STA: A Highly Scalable Low latency Butterfly Fat Tree Based 3D NoC Design

Avik Bose[†], Prasun Ghosal[†], Saraju P. Mohanty[‡]
[†]Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, WB, India
[‡]University of North Texas, Denton, TX 76203, USA
Email: {avik, p_ghosal}@it.iiests.ac.in, saraju.mohanty@unt.edu

***Abstract*—Since the past decade Network-on-Chip has evolved as the most dominant and efficient solution in on-chip communication paradigm for multi-core systems. With the growing number of on-chip processing cores modern three dimensional NoC design is facing several challenges originating from various network performance parameters like latency, hop count etc. Scalability and network efficiency have generated an important trade off in 3D NoC design, which needs to be balanced, especially for application specific NoC design. Tree based topologies outperform mesh based topologies in terms of network latency and throughput with increasing injection rate of packets/flits. But on the other hand, floor planing becomes much more complex for tree based designs with increasing number of IP blocks compared to mesh due to the hierarchical structure. This paper introduces a novel 3D NoC architecture named Split Tree Architecture (STA), based on butterfly fat tree, which is highly scalable while maintaining low network latency and hop count significantly. There are latency improvements of 51-91%, 84-96%, 55-96%, and 48-96% over mesh, torus, butterfly, and flattened butterfly topologies respectively. Average hop count is improved by 44% and 30% over mesh and torus. Average and minimum acceptance rates are improved by 3-8% and 3-12% over torus and, 4-7% and 4-12% over flattened butterfly. In comparison to the previously reported state of the art 3D BFT based designs, STA achieves performance improvements of 19-78%, 2-42%, 0.2-0.6%, and around 20%, for average latency, average acceptance rate, minimum acceptance rate, and average hop count respectively.**

***Index Terms*—Network on Chip, 3D NoC, Split Tree Architecture, High Scalability, Low Network Latency.**

I. Introduction and Motivation

A. Introduction

Integration technology in deep sub micron regime was facing major performance bottleneck at its earlier phase, due to long interconnect delays [1]. Major challenge for System-on-Chip (SoC) designers was to provide functionally correct and reliable operation of the interacting components [2]. On-chip physical interconnections were facing a limiting factor for performance [3]. With the advent of SoC technology, the focus was shifted from a computation centric design to a communication centric paradigm [4]. Limitations of bus based interconnect system have led NoC to come into the picture. In addition to provide a solution for the global wire delay problem, the NoC paradigm also eases integration of high numbers of intellectual property (IP) cores in a single SoC [5]. However, with growing scalability along with growing design footprint, 2D floor planing was difficult due to overall chip area constraints. Hence three dimensional NoC design eventually evolved as a more promising and sustainable solution. With growing scalability modern 3D NoC designs are facing challenges ralated to various network performance parameters viz. latency, hop count etc. Network topology and routing play an important role in 3D NoC design as it has profound effect on overall performance. This work presents an advancement in the same track.

B. Related Research

It is investigated in [6], how the interconnection network started to play a more and more important role in determining the performance of the entire chip. To provide low latency and high bandwidth communication in NoCs many researches have tried various approaches to optimize the design by developing fast routers [7]–[10] and designing new network topologies [11]–[13]. Performance of NoC has been improved with the addition of diagonal links in mesh, as proposed in [14] and [15]. Where as, [16] shows performance improvements over normal mesh connection, by incorporation of long range links in 2D scenario. A comparative study on mesh and tree based topologies for both of their 2D and 3D counterparts can be found in [5], where it is shown that with efficient router design 3D tree-based NoCs will exhibit performance benefits in terms of latency and bandwidth along with significant gain in energy dissipation and area overhead. Inter layer vertical distance is very low in comparison to flat design footprint consisting equal number of IP blocks [5]. Therefore, modern research trends have focused mainly on optimizing the vertical communications including choice of good vertical link architectures. Performance of Dynamic Time Division Multiple Access (DTDMA) bus, as a very fast vertical way of communication and as opposed to conventional Through Silicon Vias (TSV), is examined in [4]. Using DTDMA pillar, non-uniform cache architecture has been implemented as 3D Network-in-Memory to solve various L2 cache coherence issues along with 3D stacking of CPU cores [4]. Designing more efficient routers to reduce vertical hop count [17] and reducing power consumption of routers via a multi-layered 3D technology [18] are also investigated. Flattened Butterfly based long interconnect 3D network is presented in [19]. A 3D design has been proposed with four 4-ary 3-fly BFTs pertaining to each layer [20] with very low latency but without enough flexibility with increasing scalability.

