

A Quadrant-XYZ Routing Algorithm for 3-D Asymmetric Torus Network-on-Chip

Mohammad Ayoub Khan
Centre for Development of Advanced Computing
B-30, Sector 62, Institutional Area,
Noida, Uttar Pradesh, INDIA 201307
+91-120-3063371
ayoub@ieee.org

Abdul Quaiyum Ansari
Department of Electrical Engineering
Jamia Millia Islamia
New Delhi, INDIA 110025
+91-9873824597
aqansari@ieee.org

ABSTRACT

Three-Dimensional (3-D) ICs are able to obtain significant performance benefits over two-dimensional (2-D) ICs based on the electrical and mechanical properties resulting from the new geometrical arrangement. The arrangement of 3-D ICs also offers opportunities for new circuit architecture based on the geometric capacity. The emerging 3-D VLSI integration and process technologies allow the new design opportunities in 3-D Network-on-Chip (NoC). The 3-D NoC can reduce significant amount of wire length for local and global interconnects. In this paper, we have proposed an efficient 3-D Asymmetric Torus routing algorithm for NoC. The 3-D torus has constant node degree, recursive structure, simple communication algorithms, and good scalability. A Quadrant-XYZ dimension order routing algorithm is proposed to build 3-D Asymmetric Torus NoC router. The algorithm partitions the geometrical space into quadrants and selects the nearest wrap-around edge to connect the destination node. Thus, the presented algorithm guarantees minimal paths to each destination based on routing regulations. The complexity of the algorithm is $O(n)$. The proposed routing algorithm has been compared with the traditional XYZ algorithm and the comparison results show that the Quadrant-XYZ router has shorter path length. This paper presents a Register Transfer Logic (RTL) simulation model of Quadrant-XYZ dimension order routing algorithm for 3-D asymmetric torus NoC written in Verilog. The model represents the functional behavior of the routing chip down to the flit (byte) level. The 3-D asymmetric torus NoC has achieved a maximum operating frequency 750 MHz on Xilinx Vertex-6 programmable device.

Categories and Subject Descriptors

B.5.1 [Hardware]: Data-path design, B.6.1 [Hardware]: Combinational logic, B.7.1 [Integrated Circuit]: VLSI, C.2.1 [Network Architecture and Design]: Network Topology, C.2.2 [Network Protocols]: Routing protocol

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Asymmetric, 3-D Torus, NoC, RTL, IC, flit

1. INTRODUCTION

The Network-on-Chip (NoC) represents a new communication paradigm for increasingly complex on-chip networks. The NoC provides technique for generic on-chip interconnection network realized by routers that connects processing elements (PE) like ASICs, FPGAs, memories, IP cores etc. We have already learned about routing packets instead of wires [5, 6, 7]. Therefore, we will focus on packet switched network. The NoC offers flexibility, scalability, predictability, and higher bandwidth, low latency and provision for concurrent communications. To reduce the latency and wire length we need efficient interconnection architectures. The performance of an interconnection architecture depends on degree and diameter. The diameter of an interconnection network can be defined as the longest shortest path between two arbitrary nodes in the network. In general, if the degree increases, the diameter decreases, and otherwise. Thus, the cost of interconnection architecture can be defined as a value of the degree \times diameter. The larger value of the diameter means non-efficient interconnection because the traffic latency increases. Three dimensional integrated circuits offers low interconnect latency and area efficient solution for 3-D NoC [1, 2]. We have regular and irregular topology for placement and routing of IP cores. But, this has been observed that regular network topologies are well suited for realization of VLSI chips [25, 26]. Now, we classify the regular topologies with a viewpoint of the degree criterion, these are;

- a) The Mesh class that includes the torus that has fixed degree independent of number of nodes in the network.
- b) The Hypercube class and star-graph class in which the degree increases as the increase of the number of nodes [1].

Table 1. Node Degree Analysis [3]

No. of Processors						
Network Type	512	1024	2048	4096	8192	16384
n-cube Hypercube	9	10	11	12	13	14
Torus	4	4	4	4	4	4

The above analysis shows that the n-cube hypercube interconnection network is expensive and not suitable for on-chip communication architecture. In n-cube hypercube, node degree increases greatly as the system expansion takes place and thus increases the cost of network [12, 13].

