An Adaptive, Low Restrictive and Fault Resilient Routing Algorithm for 3D Network-on-Chip

Ronak Salamat
Department of Electrical Engineering and Computer Science
University of California, Irvine
salamatr@uci.edu

Masoumeh Ebrahimi
KTH Royal Institute of Technology
and
University of Turku, Finland
masebr@utu.fi; mebr@kth.se

Nader Bagherzadeh
Department of Electrical Engineering and Computer Science
University of California, Irvine
nader@uci.edu

***Abstract*— The cost and reliability issues of TSVs move 3D-NoCs toward heterogonous designs with limited number of TSVs. However, designing a deadlock-free routing algorithm for such heterogonous architectures is extremely challenging due to the increased possibilities of forming cycles between and within layers for 3D designs. In this paper, we target designing a routing algorithm for heterogeneous 3D-NoCs with the capability of working under the technical limit in which there is just one TSV in the network. This algorithm is light-weight and provides adaptivity by using only one virtual channel along the Y dimension.**

I. INTRODUCTION

Three-dimensional (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]. 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.

Among the vertical interconnection technologies, Through-Silicon-Via (TSV) is the most promising solution since it has the greatest vertical interconnect density and exploits an extremely small inter-wafer distance. However, two architectural level design issues appear when TSVs are applied. First, a large area overhead will be imposed because of the TSV interconnect pitch according to the ITRS roadmap [3]. Second, several extra and costly manufacturing steps will be involved when the TSV technology is used for fabricating 3D ICs. Besides, the risk of defects will increase as the number of TSVs increases which results in yield reduction.

In order to take advantage of reduced interconnection latency offered by 3D ICs and to address the scalability and bandwidth bottleneck offered by network-on-chip, 3D-NoC has emerged with limited TSV consideration. Two types of routers are considered in these structures: 2D routers and 3D routers.

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 such as XYZ are not applicable. Designing a deadlock-free routing algorithm in 3D-NoC is really challenging due to the possibility of forming a cycle within and between three plains (i.e. XY, XZ, and YZ) and the current state-of-the-art is still lacking a viable solution.

This motivated us to develop an efficient and promising routing algorithm for partially connected 3D-NoCs, called East Then West (ETW). This algorithm is able to work with the minimum requirement of one TSV at the eastmost column while the performance can be improved by increasing the number of TSVs. The ETW 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 elaborated in Section II. Section III discusses the proposed deadlock-free routing algorithm. Results are presented in Section IV and finally Section V concludes the paper.

II. RELATED WORK

There are several works in literature targeting faulty routers and links in 2D-NoCs [4][5]. However, there are few works concerning routing algorithms in 3D-NoC especially with a limited number of vertical links. A fully adaptive routing algorithm without any fault consideration is presented in [6] for 3D fully connected NoCs. 4NP-First [7] introduces a fault-tolerant routing algorithm. In this routing algorithm, two redundant packets are transmitted for each destination one using the 4N-First turn model and the other using 4P-First when the fault rate is above a threshold value

Another fully adaptive routing algorithm which is called 3D-FAR is presented in [8]. 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 path between the source and destination nodes. Non-minimal routes are used in the case of faults. Elevator-first [9] 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. A modification on the elevator-first algorithm has been made in Redelf [10] 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 traverses 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.

In comparison with the elevator-first algorithm, the ETW routing algorithm in this paper uses one less virtual channel. Elevator-first is a deterministic method while we offer adaptiveness with no special limitation on moving toward X, Y or Z dimension. In ETW, during each source to destination transmission, a group of elevators is eligible and elevator selection can be made among them. Choosing an elevator is done at each router as the packet moves towards the destination. So, the ETW method has a fault-tolerant capability. In the elevator-first algorithm, however, a fixed elevator is assigned to a packet so that if the elevator is faulty, the packet is blocked. In the elevator-first algorithm, a new header is added to the packet containing the address of the elevator which adds both hardware and timing overhead. There is no such a header update in ETW.