C. Novel Contributions

But, none of the above could guarantee design scalability without affecting network latency and hop count significantly. In the present work, a novel 3D butterfly fat tree based design is presented that is highly scalable and its scalability does not affect network latency and hop count. An efficient distributed routing algorithm has also been proposed to manage routing for high radix routers.

Overall organization of the paper is as follows. Section II presents the problem description in details. Proposed solution has been discussed in section III with detailed analysis and proposed algorithms. Section IV summarizes the experimental results and observations. Finally section V concludes the paper with possible future directions.

II. Problem Description

Higher network diameter causes performance bottleneck due to higher network latency. For larger footprint when more routers are used to reduce the interconnect length, it incurs more hopping between the source and destination nodes in a communication. Each hopping in turn incurs delay that comprises routing computation delay, crossbar delay, and delay that induced because of the waiting time of packets/flits in routers' virtual channels. On the other hand, the interconnect length must be increased to reduce the hop count. Several researches have tried on extracting network performance as much as possible by using long wires and less routers in NoC [12], [19]. But with increasing network size floor planing becomes much more complex due to routing of long wires. Therefore, to address above issues, choice of topology plays an important role.

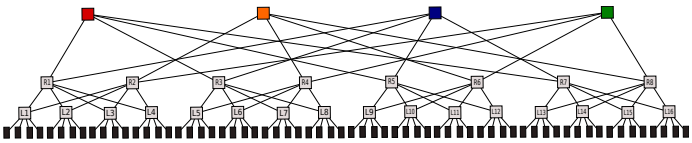


Fig. 1. Diagram of a single Butterfly Fat Tree.

Figure 1 is the logical diagram of a Butterfly Fat Tree (BFT). IP blocks that can be a processing core or cache memory bank are situated at leaf level of a BFT. For its hierarchical structure, BFT possesses uniform hopping distance between source and destination nodes (as long as they belong to the same hierarchical domain). For example, if two leaf nodes are reachable through different level 1 routers (labeled as R), the communication hop count is always four for them (no matter where they belong to in the tree). This uniform hopping characteristic makes butterfly fat tree a very attractive choice as a topology that can cope up with increasing scalability without affecting network performance. But simply increasing its size in the name of scalability imposes serious complexity in floor plan design. Also, traditional 3D butterfly fat tree design [5] lacks scalability. Previously a 3D design has been proposed with four 4-ary 3-fly BFTs pertaining to each layer [20]. Though it is able to achieve a very low latency, but still it is not flexible enough with increasing scalability. Keeping all of these factors in mind, present work proposes a novel 3D BFT based design with very high scalability and significant reduction in latency and hop count.

III. Proposed Solution

The idea behind this design is to achieve maximum scalability without affecting various network performance parameters. In order to do that the routers for a single butterfly fat tree are distributed over two layers. For this reason, the design is termed as Split Tree Architecture (STA). This strategy leaves the floor plan design much more easier with growing network diameter (to be discussed shortly).

