

# DyXYZ: Fully Adaptive Routing Algorithm for 3D NoCs

Masoumeh Ebrahimi<sup>1</sup>, Xin Chang<sup>2</sup>, Masoud Daneshtalab<sup>1</sup>, Juha Plosila<sup>1</sup>, Pasi Liljeberg<sup>1</sup>, Hannu Tenhunen<sup>1</sup>

<sup>1</sup>Department of Information Technology, University of Turku, Finland

<sup>2</sup>Department of Electronic and Microelectronics, University of Mons, Belgium  
{masebr, masdan, juplos, pakrli, hanten}@utu.fi; xin.chang@umons.ac.be

*Abstract—Traditional methods in 3D NoCs simply use a deterministic routing algorithm to deliver packets from a source to a destination node. However, deterministic methods are unable to distribute the traffic load over the network, which results in degrading the performance. In this paper, we present a fully adaptive routing algorithm for 3D NoCs, named DyXYZ. In DyXYZ, the congestion information at the input buffer of the neighboring routers is used as congestion metric to select among the output channels. This algorithm is proven to be deadlock free by using 4, 4, and 2 virtual channels along the X, Y, and Z dimensions, respectively.*

*Keywords—component; 3D network, fully adaptive routing algorithms; alternative routes*

## I. INTRODUCTION

The performance of future System-on-Chip (SoC) designs is limited by the chip area and interconnect problem. According to the Technology Roadmap for Semiconductors (ITRS), There-Dimensional (3D) integration is counted as a promising technology to sustain the performance improvement beyond 65nm [1]-[6]. In 3D integration technology, multiple active silicon layers are stacked together using Through-Silicon-Vias (TSVs). The major advantage of 3D ICs is the considerable reduction in the footprint and global interconnect length, resulting in increased performance [1]. 3D Networks-on-Chip (3D NoCs), on the other hand, are emerging as a solution for the interconnect complexity in 3D SoCs [7][8].

Routing algorithms are used to route a packet from a source to a destination. Routing algorithms can be classified according to the place where routing decisions are taken [9]. In a source routing [10], the path can be established at the source node prior to the packet injection while in a distributed routing, the path is determined in a distributed manner while the packet travels across the network. Unlike distributed routing, in the source routing approach, the path information is carried by packets, thus routers does not require any extra computation for making routing decisions. This results in a simpler routing unit and a faster communication. However, as packets are required to carry the path information (i.e. that is usually large), the bandwidth requirement and scalability become major challenges. When distributed routing is used, the path is computed at each intermediate node. Distributed routing can be classified into deterministic and adaptive [11]-[16]. A deterministic routing algorithm uses a fixed path for each pair of source and destination nodes. Implementations of deterministic routing algorithms are simple but they are unable to balance the load across the links. The simplest deterministic routing method is dimension order routing. The dimension

order routing algorithm routes packets by crossing dimensions in strictly increasing order, reducing the offset in one direction to zero before routing in the next one. Deterministic routing algorithms perform well with uniform traffic pattern while they are very inefficient under non-uniform traffic. In contrast, in adaptive routing algorithms, a packet is not restricted to a single path when traveling from a source node to its destination. Thereby, adaptive routing algorithms can decrease the probability of routing packets through congested regions.

Several partially and fully adaptive routing algorithms are introduced in 2D networks such as DyXY [17] and Odd-Even [18]. The routing selection of these algorithms is usually performed using the congestion status of the network. Many of them consider local traffic condition in the routing decision in which each router analyses the congestion conditions of its own and adjacent routers to choose an output channel. This group of algorithms could improve the performance significantly as compared to dimension order routing due to the distribution of packets over the network. However, routing decisions based on local congestion information may lead to an unbalanced distribution of traffic load. Some other algorithms such as CATRA [19] and HARAQ [20] take more global information into account, reducing the probability of making a wrong decision. However, regardless of the imposed complexity and area overhead, providing global information is complex. In sum, adaptive routing algorithms based on global congestion information improve the performance over the methods based on local congestion information. This performance gain is at the cost of a large area overhead, a more complex routing unit, and the need for congestion detection and propagation mechanism. There are few partially adaptive methods presented in 3D NoCs such as MAR [8].