III. DEADLOCK-FREE 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. 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. ETW 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 ETW 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.

Packets in Set1 have flexibility to move toward the East direction (X^+), Northward or Southward using the virtual channel number zero ($Y0^*$), or upward (Z^+). Similarly, valid movements in Set2 are as follows: moving toward West (X^-), moving Northward or Southward using the virtual channel number one ($Y1^*$), or moving downward (Z^-).

A. Proof of Deadlock-Freedom

Generally, a cycle can be formed if packets are able to take both positive and negative directions along at least two dimensions [8]. 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 (360-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. Thus, 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 X and Z direction and the virtual channel number along Y. That is, Set1 only covers positive direction of X and Z while Set2 covers the negative parts. The two sets are disjoint in virtual channel number along Y. 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). 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 will be proved.

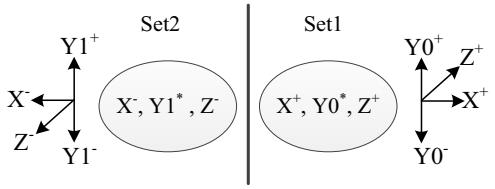

Figure 1. Two different regions

B. Routing Algorithm Procedure

Based on the presented Set definition, 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 eastmost 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 eastmost column guarantees delivery of packets between each pair of source and destination nodes. The routing algorithm can be generally described as follows:

1) Source and destination are on the same layer

If the destination is to the East of the source, Set1 will be used to deliver the packet to the destination; otherwise, Set2 will be applied.

2) Destination is on the upper layer

Set1 should be used first since moving upward is just allowed in Set1. When the packet reaches the destination layer, depending on the position of the destination router, the packet either continues routing in Set1 (destination is to the East of the current node) or switches to Set2 (destination is to the West of the current node). In more details, the destination region can be in East-Up or West-Up of its source. In order to illustrate the two scenarios, a 4*3*2 network is shown in Figure 2 having four TSVs connecting 0 to 12 (0-12), 8-20, 10-22 and 7-19. The TSVs are bidirectional. Based on the introduced algorithm, if the source node 17 targets the node 1 or node 4 as its destination, two elevators (i.e. 10-22 and 7-19, bolded in the figure) can be taken to transmit the packet to the destination layer and finally Set2 is used for delivering the packet to the destination. Moreover, when the source node 17 wants to

send a packet to the destination at the node 7, again both bolded elevators are eligible and Set1 will be sufficient to deliver the packet to the destination.

3) Destination is on the lower layer

Packets should be delivered through a TSV which is located to the east side of the destination. The reason is that as soon as using the downward channel (Z^- from Set2), no further movement to the East direction is possible. So, the packet has to move toward East sufficiently before moving downward. The destination can be in East-Down or West-Down of the source. Consider two examples according to Figure 3. First, the source node 6 sends a packet to the destination 19. In this case, the elevator 10-22 should not be used as the packet has to take the East direction after delivering to the destination layer and it is not possible when the packet is in Set2, so the packet will be blocked. The elevator 7-19 is the only eligible elevator in this example. Second, for sending a packet from the source node 6 to the destination 17, the elevator 10-22 is between the source and destination nodes, and thus it can be used. The elevator 7-19 is also valid and can be used. It is the same condition as the case when the source node 1 wants to send a packet to the destination 17. Since there is no elevator between the source and the destination, the elevators on the East side of the destination are eligible which are 10-22 and elevator 7-19.

C. *Shortest Manhattan Distance Elevator Assignment*

Elevators are assigned to routers at runtime which is based on the shortest Manhattan distance. Thereby, the elevator which has the least hop count is chosen with respect to both the source and destination. The hop count is defined as the Manhattan distance from the source to the elevator plus the Manhattan distance from the elevator to the destination. Finding the nearest elevator is done at runtime according to the source and destination. The pseudo code is shown in Figure 4. It should be mentioned that when the destination is in the west side of the source, first the algorithm tries to find an elevator between the source and destination. If this condition is not met, then the elevators on the east side of the current node will be examined for finding the least Manhattan distance.

