networks on chips. We implemented our proposed algorithm by partinitioning the network into several partitions which are controlled by local control units. Each of these local control units run a shortest path computation algorithm and identifies the best routing path for the packets that enter their designated region in such a way that highly congested paths are avoided. This algorithm is periodically called and run; therefore it is capable of dynamically reacting to continuously changing traffic conditions. The area or ardware cost of the proposed algorithm is minimal compared to the available previous work. Experimental results are based on Verilog implementation and they depict that the proposed algorithm improves the network throughput and latency considerably better than DyXY adaptive algorithm, and the algorithm presented in [11].

REFERENCES

[1] H.S. Kia, C. Ababei, "Improving Fault Tolerance of Network-on-Chip Links via Minimal Redundancy and Reconfiguration," International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2011.
[2] G.D. Micheli, and L. Benini, Networks on Chips: Technology and Tools, Morgan Kaufmann, 2006.
[3] H.S. Kia, C. Ababei, "Efficient high-speed current-mode links for network-on-chip performance optimization", IEEE International SOC Conference (SOCC), 2012.
[4] H.S. Kia, C. Ababei, "A new fault-tolerant and congestion-aware adaptive routing algorithm for regular Networks-on-Chip," IEEE Congress on Evolutionary Computation (CEC), 2011.
[5] M. Li, Q.A. Zeng, and W.B. Jone, "DyXY: a proximity congestion aware deadlock-free dynamic routing method for network on chip," ACM/IEEE Design Automation Conference (DAC), 2006.
[6] R. Marculescu, "Networks-on-chip: the quest for on-chip fault-tolerant communication," IEEE Computer Society Annual Symposium on VLSI, 2003.
[7] M. Yang, T. Li, Y. Jiang, and Y. Yang, "Fault tolerant routing schemes in RDT(2,2,1)/a-based interconnection for networks on chip designs," International Symposium on Parallel Architectures, Algorithms and Networks, 2005.
[8] T. Mak, P.Y.K. Cheung, W. Luk, and K.P. Lam, "A DP-network for optimal dynamic routing in Network-on-Chip," ACM/IEEE International Conference on Hardware Software Codesign, 2009.
[9] J. Hu and R. Marculescu, "DyAD: smart routing for networks-on-chip," ACM/IEEE Design Automation Conference (DAC), 2004.
[10] L. Tedesco, F. Clermidy, and F. Moraes, "A path-load based adaptive routing algorithm for Networks-on-Chip," ACM Annual Symposium on Integrated Circuits and System Design, 2009.
[11] F. Ge, N. Wu, and Y. Wan, "A network monitor based dynamic routing scheme for network on chip," IEEE Asia Pacific Conference on Microelectronics and Electronics, 2009.
[12] G. Ascia, V. Catania, M. Palesi, and D. Patti, "Implementation and analysis of a new selection strategy for adaptive routing in networks on chip," IEEE Trans. on Computers, vol. 57, no. 6, pp. 809-820, 2008. doi: 10.1109/TC.2008.38. (IEEE Transactions)
[13] M.A. Al Faruque, T. Ebi, and J. Henkel, "Run-time adaptive on chip communication scheme," ACM/IEEE International Conference on Computer Aided-Design (ICCAD), 2007.
[14] M. Ali, M. Welzl, and S. Hellebrand, "A Dynamic Routing Mechanism for Network on chip," NORCHIP Conference, 2005.
[15] Xilinx ISE Tools, http://www.xilinx.com.
[16] M. Lai, L. Gao, N. Xiao, and Z. Wang, "An accurate and efficient performance analysis approach based on queuing model for Network on Chip," ACM/IEEE International Conference on Computer-Aided Design (ICCAD), 2009.