In this paper, we present a fully adaptive routing algorithm, named DyXYZ, to route packets adaptively in the network. To the best of our knowledge this is the first work presenting a fully adaptive routing algorithm in 3D mesh NoCs while guaranteeing the deadlock freeness. This method can efficiently alleviate congestion in the network by choosing among multiple paths based on the traffic condition.

The paper is organized as follows. The proposed fully adaptive routing algorithm (DyXYZ) in 3D NoCs is given in Section II. Results are reported in Section III while the conclusion is given in the last section.

## II. DYNAMIC XYZ FOR 3D NOCS

### A. Dimension Order Routing in 3D NoCs

In a dimension order routing, packets are crossing dimensions in a strictly increasing (or descending) order,

reducing the offset along one dimension to zero before routing in the next dimension. For example in Fig. 1, three packets (i.e. P1, P2, and P3) are sent from the source node 0 to destination nodes 8, 16, and 22. As shown in this figure, at the source node, all the three packets should be sent through the east direction. As they cannot be sent through the east link simultaneously, a queue of packets is created behind this link (acting as a performance bottleneck) while the links in the north and up directions are free. Similarly, at node 1, packets P2 and P3 are in competition with each other to receive the north output channel while the link in the up direction is free. These examples show the inefficiency of the XYZ routing algorithm in distributing packets.

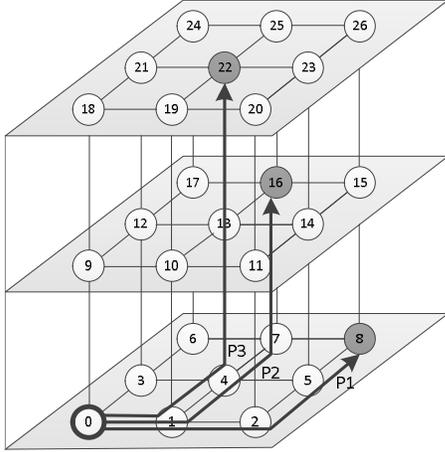


Fig. 1. XYZ routing algorithm.

### B. Fully Adaptive Routing Algorithm in 3D Mesh NoCs

In this subsection, we propose a fully adaptive routing algorithm for 3D NoCs. Using this algorithm, at each intermediate node, packets can be delivered through three directions, as long as the distance along a direction has not reached zero. In this method, the network is divided into eight sub-networks as ENU, END, ESU, ESD, WNU, WND, WSU, and WSD where N, S, E, W, U, and D stand for North, South, East, West, Up, and Down directions. Since packets can be delivered on X, Y, or Z direction without any restriction the deadlock freeness should be guaranteed. If we could prove that the sub-networks are disjoint and each uses a separate set of virtual channels, then the whole network is deadlock free. According to Table 1, there are four sub-networks covering East direction (i.e. ENU, END, ESU, and ESD). For each sub-network, a separate virtual channel number is allocated, keeping the sub-networks disjoint in the X dimension. For instance,  $vc_0$ ,  $vc_1$ ,  $vc_2$ , and  $vc_3$  along the X dimension are allocated to ENU, END, ESU, and ESD sub-networks, respectively. WNU, WND, WSU, and WSD sub-networks use the same virtual channels in the X dimension, but in the opposite directions. In other words, each of east and west direction involves half of virtual channels in the X dimension. Similarly, four virtual channels are enough to cover the Y+ and Y- directions in all regions. ENU, END, WNU, and WND utilize the positive directions of the virtual channels in the Y dimension while ESU, ESD, WSU, and WSD contain the

negative directions of them. The similar perspective is applied to the Z dimension. As shown in Table 1, virtual networks cover separate sets of virtual channels and thus no deadlock has occurred in the network.