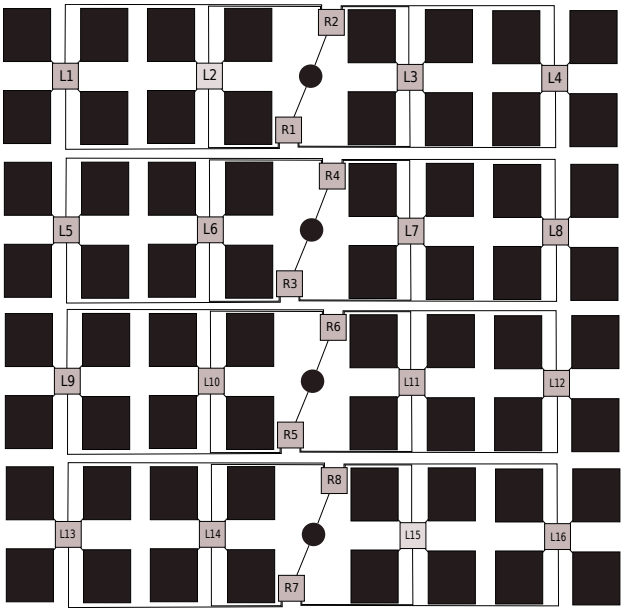


Fig. 2. Floor plan of the core layer for a single Butterfly Fat Tree.

A. The Core Layer

The lower most layer of a tree is named as core layer as it consists of all the IP blocks (processing cores and cache memory banks) along with some of the routers of the tree hierarchy. Figure 2 depicts the floor plan of the core layer for a single butterfly fat tree (shown in Figure 1). The traditional H-pattern connectivity between IP blocks and routers has been changed here to achieve the overall design plan. The reason for changing the floor plan will be clear in subsequent section. The black coloured rectangles denote IP Blocks and white ones are routers. The floor plan shown in Figure 2 is divided into four regions. Each region consists of four localities along with two regional routers labeled as R, and a circular DTDMA pillar node. DTDMA bus is a well established mechanism for faster on chip 3D communication as the time required to jump from one layer to another is equivalent to a single router hopping time [4]. Each locality, in turn, comprises one local router labeled as L and four IP Blocks connected to that router. All the local routers of a region are connected to both the regional routers.

B. The Root Layer

The root level routers of a BFT are connected to its regional routers in a specific manner. It can be seen in Figure 1 that odd numbered root routers, which are red and blue, are connected to odd numbered regional routers of the butterfly fat tree. Similarly, even numbered root routers, which are orange and green, are connected to even numbered regional routers of the butterfly fat tree. The root routers of a BFT are placed on another layer above the core layer. This is the key feature of the design that makes it highly scalable. Figure 3 shows an overlap between a core layer and the associated root layer for a single BFT.

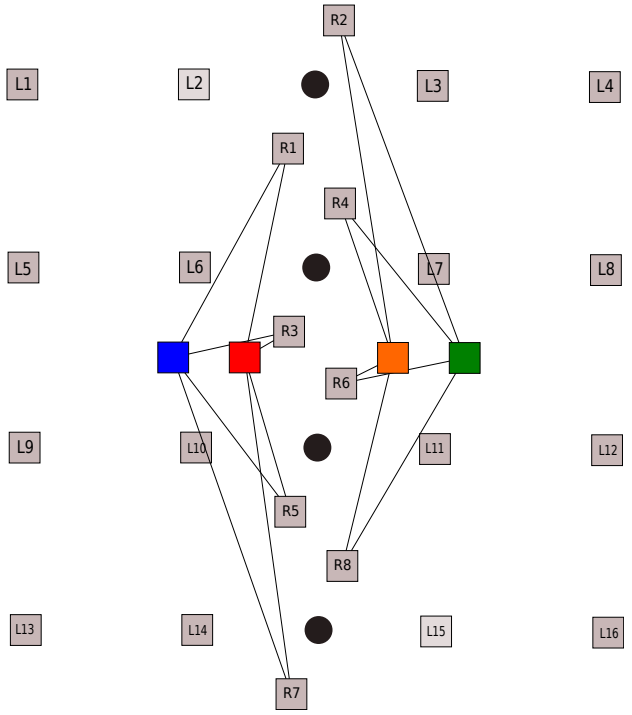


Fig. 3. An overlap of the core and root layer of for a single Butterfly Fat Tree.