From above discussion, we conclude that the torus network topology produces an optimum result for Network-on-Chip. Therefore, we need to exploit the characteristics of torus (mesh class) topology. Mesh class architecture is the most regular and simple architecture that is used in NoC designs. The implementation of mesh is simple and easy to understand. In mesh architecture every node is connected to four neighbors (except boundary nodes) as shown in figure 1. They need to communicate among each other for transfer of information. In N-dimensional mesh network, every node is connected to 2N of the neighboring node (except boundary nodes). Thus, the degree of a non-boundary node in an N-dimensional mesh is 2N. The number of physical connections per node remains fixed in a mesh network even if the size of the network increases. The performance of mesh network degrades due to phenomenal increase in the diameter. Therefore, the mesh class outperforms if the number of IPs on silicon are small.

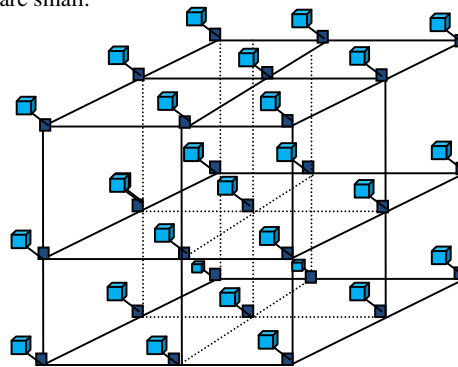


Figure 1. Interconnection in 3-D mesh topology

The diameter of 3-D mesh can be defined as $D = d(k - 1)$, where d represents dimension and k is the number of nodes in plane. A Torus network is same as mesh network with boundary nodes connected by wrap-around edges. These wrap-around edges significantly reduce the overall diameter of the network and thus improving the throughput and latency. Figure 2 shows 3-D torus architecture and partitioning approach into quadrants. The diameter of torus can be defined as follows:

$$D = d\left(\left\lfloor \frac{nx}{2} \right\rfloor + \left\lfloor \frac{ny}{2} \right\rfloor + \left\lfloor \frac{nz}{2} \right\rfloor\right), \text{ where } nx, ny, nz \text{ is the number of nodes in plane } x, y \text{ and } z \text{ respectively.}$$

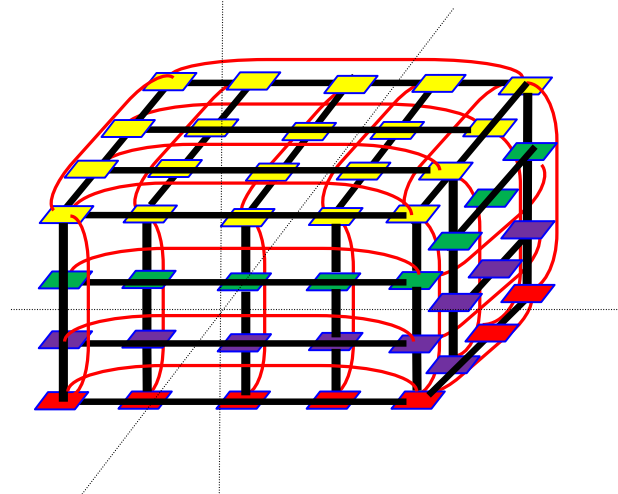


Figure 2. Interconnection in 3-D Torus topology

1.1 Motivation

The scales in VLSI technology to deep sub-micron (DSM) have started integrating large number of processing elements into silicon. The existing shared bus based communications are not able to achieve the required latency. This has raised the requirement of good communication architecture and topology. There is a lot of ongoing effort to design highly scalable communication and low-latency architectures. This work was in part motivated by our investigations of 3-D topologies for network-on-chip where we found scope to explore asymmetric network-on-chip structures. The routing strategy in 3-D considers routing at every layer apart from via interconnects. The architecture shown in figure 2 has asymmetric number of nodes in planes. There is large number of applications where every dimension has different number of processing elements, thus produces different diameter for each plane. In such scenario, the simple XYZ algorithm produces non-optimal shortest path. In the presented algorithm, we partition the torus space into quadrants and select the nearest wrap-around edge to connect the destination node. Thus, the presented algorithm guarantees minimal path to each destination based on simple routing regulations. The complexity of the algorithm is $O(n)$.