Table 1. Assigning virtual channels to sub-networks

Sub-network	Virtual Channels
ENU	X0,Y0,Z0
END	X1,Y1,Z0
ESU	X2,Y0,Z1
ESD	X3,Y1,Z1
WNU	X0,Y2,Z2
WND	X1,Y3,Z2
WSU	X2,Y2,Z3
WSD	X3,Y3,Z3

In Fig. 2, different colors are used to distinguish between separate set of virtual channels for each sub-network. Fig. 3 shows the paths diversity of sending a packet from the source node 13 to destinations 2 and 24, located in sub-networks ESD and WNU, respectively. Packets can be delivered through the bolded lines to the destination nodes.

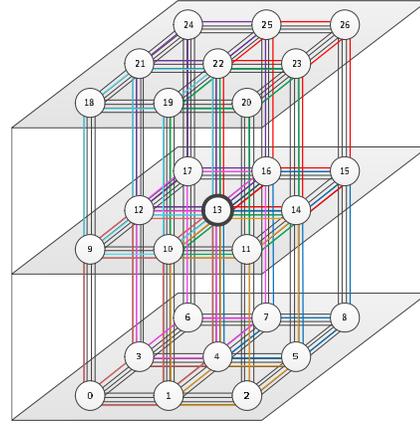


Fig. 2. Different virtual networks cover different sets of virtual channels.

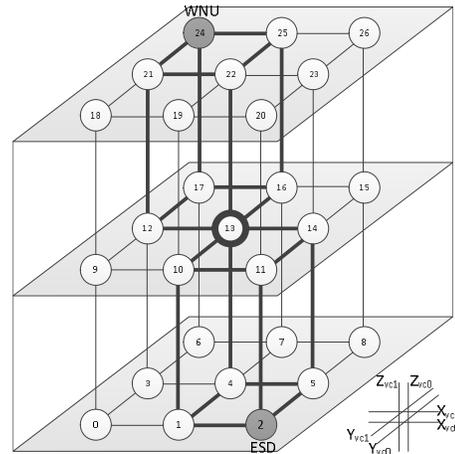


Fig. 3. Path diversity from source node 13 to destinations 2 and 24

### C. Reducing the Number of Virtual Channels to Two along the Z dimension

The presented method requires four virtual channels long each dimension. However, the number of virtual channels can be reduced to two virtual channels along one of dimensions, we called this method Dynamic XYZ (DyXYZ). In this way, the whole network can be split into two main sub-networks, each having four sub-networks. Dividing the network into two parts can be done along one of dimensions, reducing the number of virtual channels from four to two for that dimension. Since two main sub-networks are separated, the network remains deadlock free. On the other hand, the sub-networks within each main sub-network use different sets of virtual channels on the two remaining dimensions. Therefore, the eight sub-networks are disjoint and the network is deadlock free. By using DyXYZ, packets can be routed along X, Y or Z dimension at each intermediate node without creating any cycles. All the routing options for DyXYZ routing algorithm using two virtual channels in the Z dimension and four virtual channels in the X and Y dimensions is shown in Table 2. Note that, the positions are determined regarding the source and destination positions but not the current node.

Table 2. Routing options in DyXYZ routing algorithm.

Direction	Virtual Channels
ENU	X0,Y0,Z0
END	X1,Y1,Z1
ESU	X2,Y0,Z0
ESD	X3,Y1,Z1
WNU	X0,Y2,Z0
WND	X1,Y3,Z1
WSU	X2,Y2,Z0
WSD	X3,Y3,Z1
EN	(X0,X1),(Y0,Y1)
ES	(X2,X3),(Y0,Y1)
EU	(X0,X2),(Z0)
ED	(X0,X3),(Z1)
WN	(X0,X1),(Y2,Y3)
WS	(X2,X3),(Y2,Y3)
WU	(X0,X2),(Z0)
WD	(X1,X3),(Z1)
NU	(Y0,Y2),(Z0)
ND	(Y1,Y3),(Z1)
SU	(Y0,Y2),(Z0)
SD	(Y1,Y1),(Z1)
E	(X1,X2,X3,X4)
W	(X1,X2,X3,X4)
N	(Y0,Y1,Y2,Y3)
S	(Y0,Y1,Y2,Y3)
U	(Z0,Z1)
D	(Z0,Z1)