From Figure 1 and Figure 3 it is clearly evident that to maintain the logical connectivity between root and regional routers of a BFT we need to place the root routers on the root layer in such a manner that it can easily be connected to the regional routers of the corresponding tree in the associated core layer. Figure **??** is the floor plan of a root layer associated with the core layer. Packets/flits destined for different regions of the same tree or a different tree on the same core layer are routed through the root routers of the associated core layer.

C. The Border Layer

To route packets/flits generated in a core layer and destined for another core layer; a new layer has been introduced above the root layer, which is called, the border layer. For every pillar node in a border layer there is a border router connected to it.

For each tree in the core layer there are four border routers in the associated border layer (B1, B2, B3, and B4). That means each border router connects a region in the associated core layer. Every border router associated with a tree in the core layer is connected to every other border routers of that tree. In a border layer a complete connection can be found among the border routers those belong to a different tree but same numbered region in the associated core layer. The reason behind this type of connectivity is discussed in subsequent section. No connections exist between the root and the border layer.

D. Scalability

Placing the root routers in one layer above, makes the design highly scalable. IP blocks can be incorporated easily in the core layer as necessary. Starting from a single IP block, we can add a locality, a region, upto a tree as per necessity.

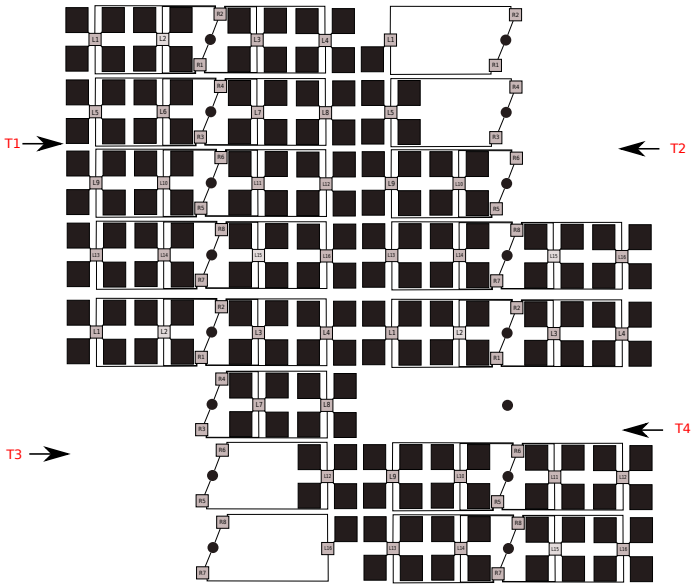


Fig. 4. A core layer depicting scalability of the design.

Figure 4 depicts a core layer for four BFTs. The trees are labeled as T1, T2, T3, and T4 respectively. Here T1 is a complete tree. T2 has four regions but the first region has only one single IP block. The second region has one locality, the third region has half of a region, and the fourth region is complete, as it has all of four localities. Same situation can be found in case of T3 but here the order is reverse. T4 does not have the second region at all. An important factor of the scalability issue is routers of a region have to be incorporated as long as a single IP block exists in the region. In spite of not having the second region T4 has the DTDMA pillar node corresponding to that region. Incorporation of DTDMA pillar depends on the maximum number of pillars on a layer across the chip. May be, in some core layer, lower or above the one shown in Figure 4, it can have the full, or a portion of the second region in the fourth BFT (T4). That is why in Figure 4, the pillar node in the second region is incorporated because a DTDMA pillar is a continuous bus through the layers of a chip. In this manner more trees can be added to the core layer. With the incorporation of a tree in the core layer all the root and border routers have to be incorporated in the associated root and border layers respectively. In this manner, the design is scalable in both 2D and 3D perspective.

E. Routing

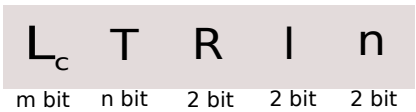


Fig. 5. Address format for routing packet/flit.