1.2 Applications

The Network-on-Chip is the latest research and development area in VLSI integration. The increasing system level integration has produced various types of applications. These applications have different traffic characteristics. The use of shared buses is becoming obsolete as they have high latency and large diameter. As a result, features of computer network in on-chip communications has emerged as NoC to establish data exchange within the chip. The future System-on-Chip (SoC) will contain billions of transistors, composing hundreds to thousands of IP cores. The SoC that implements complex multimedia, security, and network applications should be able to deliver the services in minimum amount of time. This needs an efficient cooperation and routing regulations among these IP cores. The topology and interconnection techniques has an important role in determining the routing efficiency for a set of applications. The 3-D offers a considerable reduction in the number and length of the wire. The quadrant-based approach has been proposed for such low latency applications. In addition to Network-on-Chip, the proposed algorithm has application in parallel massive computer networks.

1.3 Related Work

The NoC is derived from massive parallel computer networks and distributed computing. The routing technique in NoC has constraints on memory, computing resources, and routing techniques in addition to low latency and high throughput. There are several routers that have been developed for NoC [1, 2, 3, 4, 8, 9, 12, 27] employing XYZ routing algorithm for selection of next output channel. The routing technique used in [1, 2, 3, 4, 8, 9] doesn't acquire information about the nearest wrap-around edge. Thus, produces larger average distance. In order to avoid the congestion, the routing algorithm presented in [12] has an approach to balance move in each place. This essentially avoids the deadlock but has larger average distance. In paper [27], author has compared 3 x 3 mesh topology on XY and Odd-Even (OE) algorithm. The presented OE routing algorithm appears to be complex in implementation on the hardware. The author has claimed OE algorithm better over XY algorithm. This approach is also based on the existing regulation. In all the available routing technique [1, 2, 3, 4, 8, 9, 12, 27], authors has not discussed about the asymmetric structure of 3-D torus and regulation for wrap-around edges. The motivation of this work is to improve average distance and to maintain the simplicity of XYZ algorithm for asymmetric structure.

2. QUADRANT-XYZ Algorithm

The 3-D Mesh topology is developed with addition of z dimension to 2-D Mesh structures. Similarly, a 3-D Torus topology is build using a 3-D Mesh network that adds extra links at terminal nodes called wrap-around edges. Consequently, degree increases in both the 3-D networks as shown in table 2.

Table 2. Degree of 3-D Topology

Network	2D-Mesh	3D-Mesh	2D-Torus	3D-Torus
Degree	4	6	4	6

These topologies are regular and easy to implement from fabrication point of view. Commonly, Dimension-Ordered Routing (DOR) is used in mesh topologies. In this routing strategy, the route is determined in one dimension till the destination is reached in that dimension. While using DOR in torus topology, wrap-around edges provide some efficiency based on the location of source and destination node. The simple XYZ algorithm applied to torus doesn't guarantees minimal path. The proposed quadrant based approach always locates the nearest wrap-around edge to produce minimal path. In the next section, we present the proposed algorithm for 3-D torus topology.

2.1 Quadrant Approach

The torus space is divided into quadrants as shown in figure 3. The size of torus network is 5 x 6 x 3. Here, we can see that each plane has different number of nodes. In this approach, we find the center of each plane as follows:

$$X_0 = \left\lfloor \frac{nx}{2} \right\rfloor, Y_0 = \left\lfloor \frac{ny}{2} \right\rfloor, Z_0 = \left\lfloor \frac{nz}{2} \right\rfloor, \text{ where } nx, ny, nz \text{ is the number node in } x\text{-plane and similarly in } y \text{ and } z\text{-plane.}$$

The values for $X_0 = \left\lfloor \frac{5}{2} \right\rfloor$, $Y_0 = \left\lfloor \frac{6}{2} \right\rfloor$, $Z_0 = \left\lfloor \frac{3}{2} \right\rfloor$ is 2, 3 and 1 respectively. The source node has a coordinates (1, 2, 0) while destination node is located at (4, 2, 0). The values for Δx , Δy , Δz are 3, 0, and 0 respectively. This implies that we need to traverse only x-plane as the destination node is located on the same plane (y and z). In, XYZ routing algorithm the node will move in x-direction as shown in the figure 3. The XYZ routing algorithm produces a distance of 3 while our approach will produces a minimal distance of 2.