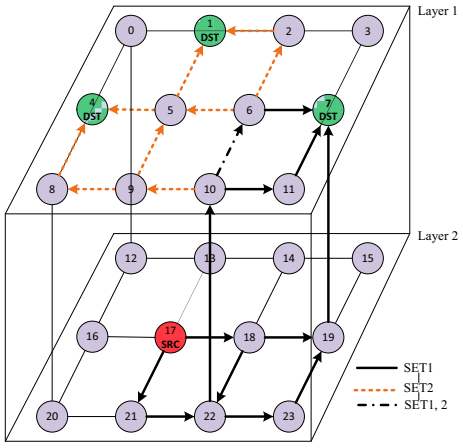

Figure 2. Example of destination in the upper layer

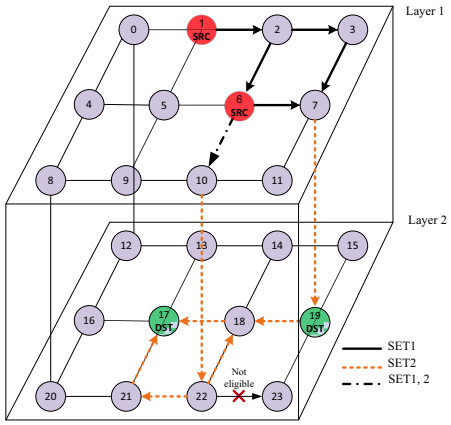

Figure 3. Example of destination in the lower layer

```
Xc, Yc and Zc: X, Y and Z coordinates of the current router
Xd, Yd and Zd: X, Y and Z coordinates of the destination router
Xe, Ye and Ze: X, Y and Z coordinates of the elevator

Manhattan Distance (elevator, current, destination) =
(abs(Xe-Xc) + abs(Ye-Yc)) + (abs(Xd-Xe) + abs(Yd-Ye))

If ((Zd >= Zc) or ((Zd < Zc) and (Xd >= Xc))){
    For (all elevators)
        If (Xe >= Xd){
            The elevator is eligible;
            Calculate Manhattan Distance;}
Else if ((Zd < Zc) and (Xd < Xc)){
    For (all elevators)
        If ((Xd <= Xe) and (Xe <= Xd)) or (Xe >= Xc){
            The elevator is eligible;
            Calculate Manhattan Distance;}
Choose an eligible elevator with the shortest Manhattan distance;
```

Figure 4. Pseudo code for the SMD algorithm

IV. Results and Discussion

A. *Reliablity Evaluation*

In the elevator-first routing algorithm, 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.

The ETW can tolerate faults on TSVs for as long as there is at least one healthy elevator at the eastmost 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 eastmost elevator for delivering the packet to the destination.

B. *Latency Analysis*

In order to evaluate the efficiency of the proposed routing algorithms, AccessNoxim simulator is used [11]. 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).

In order to analyze the latency of the proposed routing algorithm, a 4*4*4 network is considered. All the routers

have 5-flit FIFOs and the packet size is 8 flits. The TSVs are pillars and they are located at nodes 0, 2, 7, 8 and 10 in the first layer according to Figure 2 although it is a 4*3*2 network. The ETW algorithm is compared with the elevator-first routing algorithm. Figures 5 and 6 illustrate the latency for the network for random and shuffle traffic respectively.

In the random traffic pattern, each router generates data packets and sends them to a random destination. As Figure 5 shows, for the packet injection rate lower than 0.02 the proposed routing mechanism works nearly the same as the elevator-first routing. Utilizing an extra virtual channel in the elevator-first algorithm will improve the saturation point of the network as it is illustrated in Figure 5.