The conventional three dimensional topological design concept is faded in this work. Butterfly Fat Trees of a layer, themselves are identified with three layers; the core layer, the root layer, and the border layer. Taking these as a single unit, it is subdivided into different routing scopes; which are tree, region, and, locality respectively. The routing for this design is distributed in nature as responsibility differs across routers from one scope to another. A router's up-link is the link through which it is connected to one of the upper level routers and down-link is the link through which it is connected to one of the lower level routers; according to the tree hierarchy shown in Figure 1. In case of root routers, every link that connects one router to another in the root layer is an up link. In border layer there is no such discrimination between up and down link, as border routers are not the part of tree hierarchy. Figure 5 shows the address format of packets/flits generated by NIU (Network Interface Unit) attached to every IP block in a core layer. L_c denotes destination core layer number, T denotes the tree number within that layer, R denotes the region in that tree, l denotes the locality within that region, and n is the node within that locality for which the generated packet/flit is destined. This numbering technique in different scopes is implementation dependent. The exact values of length in bits for L_c and T are omitted because theoretically there can be any number of layers in a chip and any number of tree as well in a core layer.

1) Routing Algorithms: To denote the current router, that is processing a packet/flit, in order to find the next hop along the routing path; a nomenclature has been used in routing algorithms here, where a 'r' in suffix is used with every unit of the address format shown in Figure 5. For example, L_{c_r} is the corresponding core layer number of the current router along the routing path, which is processing a packet/flit. A border router B_{rt} denotes that it is associated with the region r of the tree t in the associated core layer. Every border router must know these two information.

Detailed routing algorithms for regional, root, and border routers are omitted due to paucity of space.

2) Analysis of Algorithms: It is evident from the routing algorithms that the regional routers are responsible for channeling both intra and inter core layer traffic. All the other routers in a core layer manage intra core layer traffic. The border routers plays an important role in inter core layer communication.

a) Measure of intra and inter core layer communication hop count: Figure 6 depicts the maximum intra and inter core layer hop count measurements.

Route( )

Input : Packet/Flit address.
Output: Next hop.

```
begin
    if The packet/flit is recieved from a down link then
        if L_c = L_{c_r} then
            if T = T_r then
                if R = R_r then
                    if l = l_r then
                        Forward the packet/flit to the down link that
                        connects the processing local router to
                        destination node n;
                    end
                    else
                        Forward the packet/flit to either of the two up
                        links those connect the processing local router
                        with two regional routers;
                    end
                end
                else
                    Forward the packet/flit to either of the two up links
                    those connect the processing local router with two
                    regional routers;
                end
            end
            else
                Forward the packet/flit to either of the two up links those
                connect the processing local router with two regional
                routers;
            end
        end
        else
            Forward the packet/flit to either of the two up links those connect
            the processing local router with two regional routers;
        end
    end
    else if The packet/flit is recieved from an up link then
        Forward the packet/flit to the link that connects n;
    end
end
```

Algorithm 1: Routing algorithm of a local router.

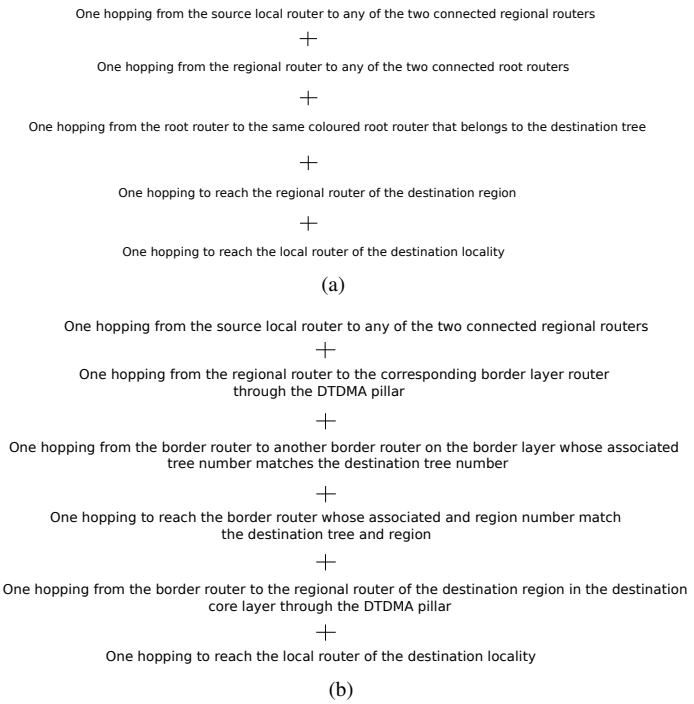


Fig. 6. a) Intra core layer hop count measurement. b) Inter core layer hop count measurement

