A Fault Resilient Routing Algorithm for Sparsely Connected 3D NoCs

Masoumeh Ebrahimi[1,2], Ronak Salamat[3], Nader Bagherzadeh[3], Masoud Daneshtalab[1,2],
[1]Department of Information Technology, University of Turku, Finland
[2]Department of Electronic Systems, KTH Royal Institute of Technology, Sweden
[3]Department of Electrical Engineering and Computer Science, University of California, Irvine, US
masebr@utu.fi; salamat@uci.edu; nader@uci.edu; masdan@utu.fi;

***Abstract*— 3D NoC has its outstanding advantages over 2D design such as the smaller footprint and the shorter global interconnects. However, vertical interconnects are expensive and prone to faults. It implies that 3D NoCs might be sparsely connected. Such non-fully connected 3D NoCs require proper routing algorithms to support the limited number of TSVs in the network where traditional algorithms such as dimension-order routing, XYZ, is not applicable anymore. In this paper, we propose a routing algorithm for the sparsely connected 3D NoCs with one, two and one virtual channels along the X, Y and Z dimensions, respectively. This algorithm is fault-resilient, meaning that it is functioning correctly for as long as there is at least one TSV in the right-most column of the network.**

Keywords-component: *Fault Tolerance;* Sparsely Connected *Networks-on-Chip; 3D Design; Routing Algorithm*

I. INTRODUCTION

The emergence of multicores in embedded systems requires a reliable, scalable and efficient communication infrastructure. The bus-based architectures have advantages in terms of simplicity and ease of implementation, but they suffer from fundamental limitations such as scalability and bandwidth. To overcome these limitations, a new communication infrastructure is needed for multicore chips. Network-on-Chip (NoC) has been introduced as a promising infrastructure which is based on the packet switching approach. NoC can tackle the scalability challenges faced by the traditional bus architectures.

By integrating hundred cores on a single chip, two dimensional NoC (2D NoC) faces new challenges such as large floorplan, long global wires, and increased latency. This has led technology (3D) toward three dimensional designs. 3D integration offers a drastic increase in transistor density by vertically stacking multiple dies with a dense and high-speed die-to-die interconnection [1][2]. Therefore, 3D integration results in considerable reduction in the length and the number of long global wires which are dominant obstacles for delay and power consumption. Moreover, 3D integration will improve heterogeneity capabilities in 3D ICs. However, there are some challenges regarding TSVs. TSV interconnect pitch imposes a large area overhead while several extra and costly manufacturing steps will be involved when the TSV technology is used for fabricating 3D ICs. In addition, the risk of defects grows as the number of TSVs increases, resulting in the yield reduction. The yield exponentially decreases when the number of TSVs goes beyond a certain value.

In order to take advantage of reduced interconnection latency as well as the heterogeneous integration offered by 3D IC and to address the scalability and bandwidth bottleneck offered by Network-on-Chip, 3D-NoC has emerged with limited TSV consideration. Sparsely connected 3D-NoCs introduce a great tradeoff between the benefit of high-speed and short-length vertical interconnections and the imposed cost and degraded reliability of using TSVs.

The main problem in such vertically partially connected 3D-NoC is the packet routing strategy which determines a path between each pair of source and destination as the usual simple routing algorithms are not applicable. Routing algorithms have to be deadlock-free. Deadlock happens when two or more packets are waiting permanently for each other to release a shared channel in a circular dependency. Designing a deadlock-free routing algorithm has been always a challenge even in the 2D-NoC involving only one plain (i.e. XY). In the 3D-NoC, the situation becomes even more difficult due to the possibility of forming a cycle within and between three plains (i.e. XY, XZ, and YZ). Designing such an algorithm for sparsely connected 3D-NoCs is more challenging and the current start-of-the-art is still lacking a viable solution.

This motivated us to develop an efficient and promising routing algorithm for sparsely connected 3D-NoCs. This algorithm is able to work with only one TSV at the east-most column while the performance can be improved by increasing the number of TSVs. The proposed algorithm is extremely light-weight. That is, it only requires one virtual channel along the Y dimension. This algorithm provides adaptivity to deliver packets, preferably using the shortest paths.