In the shuffle traffic (Figure 6), 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 ETW routing algorithm works even better than the elevator-first algorithm.

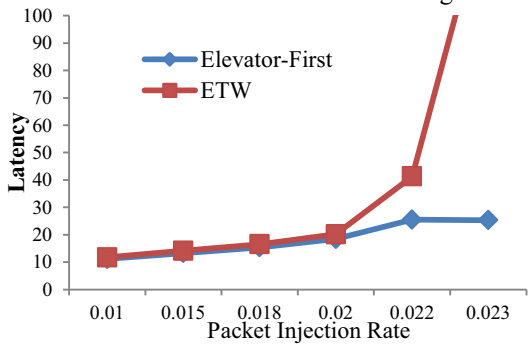

Figure 5. Performance under Random traffic

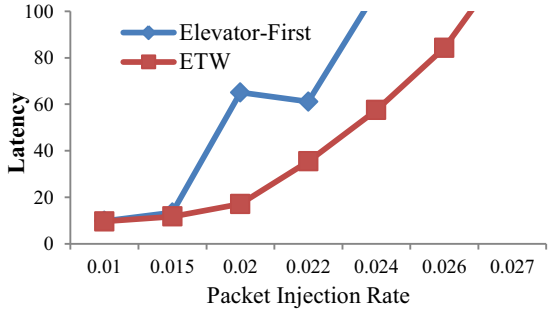

Figure 6. Performance under Shuffle traffic

C. *Power and Area Analysis*

Table 1 compares power consumption of the elevator-first routing algorithm versus the proposed routing mechanism for a network having five TSVs reported by AccessNoxim [11]. According to the results, the power consumption of the elevator first routing algorithm is more than the proposed routing mechanism. As the results show, the power consumption for shuffle traffic transferring most transmission to the vertical link is less than the power consumption for random traffic pattern.

Table 1. Power Consumption Comparison

	Random		***Shuffle***	
	Average power*e006	***Average power per router*e-008***	***Average power*e006***	***Average power per router*e-008***
Elevator-First	4.50	7.03	3.96	6.19
ETW	4.09	6.39	3.33	5.2

V. CONCLUSION

A 3D-NoC needs a large number of vertical links, while 3D integration has a major limitation on the number of TSVs. In order to reduce the number of vertical links, partially connected 3D-NoC is considered. Toward this direction, a distributed adaptive routing algorithm for heterogeneous 3D-NoCs is proposed. The algorithm is deadlock-free by having only one extra virtual channel along the Y dimension. The algorithm uses minimal and non-minimal paths for delivering a packet to a destination according to the region of destination as compared to the source. Moreover, the algorithm is fault resilient as long as there is at least one healthy elevator at the eastmost column.

ACKNOWLEDGMENT

This work supported by VINNOVA-MarieCurie within the ERoT project 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] International Technology Roadmap for Semiconductors, 2009.

[4] M Ebrahimi, M Daneshtalab, J Plosila, H Tenhunen, "MAFA: adaptive fault-tolerant routing algorithm for networks-on-chip", *in Proc. of DSD*, pp. 201-207, 2012.

[5] M Ebrahimi, M Daneshtalab, J Plosila, F Mehdipour, "MD: minimal path-based fault-tolerant routing in on-chip networks", *in Proc. of ASP-DAC*, pp. 35-40, 2013.

[6] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila and H. Tenhunen, "CATRA: Congestion Aware Trapezoid-Based Routing Algorithm for On-Chip Networks," *Design Automation and Test in Europe*, pp. 320-325, 2012.

[7] S. Pasricha and Y. Zou, "A Low Overhead Fault Tolerant Routing Scheme for 3D Networks-on-Chip," *International Symposium on Quality Electronic Design*, pp. 1-8, 2011.

[8] 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.

[9] 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.

[10] 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.

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