IV. Experimental Results and Validation

A. Experiment details

Each simulation has three basic phases viz. warm up, measurement, and drain. Current latency and throughput (rate of accepted packets) for the simulation is determined after each sample period and overall throughput is determined by the lowest throughput of all destinations in the network. Simulation is performed for BFT and compared to mesh, torus, butterfly, and flattened butterfly topologies. Simulation results are shown in Figures 7, 8, 9, and 10. Comparisons are done against overall average latency, overall average acceptance rate, overall minimum acceptance rate, and average hop counts. Comparative improvements over other topologies are summarized in table I.

TABLE I
SUMMARY OF COMPARATIVE IMPROVEMENTS (%) OF PROPOSED NOC OVER OTHER TOPOLOGIES FOR DIFFERENT PERFORMANCE METRICS

Performance metrics → Topologies ↓	Avg Latency	Avg Acceptance Rate	Min Acceptance Rate	Avg Hop Count
Mesh	51-91	NIL	NIL	44
Torus	84-96	3-8	3-12	30
Butterfly	55-96	NIL	NIL	NIL
Flattened Butterfly	48-96	4-7	4-12	NIL
Previous BFT based design	19-78	2-42	0.2-0.6	20

V. Conclusion and Future Direction

Split Tree Architecture overcomes a major limitation that is imposed on any NoC design i.e. network performance degradation due to increasing scalability. Distribution of routers across different layers left the floor planning much more easier. Also extensive simulation proves that network efficiency significantly improved in terms of both communication latency and hop counts. Future enhancement of this design can flow in two major directions. One is fault tolerance, and the other is thermal efficiency. As all the regions across the chip that are situated at same relative positions are connected through DTDMA pillar, therefore, in case of fault situation, even a whole unit (a core layer and associated root and border layers) can be bypassed. Hence this design can be a robust one in case of faulty scenario where one or more portions of the chip is down. The design has an inherent thermal efficiency. In 3D design, generation of thermal hot spot is a major concern that occurs from 3D CPU core stacking and close proximity of the cores in a layer. In the present work, every pair of core layers have two additional layers (root and border layers) in between them. So even in case of stacking of processing cores the design is not vulnerable to generation of thermal hot spot. For uniform hopping characteristic in a core layer we can scatter the cores as much as possible without significant degradation in latency.

References

[1] D. Sylvester and K. Keutzer, "A Global Wiring Paradigm for Deep Submicron Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 19, pp. 242–252, February 2000.

[2] S. P. Mohanty, *Nanoelectronic Mixed-Signal System Design*. No. 978-0071825719 and 0071825711, McGraw-Hill Education, 2015.

[3] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, pp. 70–78, January 2002.

[4] C. Nicopoulos, V. Narayanan, and C. Das, *Network-on-Chip Architectures: A Holistic Design Exploration*, vol. 45 of *Lecture Notes in Electrical Engineering*. Springer Netherlands, September 2009.

[5] B. Feero and P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation," *IEEE Transactions on Computers*, vol. 58, pp. 32–45, January 2009.

[6] R. Kumar, V. Zyuban, and D. Tullsen, "Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads and Scaling," in *32nd International Symposium on Computer Architecture (ISCA '05)*, pp. 408–419, June 2005.

[7] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A Low Latency Router Supporting Adaptivity for On-chip Interconnects," in *42nd Annual Design Automation Conference (DAC)*, no. 6 in DAC '05, (New York, NY, USA), pp. 559–564, ACM, 2005.

[8] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. Yousif, and C. Das, "A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks," in *33rd International Symposium on Computer Architecture (ISCA '06)*, pp. 4–15, 2006.