The remainder of this paper is organized as follows: Related work is given in Section II. The proposed routing algorithm is discussed in Section III along with proving its deadlock freeness. Experimental results are presented in Section IV and finally Section V concludes the paper.

II. RELATED WORK

There are many fault-tolerant approaches presented for 2D NoCs. These methods are able to tolerate faults in links [3][4], routers [5][6], or both [7].

In [3], the authors take advantage of a routing table at each router and an offline process to fill out the tables. A recursive method is used to fill out the tables. This method is based on a deterministic routing algorithm, degrading the performance of

NoCs. It is suitable for tolerating a large number of faulty links in the network. MD [4] is able to tolerate faulty links while providing adaptivity and routing packets through minimal paths. HiPFaR [5] and MiCoF [6] target tolerating faulty routers by avoiding non-minimal paths as along as possible. MiCoF takes advantage of bypassing links to avoid turning around the faulty area. The presented routing algorithm in [7] is able to tolerate both faulty links and routers.

In addition to the proposed approaches in 2D NoCs, several works have been done in the 3D domain. A fully adaptive routing algorithm with congestion consideration is presented in [8]. DyXYZ works on 3D fully connected meshes and it is proven to be deadlock-free by using 4, 4, and 2 virtual channels along the X, Y and Z dimensions, respectively. DyXYZ is not a fault-tolerant algorithm and it is based on homogenous 3D-NoCs.

Limited bandwidth in the vertical dimension has been considered in [9] which is a traffic-distributing routing algorithm for the 3D mesh network. In this algorithm, the congestion information of the neighboring nodes and the distance from the current node to the destination node has been considered in the routing decision. Different weights are assigned to vertical and horizontal directions. This algorithm allows using a non-minimal path adaptive routing algorithm to distribute traffic load over the network. This algorithm does not also deal with faults and its main focus is on congestion avoidance. In addition, this algorithm can be applied on non-fully connected 3D-NoCs.

Another fully adaptive routing algorithm which is called 3D-FAR is presented in [10]. This algorithm uses two, two and four virtual channels along the X, Y and Z dimensions respectively. In this algorithm, the network is divided into four disjoint networks and packets can use any shortest paths between the source and destination nodes. Non-minimal routes are used in the case of faults. Although this algorithm provides packet adaptivity and is fault-tolerant but it is designed for homogenous networks and cannot work in sparsely connected 3D-NoCs.

There are few works in literature suggesting fault-tolerant routing algorithm for sparsely connected 3D-NoCs, similar to the proposed algorithm in this work. Elevator-first [11] is the most relevant approach to our work which is a distributed routing algorithm for vertically partially connected 3D-NoC. To prove deadlock-freedom, two virtual channels per physical link in X and Y dimensions are used while there is no additional virtual channel in the Z dimension. The network is divided into two virtual networks Z+ and Z-. The former is for ascending packets and the latter is for descending ones. Every router statically knows the location of at least two vertical links in both ascending and descending directions. Each packet that wants to ascend or descend must know the location of the elevator and must go toward the elevator in the first step.

A modification on the elevator-first algorithm has been made in Redelf [12] which requires no virtual channels to ensure deadlock-freedom. In Redelf, certain rules are applied for choosing an elevator. To make distinguishable differences between elevator-first and Redelf, it is necessary to mention that in the elevator-first routing algorithm, there is no limitation on choosing an elevator when a packet travels between layers. However, it costs at the use of two separate virtual channels to ensure deadlock-freedom. Redelf on the other hand omits using virtual channels, but in order to guarantee deadlock-freedom, certain rules are applied which are limitative. Both of the routing algorithms are deterministic which are not able to distribute packets in congested networks.

III. THE PROPOSED ROUTING ALGORITHM

The suggested routing algorithm is proposed for vertically partially connected 3D-NoCs in which each node does not have a vertical link in order to deliver a packet to the destination layer. Therefore, conventional routing algorithms, such as XYZ, are not applicable anymore as in the sparsely connected 3D-NoCs the topology is not fully connected. The solution is that every router is statically informed about the location of the available vertical links. This information is stored locally at the router registers. The vertical links are considered as pillars. That is, the TSV in the first layer connects to all the layers. In vertically partially connected 3D-NoCs, in order to deliver a packet to the destination layer, the packet should be first sent toward the vertical link or elevator, next delivered to the destination layer and finally it will be routed toward the destination.