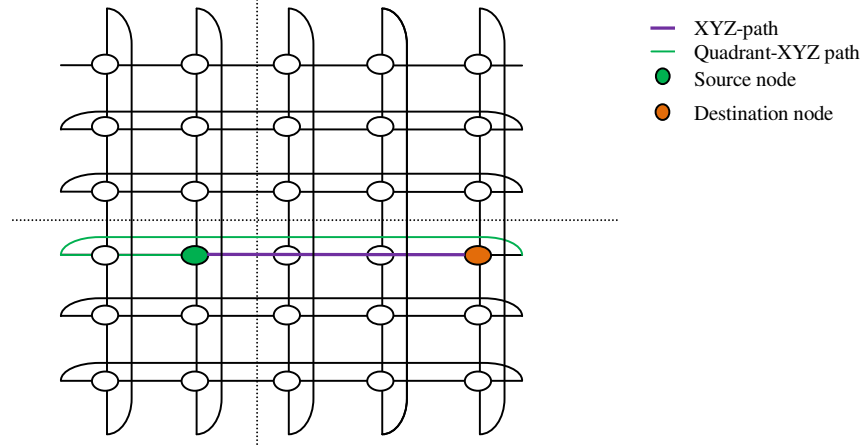


Figure 3. Quadrant-based 3-D Torus topology (Top View)

In quadrant-based approach, we first query two basic questions as follows:

1. Are the quadrants of source and destination nodes are different?
2. Is $\Delta x, \Delta y, \Delta z$ are greater than center X_0, Y_0, Z_0 respectively?

If above two queries returns true then we locate a nearest wrap-around edge to source node and we apply quadrant-based algorithm. In case, the above conditions are not true then we conclude that there is no advantage in applying quadrant-based approach. In such scenario, a simple XYZ-routing will be followed. A detailed description of algorithm is presented in the following section.

2.2 Proposed Algorithm

The quadrant-based algorithm introduces decision parameters and simple regulation to take the advantage of wrap-around edges. The simple regulation of quadrant-based algorithm helps in locating the nearest wrap-around edge. The algorithm guides the packet to move in forward or backward direction in the plane to locate the nearest wrap-around edge. The presented algorithm is generic, flexible and asymmetric. This novel asymmetric algorithm allows non-equal number of nodes in each plane. The complexity of algorithm is $O(n)$.

```

; setting up routing variable
Min = 0 , Max = n - 1      ; n is the number of nodes in a plane
 $\Delta x = X_d - X_c$       ; difference between source and destination in x-plane
 $\Delta y = Y_d - Y_c$       ; difference between source and destination in y-plane
 $\Delta z = Z_d - Z_c$       ; difference between source and destination in z-plane
 $X_0 = \lfloor \frac{nx}{2} \rfloor, Y_0 = \lfloor \frac{ny}{2} \rfloor, Z_0 = \lfloor \frac{nz}{2} \rfloor$       ; finding the center of each plane
 $X_n, Y_n, Z_n$           ; next destination node
; start of routing packets in X-plane
while ( $\Delta x \neq 0$ )      ; routing in X-plane till destination is found
{
    ; find out the nearest wrap-around edge in X-plane as destination is in the next quadrant
    If ( $\Delta x \in (X_0, Max] \parallel \Delta x \in [-Max, -X_0)$ )
        If ( $\Delta x > 0$ )
             $X_n = X_c - 1$  ; go to next node in west direction
        else
             $X_n = X_c + 1$  ; go to next node in east direction
    else
        ; simple X-routing
        If ( $\Delta x > 0$ )
             $X_n = X_c + 1$  ; go to next node in east direction
        else
             $X_n = X_c - 1$ ; go to next node in west direction
     $X_c = X_n$  ; now make next node as current node
     $\Delta x = X_d - X_c$ ; re-compute  $\Delta x$  from new source
}
; start of routing packets in Y-plane
while (( $\Delta x = 0$ ) && ( $\Delta y \neq 0$ ))
{
    ; find out the nearest wrap-around edge in Y-plane as destination is in the next quadrant
    If ( $\Delta y \in (Y_0, Max] \parallel \Delta y \in [-Max, -Y_0)$ )