[9] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express Virtual Channels: Towards the Ideal Interconnection Fabric," in *34th Annual International Symposium on Computer Architecture*, no. 12 in ISCA '07, (New York, NY, USA), pp. 150–161, ACM, 2007.

[10] R. Mullins, A. West, and S. Moore, "Low-latency Virtual-channel Routers for On-chip Networks," in *31st Annual International Symposium on Computer Architecture (ISCA '04)*, pp. 188–197, June 2004.

[11] W. Dally, "Express cubes: improving the performance of k-ary n -cube interconnection networks," *IEEE Transactions on Computers*, vol. 40, pp. 1016–1023, September 1991.

[12] J. Kim, J. Balfour, and W. Dally, "Flattened Butterfly Topology for On-Chip Networks," in *40th Annual IEEE/ACM International Symposium on Microarchitecture*, no. 11 in MICRO 40, (Washington, DC, USA), pp. 172–182, IEEE Computer Society, 2007.

[13] U. Ogras and R. Marculescu, ""It's a small world after all": NoC performance optimization via long-range link insertion," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 693–706, July 2006.

[14] P. Ghosal and T. S. Das, *Advances in Computing and Information Technology: Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India - Volume 3*, ch. A Novel Routing Algorithm for On-Chip Communication in NoC on Diametrical 2D Mesh Interconnection Architecture, pp. 667–676. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[15] P. Ghosal and T. S. Das, "Sd2d: A novel routing architecture for network-on-chip," in *Electronic System Design (ISED), 2012 International Symposium on*, pp. 221–225, Dec 2012.

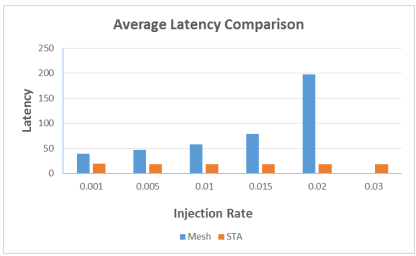
[16] P. Ghosal and T. S. Das, "L2star: A star type level-2 2d mesh architecture for noc," in *2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics*, pp. 1555–159, Dec 2012.

[17] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das, "A Novel Dimensionally-decomposed Router for On-chip Communication in 3D Architectures," in *34th Annual International Symposium on Computer Architecture*, ISCA '07, (New York, NY, USA), pp. 138–149, ACM, 2007.

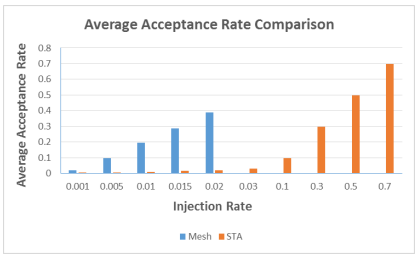
[18] D. Park, S. Eachempati, R. Das, A. Mishra, Y. Xie, N. Vijaykrishnan, and C. Das, "MIRA: A Multi-layered On-Chip Interconnect Router Architecture," in *35th International Symposium on Computer Architecture (ISCA '08)*, pp. 251–261, June 2008.

[19] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, "A Low-radix and Low-diameter 3D Interconnection Network Design," in *IEEE 15th International Symposium on High Performance Computer Architecture (HPCA 2009)*, pp. 30–42, February 2009.

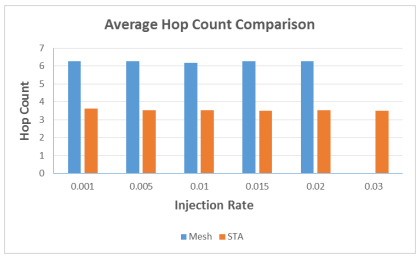
[20] A. Bose, P. Ghosal, and S. Mohanty, "A Low Latency Scalable 3D NoC Using BFT Topology with Table Based Uniform Routing," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 136–141, July 2014.