The routing algorithm requires two virtual channels along the Y dimension while there is no need to have any further virtual channel along the X and Z dimensions. In the proposed algorithm, the network is partitioned into two disjoint sets including different channels as: Set1 (X^+, $Y0^*$, Z^+), Set2 (X^-, $Y1^*$, Z^-) where "+", "-" represent channels along the positive and negative directions, respectively, while "*" stands for both positive and negative directions (bidirectional channels) as it is shown in Figure 1.

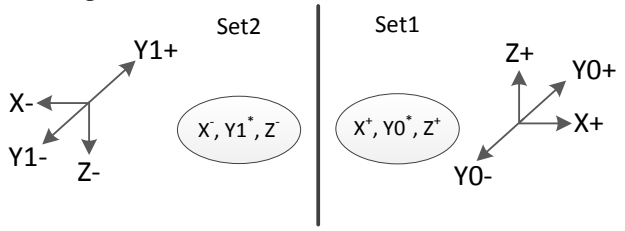

Figure 1. Determining two sets covering different channels

Packets in Set1 have flexibility to move toward the East direction (X^+), northward using the virtual channel number zero ($Y0^+$), southward using the virtual channel number zero ($Y0^-$), or upward (Z^+). Similarly, valid movements in Set2 are as follows: moving toward West (X^-), northward using the virtual channel number one ($Y1^+$), southward using the virtual channel number one ($Y1^-$), or moving downward (Z^-). As it can be obtained, packets in each set can switch between the directions dynamically and not necessarily following the dimension ordered routing.

The basic idea of this routing algorithm is that packets are allowed to use any channels either in Set1 or Set2 or move from Set1 to Set2 and then use any channels of Set2 (no transfer from Set2 to Set1 is allowed). Thereby, if any Eastward movement is needed it should be taken using the channels of Set1 before using any channels of Set2. At the

worst case, packets should reach the east-most column with the flexibility to take Y0* and then deliver to the desired layer and finally to the destination node. In other words, having at least one TSV in the east-most column guarantees delivery of packets between each pair of source and destination nodes.

A 4*3*2 network is shown in Figure 2 where four bidirectional TSVs are used in the network to connect the node 0 to the 12 (0-12); the node 8 to 20 (8-20); the node 10 to 22 (10-22); and the node 7 to 19 (7-19). An example is shown in Figure 2 where the source node 17 sends a packet to the destination node 7. In this case two elevators become eligible to deliver the packet to the upper layer as elevator 10-22 and 7-19. The elevator 7-19 is a better choice as it leads to a shorter path. In both cases, all channels between the source and destination belong to Set1 and no channel from Set2 is needed. In the example of Figure 3, the packet is sent from the source node 17 to the destination 1. Again both elevators (i.e. 10-22 and 7-19) can be taken to transmit the packet to the destination layer. However, as soon as the packet makes a movement toward the west direction, Set2 is utilized as it is not covered by Set1.

THE PROPOSED ALGORITHM IS DEADLOCK-FREE:

Generally, a cycle can be formed if packets are able to take both positive and negative directions along at least two dimensions [10]. As an example, to form a cycle in the XY plane, it is necessary to take the X+, X-, Y+ and Y- directions. The same trend is true for XZ and YZ as well. No U-turn (180-degree turn) is allowed in the algorithm. As can be obtained from the set definition, only the Y dimension is completed in each of the two sets. Thereby, there is no possibility of forming a cycle in each set. To prove that the network is deadlock-free between sets, it is enough to show that the two sets are disjoint from each other. A pairwise comparison between the two sets reveals that these two sets are different in the X and Z direction and the virtual channel number along Y. That is, Set1 only covers the positive direction of X and Z while Set2 covers the negative directions. The two sets are disjoint in virtual channel number along Y. Since no transfer from Set2 to Set1 is allowed, a cycle can never be formed. Therefore, moving toward X+ and Z+ will not be made after moving toward X- and Z- and deadlock-freedom is proved.