As shown in Table 2, packets have some degree of adaptiveness on selecting among virtual channels when the distance along one or two dimensions are non-zero (i.e. the destination is to the EN, ES, WN, ES, EU, ED, WU, WD, NU, ND, SU, SD, E, W, N, S, U, and D position of the source node). For example, when the destination is to the EN of the source node, the packet can be delivered through each of the vc0 or vc1 of the X or Y dimension. When destination is to the E, W, N or S directions, each of four available virtual channels can be selected by packets. As the selection function and for a better efficiency, the routing unit chooses a dimension which is less congested. The congestion is determined based on the number of free buffer slots at the corresponding input buffer of the neighboring router. When the number of free buffer slots is greater than the threshold value, the input buffer is counted as a congested one. In this work, the threshold value is considered to be 70% of the total input buffer size.

```

Dir:Direction
N:North; S:South; E:East; W:West; U:Up; D:Down
-----
--Regions regarding source and dest position--
If (Dir=ENU or Dir=EN or Dir=U or Dir=N)
    Selects X0 or Y0 or Z0;
If (Dir=END or Dir=ED or Dir=E)
    Selects X1 or Y1 or Z1;
If (Dir=ESU or Dir=ES or Dir=EU)
    Selects X2 or Y0 or Z0;
If (Dir=ESD or Dir=SD or Dir=S)
    Selects X3 or Y1 or Z1;
If (Dir=WNU or Dir=WU or Dir=NU)
    Selects X0 or Y2 or Z0;
If (Dir=WND or Dir=ND or Dir=WN)
    Selects X1 or Y3 or Z1;
If (Dir=WSU or Dir=WS or Dir=SU)
    Selects X2 or Y2 or Z0;
If (Dir=WSD or Dir=WD or Dir=W or Dir=D)
    Selects X3 or Y3 or Z1;
End If;

```

Fig. 4. Assigning virtual channels to directions in the DyXYZ routing algorithm

### III. RESULTS AND DISCUSSION

To assess the efficiency of the proposed DyXYZ, we have developed a cycle-accurate NoC simulator based on wormhole switching in 3D-mesh configuration. The simulator calculates the average delay and the power consumption for the message transmission. To estimate the power consumption of networks, we have used Orion [29] as well as the power and delay values of vertical links in [28]. XYZ and the proposed DyXYZ routing algorithms use four virtual channels in the X and Y dimensions and two virtual channels in the Z dimension. The on-chip network, considered for experiment is formed by a typical wormhole router structure including input buffers, a routing unit, a switch allocator and a crossbar. Each router has 7 input/output ports, a natural extension from a 5-port 2D router by adding two ports to make connections to the upper and lower layers [25][26]. There are some other types of 3D routers such as the hybrid router [26] and MIRA [27]. However, since

router efficiency is out of the concept of this paper, we have chosen simple 7-port router in our simulation. The arbitration scheme of the switch allocator in the typical router structure is round-robin. The data width and the frequency were set to 32 bits and 1 GHz, respectively, and each input channel has a buffer size of 7 flits. The packet size was assumed to be 5 flits. We also assume that the 3D mesh topology is regular and the delays on wires will not exceed the clock period. For the performance metric, we use the latency defined as the number of cycles between the initiation of a packet and the time when the tail of the packet reaches the destination.