```

```

        If ( $\Delta y > 0$ )
             $Y_n = Y_c - 1$  ; go to next node in south direction
        else
             $Y_n = Y_n + 1$  ; go to next node in north direction
    else ; simple Y-routing
        If ( $\Delta y > 0$ )
             $Y_n = Y_n + 1$  ; go to next node in north direction
        else
             $Y_n = Y_c - 1$  ; go to next node in south direction
     $Y_c = Y_n$  ; make next node as current node
     $\Delta y = Y_d - Y_c$  ; re-compute  $\Delta y$  from new source
}
; now routing the packets in Z-plane
while (( $\Delta x = 0$ ) && ( $\Delta y = 0$ ) && ( $\Delta z \neq 0$ ))
{
    ; find out the nearest wrap-around edge in Z-plane as destination is in the next quadrant
    If ( $\Delta z \in (Z_0, Max]$  ||  $\Delta z \in [-Max, -Z_0)$ )
        If ( $\Delta z > 0$ )
             $Z_n = Z_c - 1$  ; go to next node in down direction
        else
             $Z_n = Z_n + 1$  ; go to next node in up direction
    else ; simple Z-routing
        If ( $\Delta z > 0$ )
             $Z_n = Z_n + 1$  ; go to next node in up direction
        else
             $Z_n = Z_c - 1$  ; go to next node in down direction
     $Z_c = Z_n$  ; make next node as current node
     $\Delta z = Z_d - Z_c$  ; re-compute  $\Delta z$  from new source
}
    
```

In the next section we present simulation results and analysis on the hardware implementations.

3. SIMULATION RESULTS AND ANALYSIS

The proposed algorithm has been implemented on Field Programmable Gate Arrays (FPGA) using Verilog Register Transfer Logic language. The layout has been developed using Mentor Graphics EDA Tools (IC Station, Eldo, CaliberDRC etc). The architecture uses few logic gates, adder/subtractor, multiplexer, register and a controller to regulate the loop. The circuit accepts network size, source node, and destination node and produces route for next hop (node) to be traversed.

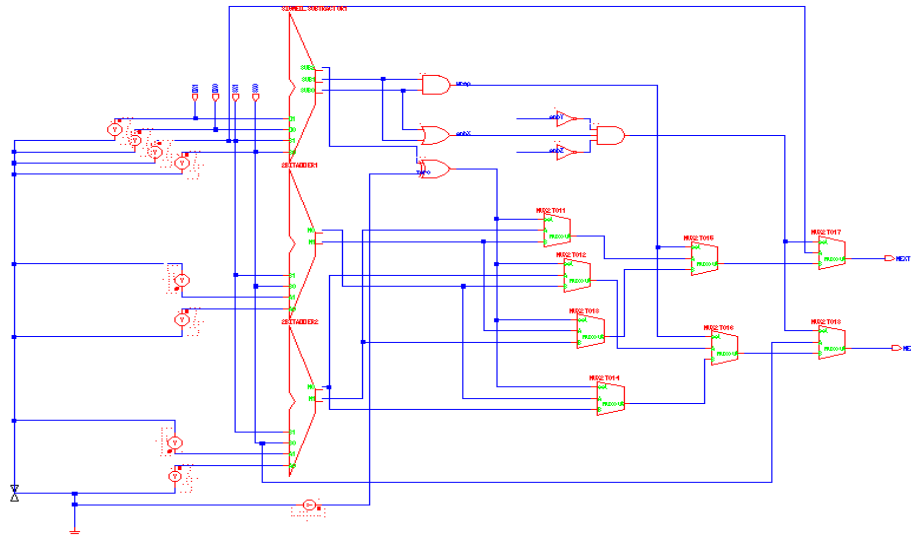


Figure 4. Schematic for Quadrant-XYZ routing algorithm

The schematic shows the circuit of proposed quadrant-XYZ routing algorithm. The schematic uses signed bit subtractor, adders and 2-to-1 multiplexers for each dimension. Signed subtractor gives the difference between source and destination nodes with the information whether the difference is positive or negative.