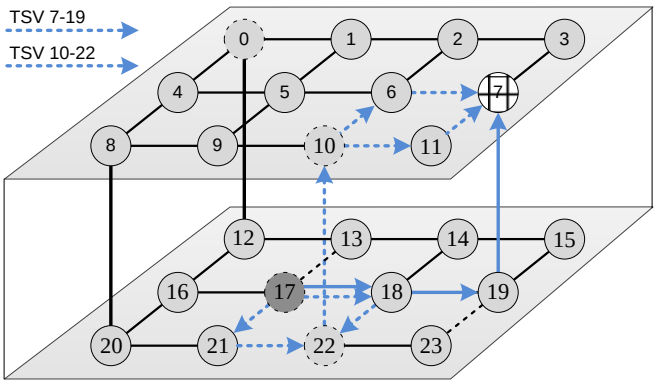

Figure 2. An example of the proposed algorithm with the source 17 and destination 7

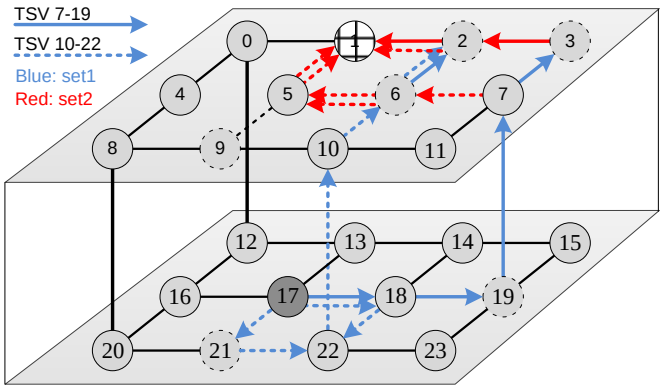

Figure 3. An example of the proposed algorithm with the source 17 and destination 1

IV. RESULTS AND DISCUSSIONS

In order to evaluate the efficiency of the proposed routing algorithms, AccessNoxim simulator is used [13]. AccessNoxim is a co-simulation platform for 3D-NoC systems that combines the network model, power model and thermal model. AccessNoxim has integrated Noxim (i.e. a cycle-accurate SystemC NoC simulator) and HotSpot (i.e. providing the architecture-level thermal model).

A. Reliability Measurement

The elevator-first routing algorithm utilizes two virtual channels along the X and Y dimension and one virtual channel along the Z dimension. This algorithm improves the performance but cannot directly apply to tolerate faults. In this method, when a router issues a packet whose destination is on a different tier, it adds a new header containing the elevator coordination to the original packet. Then, the routing algorithm routes the packet toward the elevator. Since the algorithm is deterministic, when the elevator is faulty, there is no other way to deliver the packet to the destination.

On the other hand, the proposed routing algorithm uses one less virtual channel than the elevator-first algorithm, i.e. one, two, and one virtual channel(s) along the X, Y, and Z dimensions, respectively. The proposed algorithm is fault-tolerant as long as there is at least one healthy elevator at the east-most column. Therefore, in the case of a faulty elevator, the router tries to find another elevator or even in the worst case it will use the east-most elevator for delivering the packet to the destination. In addition the algorithm is adaptive which improves the reliability by providing multiple paths to route packets.

B. Performance Measurement

Experimental results are measured on a 4*4*4 network. All the routers have 8-flit FIFOs and the packet size is 8 flits. Some TSVs are removed in the fully connected 3D mesh in order to make a partially connected 3D-NoC. Moreover, one virtual channel is added along the Y dimension. The elevator-first algorithm is implemented as a baseline method. Latency is defined as the number of cycles between the initiation of a message and the time when the packet is completely delivered to the destination. Experimental results are investigated for

random and shuffle traffic. The TSVs are pillars and they are located at nodes 0, 2, 7, 8 and 10.

In the random traffic pattern, each core generates data packets and sends them to a random destination. For the packet injection rate lower than 0.025, the proposed routing mechanism works nearly the same as the elevator-first routing algorithm. However, the elevator-first algorithm saturates in higher injection rates mainly because of utilizing an extra virtual channel.

In the shuffle traffic pattern, the third and fourth layers sends packets to the destination at the first and second layers while the second and half of the sources at the first layer deliver their packets to the destinations in the third and fourth layers. Under this traffic, the proposed algorithm works better than the elevator-first algorithm. This is due to the fact that the proposed algorithm can achieve a better distribution of packets because of its adaptivity characteristics.