### A. Performance Analysis

#### 1) Uniform Traffic Profile

In the uniform traffic profile, each processing element (PE) generates data packets and sends them to another PE using a uniform distribution. In Fig. 5, the average communication delay as a function of the average packet injection rate is plotted for all schemes. As observed from the results, XYZ leads to a lower latency than the DyXYZ method. The reason is that, under uniform traffic, dimension order routings are the best matches for evenly distributing traffic over the network.

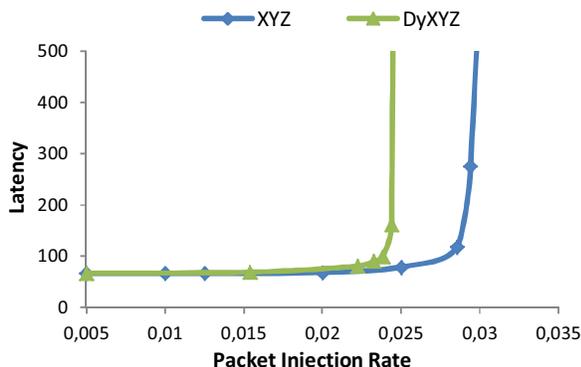


Fig. 5. Performance under uniform traffic profile.

#### 2) Hotspot Traffic Profile

Under the hotspot traffic pattern, some nodes are chosen as hotspots receiving an extra portion of the traffic in addition to the regular uniform traffic. In simulations, given a hotspot percentage of  $H$ , a newly generated message is directed to each hotspot node with an additional  $H$  percent probability. We simulate the hotspot traffic with four hotspot nodes at positions (2, 1) and (3, 1) in layer 2 and the same positions in layer 3 in the  $4 \times 4 \times 4$  mesh network. The performance of each network with  $H=10\%$  is illustrated in Fig. 6. As observed from the figure, DyXYZ leads to a better performance since it considers the congestion condition of the network in the routing decision. In fact, the improvement is achieved by smoothly balancing the traffic load over the network which reduces the number of the hotspots and, hence, improving the performance rather than the XYZ routing algorithm. It is worth mentioning that DyXYZ can adaptively select between output directions based on the congestion condition in the input buffer of the neighboring routers. To support in-order delivery when the underlying routing algorithm is DyXYZ, either reordering buffer or flow

control mechanism is required. In the XYZ routing algorithm, all packets follow the same path and do not require a reordering mechanism. However, by using virtual channels in XYZ, it cannot support in-order delivery since packets might reach destination in an out-of-order manner due to facing different congestion in different virtual channels. To deal with this problem, an approach similar to [22] is needed which assigns a virtual channel to a flow, requiring look up table at each intermediate node. In both Fig. 5 and Fig. 6, the performance of the XYZ routing algorithm is obtained for the best case where switching between virtual channels is possible (i.e. in-order delivery cannot be supported).