The synthesis of the 3-D Asymmetric Torus Network-on-Chip is targeted for Xilinx Vertex-6 device. The device has model 6VLX75TFF484. The circuit operates a maximum speed of 750 MHz. The FPGA device utilization for the synthesized design is shown in the following table.

Table 3. Device utilization summary

Resources	Used	Avail	Utilization (%)
IOs	4	240	1.66
Global Buffers	1	32	3.12
Fn. Generators	10	46560	0.02
CLBs Slices	20	11640	0.17
DDF/Latches	24	93120	0.03
Block RAMs	0	156	0
DSP48E1	0	288	0

The presented table 3 shows that the design has an optimum utilization of hardware resource on the programmable device. We have generated 128 test vectors for 4 x 4 x 8 asymmetric torus network. The exhaustive test vectors and cost analysis are shown in appendix A. We have found that proposed quadrant-based XYZ algorithm always produces guaranteed minimal path in all the cases.

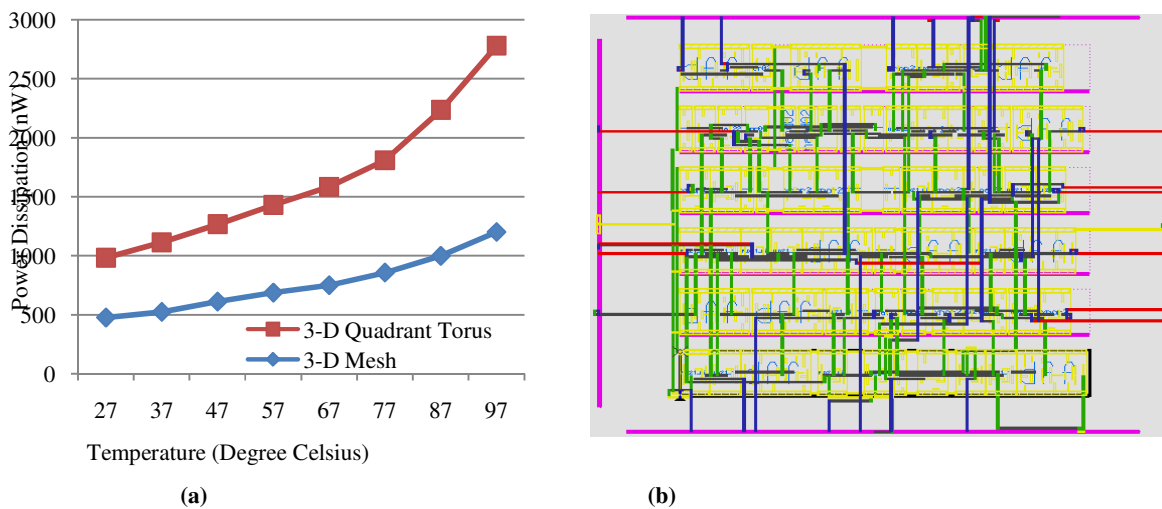


Figure 5. (a) Power Analysis (b) Layout of Quadrant-XYZ routing algorithm

The layout design has passed Design Rule Check (DRC) using TSMC 0.180 micron technology library in Mentor graphics EDA Tools (ICStation, Eldo, caliberDRC). To verify the functionality after net-list generation, a post-layout simulation was conducted and verified to be correct. The layout obtained is shown in figure 5(b). As mentioned earlier, 3-D torus has extra links (wires) for connecting wrap-around edges. Therefore, power dissipation will be more as compared to 3-D mesh. We have shown the power analysis in figure 5(a) at different temperature.