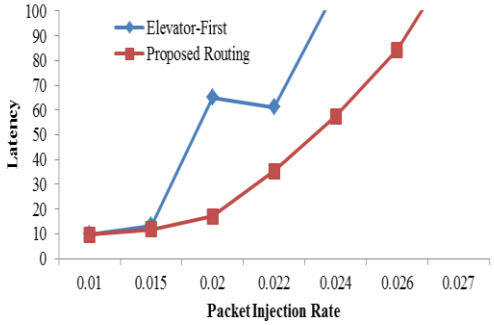

Figure 4 Performance under Shuffle traffic

V. CONCLUSION

3D NoC offers a better scalability, lower area, and shorter overall distance between the cores in comparison with the 2D NoC design. Beside these advantages, the main challenge in the 3D design is the imposed cost and manufacturing efforts of using TSVs in addition to the large area footprint required for them. A promising solution for this problem is to use limited number of TSVs in the network while taking advantage of the 3D designs. In this paper, we proposed a routing algorithm which is suitable for a sparsely connected 3D heterogeneous design with a limited number of TSVs. This algorithm is reliable and works as long as there is at least one TSV in the east-most column.

Acknowledgments

This work was supported by VINNOVA-MarieCurie within the CUBRIC and ERoT projects and Academy of Finland.

REFERENCES

[1] M. Palesi and M. Daneshtalab (Eds.), "Routing Algorithms in Networks-on-Chip," *Springer* 2014.

[2] W.R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer and P.D. Franzon, "Demystifying 3D ICs: The Pros and Cons of Going Vertical," *IEEE Desisn and Test*, vol 22, no. 6, pp. 498-510, 2005.

[3] D. Fick. A. DeOrio, G. Chen, v. Bertacco, D. Sylverster, D.Blaauw, "A highly resilient routing algorithm for fault-tolerant NoCs", *in Proc. of Design and Automation and Test in Europe (DATE)*, pp. 21-26, 2009.

[4] M. Ebrahimi, M. Daneshtalab, J. Plosila, F. Mehdipour, "MD: Minimal path-based Fault-Tolerant Routing in On-Chip Networks", *in Proceedings of 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 35-40, Jan 2013, Japan.

[5] M. Ebrahimi, M. Daneshtalab, J. Plosila, "High Performance Fault-Tolerant Routing Algorithm for NoC-based Many-Core Systems", *in Proc. of Parallel, Distributed, and Network-Based Processing (PDP)*, pp. 462-469, 2013.

[6] M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen, "Minimal-Path Fault-Tolerant Approach Using Connection-Retaining Structure in Networks-on-Chip", *in Proceedings of 7th ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, pp. 1-8, April 2013, US.

[7] M. Ebrahimi, M. Daneshtalab, "A Light-weight fault-tolerant routing algorithm tolerating faulty links and routers," *Journal of Computing*, 2014.

[8] M. Ebrahimi, X. Chang, M. Daneshtalab, J. Plosila, P. Liljeberg and H, Tenhunen, "DyXYZ: Fully Adaptive Routing Algorithm for 3D-NoCs," *International Conference on Parallel, Distributed and Network-Based Processing*, pp. 499-503, 2013.

[9] M. Zhu, J. Lee and K. Choi, "An Adaptive Routing Algorithm for 3D Mesh NOC with Limited Vertical Bandwidth," *International Conference on VLSI and System-on-Chip*, pp. 18-23, 2012.

[10] M. Ebrahimi, M. Daneshtalab, P. Liljeberg and H. Tenhunen, "Fault-Tolerant Method with Distributed Monitoring and Management Technique for 3D Stacked Meshes," *International Symposium on Computer Architecture and Digital Systems*, pp. 93-98, 2013.

[11] F. Dubois, A. Sheibanyrad, F. Petrot and M. Bahmani, "Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCS" *IEEE Transaction on Computers*, pp. 609-615, 2013.

[12] J. Lee and K. Choi,"A Deadlock-Free Routing Algorithm Requiring No Virtual Channel on 3D-NoCs with Partial Vertical Connections," *International Symposium on Networks on Chip*, pp. 1-2, 2013.

[13] http://access.ee.ntu.edu.tw/noxim/index.html.