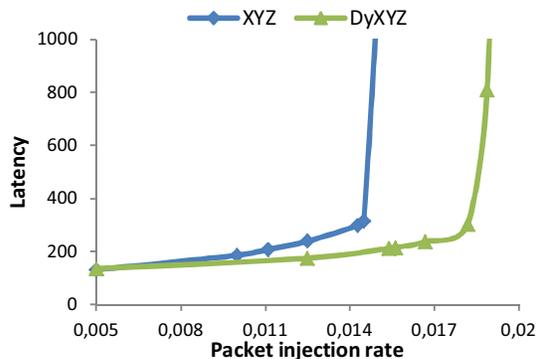


Fig. 6. Performance under hotspot traffic

### B. Physical Analysis

To assess the area overhead and power consumption of the presented communication architectures, the whole platforms including network interfaces [30] and routers are synthesized using Synopsys Design Compiler with the 65nm LP CMOS technology from STMicroelectronics, while the backend is performed with the Cadence Encounter tool. Depending on the technology and manufacturing process, the pitches of TSVs can range from  $1\mu\text{m}^2$  to  $10\mu\text{m}^2$  [26][31][32]. In this work, the pad size for TSVs is assumed to be  $5\mu\text{m}^2$  with the pitch of around  $8\mu\text{m}^2$ . Since all algorithms (i.e. the XYZ and DyXYZ routing algorithms) uses the same number of virtual channels, the total area cost of a layer for all platforms is almost similar and about 18 mm<sup>2</sup>. The power dissipation of the XYZ and DyXYZ methods were calculated and compared under the hotspot traffic model. The results for the average power under hotspot traffic are shown in Table 3. The average power values are computed near the saturation point, 0.014 under the hotspot traffic profile. As a result, the average power consumption of the DyXYZ scheme is 11% larger than that of the XYZ scheme. This indicates that DyXYZ distributes the traffic better.

Table 3. Power dissipation

Network platforms	Power (W) dynamic & static
XYZ	19.34
DyXYZ	17.21

#### IV. CONCLUSION

Traffic can be distributed across the network by using alternative paths. In this paper, we proposed a fully adaptive routing algorithm, named DyXYZ. In this method, at each intermediate node, a packet is sent to a direction which is less congested. This algorithm is proven to be deadlock free by using 4, 4, and 2 virtual channels along the X, Y, and Z dimensions, respectively.

Although this algorithm can alleviate the congestion, it cannot be used for the applications demanding in-order delivery. In-order delivery is a basic requirement for many applications but it has been rarely investigated in literature. In order to address the in-order delivery issue in 3D network, we are working to propose a method as a future work.

#### ACKNOWLEDGMENT

The authors wish to acknowledge Nokia, Elisa, and KAUTE Foundations for the partial financial support during the course of this research.