(a)

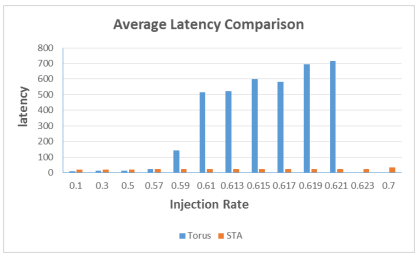


(b)

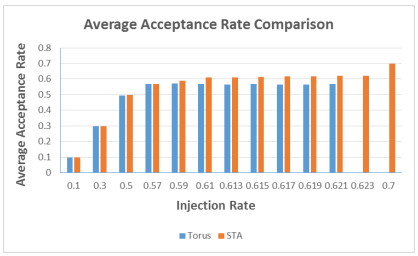


(c)

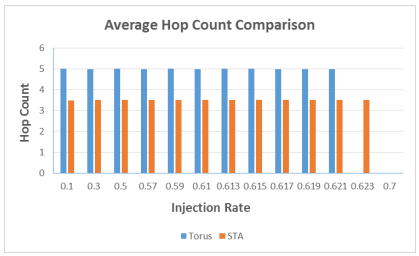
Fig. 7. Simulation Results for Mesh and STA (a) Overall average latency (b) Average acceptance rate (c) Average hop count.



(a)

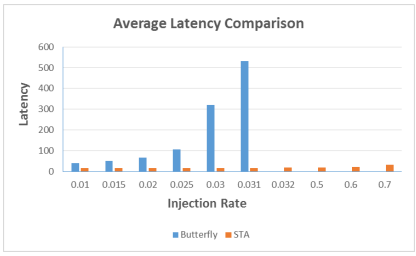


(b)

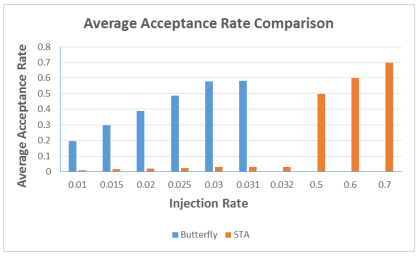


(c)

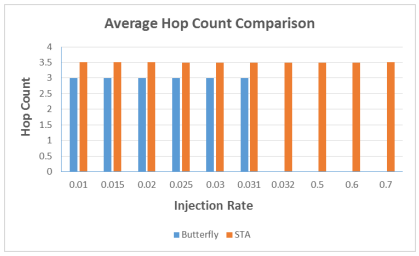
Fig. 8. Simulation Results for Torus and STA (a) Overall average latency (b) Average acceptance rate (c) Average hop count.



(a)

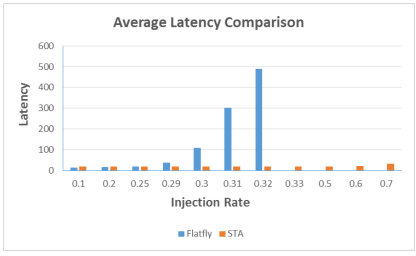


(b)

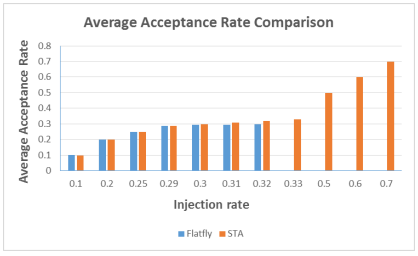


(c)

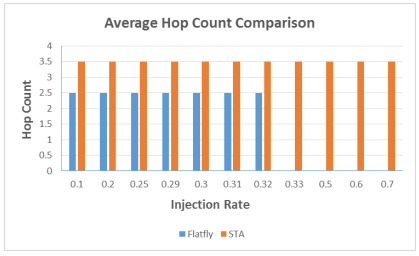
Fig. 9. Simulation Results for Butterfly and STA (a) Overall average latency (b) Average acceptance rate (c) Average hop count.



(a)



(b)



(c)

Fig. 10. Simulation Results for Flattened Butterfly and STA (a) Overall average latency (b) Average acceptance rate (c) Average hop count.