#### References

- [1] V. F. Pavlidis, E.G. Rochester Friedman, "3-D Topologies for Networks-on-Chip," Vol.15, No.10, pp. 1081-1090, 2006.
- [2] M. Ebrahimi, et al., "Cluster-based Topologies for 3D Networks-on-Chip Using Advanced Inter-layer Bus Architecture," Elsevier Journal of Computer and System Sciences (*JCSS-elsevier*), 2012.
- [3] L.P. Carloni, P. Pande, X. Yuan, "Networks-on-chip in emerging interconnect paradigms: Advantages and challenges", pp. 93-102, 2009.
- [4] M. Elgamel, M. A. Bayoumi, "Interconnect Noise Optimization in Nanometer Technologies", 2010.
- [5] T. C. Xu et al., "A Study of 3D Network-on-Chip Design for Data Parallel H.264 Coding," *Microprocessors and Microsystems*, Vol. 35, No. 7, pp. 603-612, 2011.
- [6] T. C. Xu et al., "Optimal Number and Placement of Through Silicon Vias in 3D Network-on-Chip," In Proceedings of the 14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp.105-110, 2011.
- [7] M. Daneshtalab et al., "HIBS-Novel Inter-layer Bus Structure for Stacked Architectures," in Proceedings of IEEE International 3D Systems Integration Conference (3DIC), pp. 1-7, 2012.
- [8] M. Ebrahimi et al., "Exploring Partitioning Methods for 3D Networks-on-Chip Utilizing Adaptive Routing Model," in Proceedings of 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 73-80, 2011.
- [9] J. Duato, C. Yalamanchili, L. Ni, "Interconnection networks: an engineering approach," Morgan Kaufmann Publishers, 2003.
- [10] S. Mubeen, S. Kumar, "Designing Efficient Source Routing for Mesh Topology Network on Chip Platforms", pp. 181-188, 2010.
- [11] M. Daneshtalab et al., "Adaptive Input-output Selection Based On-Chip Router Architecture," *Journal of Low Power Electronics (JOLPE)*, Vol. 8, No. 1, pp. 11-29, 2012.
- [12] M. Dehyadegari et al. "An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip," in Proceedings of 13th IEEE/NASA-ESA International Conference on Adaptive Hardware and Systems (AHS), pp. 208-214, 2011.
- [13] M. Ebrahimi et al. "HAMUM – A Novel Routing Protocol for Unicast and Multicast Traffic in MPSoCs," in Proceedings of 18th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Computing (PDP), pp. 525-532, 2010.
- [14] M. Ebrahimi et al., "An Efficient Dynamic Multicast Routing Protocol for Distributing Traffic in NOCs," in Proceedings of 12th ACM/IEEE Design, Automation, and Test in Europe Conference (DATE), pp. 1064 - 1069, April 2009.
- [15] B. Marvasti et al., "PAMPR: Power-Aware and Minimum Path Routing Algorithm for NoCs," in Proceedings of 15th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 418-421, Sep 2008, Malta.
- [16] M. Daneshtalab et al. "Distributing Congestions in NoCs through a Dynamic Routing Algorithm based on Input and Output Selections," in Proceedings of 20th IEEE International Conference on VLSI Design (VLSID), pp. 546-550, 2007.
- [17] M. Li, Q. Zeng, W. Jone, "DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip", in proceedings of DAC, pp. 849-852, 2006.
- [18] G. M. Chiu, "The odd-even turn model for adaptive routing", IEEE Transaction on Parallel and Distributed Systems, 11:729 – 738, 2000.
- [19] M. Ebrahimi et al., "CATRA-Congestion Aware Trapezoid-based Routing Algorithm for On-Chip Networks," in Proceedings of 15th ACM/IEEE Design, Automation, and Test in Europe (DATE), pp. 320-325, 2012.
- [20] M. Ebrahimi et al. "HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks," in Proceedings of 6th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 19-26, 2012.
- [21] S. Murali, et al., "A Method for Routing Packets Across Multiple Paths in NoCs with In-Order Delivery and Fault-Tolerance Guarantees," VLSI Design, vol. 2007, Article ID 37627, 11 pages, 2007..
- [22] M. Lis et al., "Guaranteed in-order packet delivery using Exclusive Dynamic Virtual Channel Allocation", technical report CSAIL-TR-2009-036, Massachusetts Institute of Technology, 2009.
- [23] M. Lis, et al., "Path-Diverse Inorder Routing", International conference on Green Circuits and Systems (ICGCS), pp. 311-316, 2010.
- [24] S. Mubeen, Sh. Kumar, "Designing Efficient Source Routing for Mesh Topology Network on Chip Platforms", in Proceedings of the 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, pp.181-188, 2010.
- [25] Z. Liu, J. Duato, "Adaptive Unicast and Multicast in 3D Mesh Networks", Proc. of the Twenty-Seventh Hawaii Int. Conf., v.1, pp.173-182, 1994.
- [26] F. Li, C. Nicopoulos, T. Richardson, et. al, "Design and Management of 3D Chip Multiprocessors Using Network-in-Memory", ISCA-33, 2006.
- [27] D. Park et. al., "MIRA: A Multi-Layered On-Chip Interconnect Router Architecture", 35th International Symp. on Computer Architecture (ISCA), pp.251-261, 2008.
- [28] V.F. Pavlidis and E.G. Friedman, "3-D topologies for networks-onchip", IEEE Transactions on Very Large Scale Integration Systems, v.15, i.10, pp.1081-1090, 2007.
- [29] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in proc. of DATE, pp. 423-428, 2009.
- [30] M. Ebrahimi, et al., "A High-Performance Network Interface Architecture for NoCs Using Reorder Buffer Sharing," in Proceedings of 18th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Computing (PDP), pp. 547-550, 2010.
- [31] M. Daneshtalab et al., "Memory-Efficient Logic Layer Communication Platform for 3D-Stacked Memory-on-Processor Architectures," in Proceedings of IEEE International 3D Systems Integration Conference (3DIC), pp. 1-8, Jan. 2012.
- [32] U. Kang et al., "8Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology," In Proc. International Solid State Circuits Conference (ISSCC), pp. 130–131, 2009.