4. CONCLUSION

As we can see that 3-D torus has the minimum degree and diameter comparably with 3-D mesh network. This has least network cost and also from the VLSI realization point of view it is closer to the current technology. We also found that the existing XYZ algorithm is not suitable for 3-D torus in its current form. The XYZ algorithm doesn't guarantee a minimal path for 3-D torus network. Moreover, this has no routing regulation for wrap-around edges. We have presented an efficient algorithm that partitions the torus space into quadrants and select the nearest wrap-around edge to connect the destination node. Thus, the presented algorithm guarantees minimal paths to each destination based on routing regulations. The complexity of the algorithm is $O(n)$. The presented algorithm has been designed for 3-D asymmetric torus topology but this could be used efficiently for 3-D symmetric torus topology without any modification. We have presented RTL model and synthesis on Xilinx FPGA device and found to be optimal. The test case has been generated for 3-D asymmetric torus of 4 x 4 x 8. The functional verification shows the correctness of the proposed algorithm.

5. ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support received from University Grants Commission, Ministry of Human Resource Development, Govt. of India, during the course of this project under the Grant F. No. 39-895/2010(SR) to Department of Electrical Engineering, Jamia Millia Islamia (A central University), New Delhi, India

6. REFERENCES

- [1] Vitor de Paulo , CristinelAbabei, “3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans”, International Journal of Reconfigurable Computing, Hindawi, (2010), DOI= <http://dx.doi.org/10.1155/2010/603059>
- [2] B. Feero and P. Pande, “Performance evaluation for three-dimensional networks-on-chip,” in IEEE Computer Society Annual Symposium on VLSI, 2007. ISVLSI’07, (March 9-11, 2007), Porto Alegre, Brazil, 305–310, DOI=<http://dx.doi.org/10.1109/ISVLSI.2007.79>
- [3] Woo-seo Ki1, Hyeong-Ok Lee, Jae-Cheol Oh ,“The New Torus Network Design Based On 3-Dimensional Hypercube”, ICACT, pp. 615-620, Feb.15-18 2009.
- [4] Faiz AI Faisal and M.M. HafizurRahman ,“Symmetric Tori Connected Torus Network,“ 12th International Conference on Computer and Information Technology, (December 21-23, 2009), Dhaka, Bangladesh, 174-179, DOI=<http://dx.doi.org/10.1109/ICCIT.2009.5407144>
- [5] P. Guerrier and A. Grenier, “A generic architecture for on-chip packet switched interconnections,” in Proceedings of ACM/IEEE Design Automation and Test in Europe Conference (March 27-30 2000), 250–256, DOI=<http://dx.doi.org/10.1109/DATE.2000.840047>
- [6] A. Hemani, A. Jantsch, S. Kumar et al., “Network on chip: an architecture for billion transistor era,” in Proceedings of IEEE Conference, November 2000.
- [7] W. J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” in Proceedings of the 38th Design Automation Conference (DAC ’01), 684–689, June 2001 , DOI= <http://dx.doi.org/10.1109/DAC.2001.156225>
- [8] N. GopalakrishnaKini M. Sathish Kumar Mruthyunjaya H.S. , “A Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture,” 2009 IEEE International Advance Computing Conference (March 6-7 2009), Patiala, India, 858-861, DOI=<http://dx.doi.org/10.1109/IADCC.2009.4809127>
- [9] M.M. HafizurRahma and Susumu Horiguchi , “High Performance Hierarchical Torus Network under Matrix Transpose Traffic Patterns ,” Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN’04), China, DOI= <http://doi.ieeecomputersociety.org/10.1109/ISPAN.2004.1300467>
- [10] M.M. HafizurRahman, Xiaohong Jiang, Md. Shahin-AI Masud, and Susumu Horiguchi ,”Network Performance of Pruned Hierarchical Torus Network ,”Sixth IFIP International Conference on Network and Parallel Computing, (Oct 9-21, 2009), Gold Coast, Australia, 9-15, DOI=<http://dx.doi.org/10.1109/NPC.2009.11>
- [11] Philip H’olzespies, Erik Schepers, Wouter Bach, MischaJonker, Bart Sikkes, GerardSmit and Paul Havinga,” A Communication Model Based on an n-Dimensional Torus Architecture Using Deadlock-Free Wormhole Routing,” Proceedings of the Euromicro Symposium on Digital System Design (DSD’03), 166 - 172 , DOI=<http://dx.doi.org/10.1109/DSD.2003.1231920>
- [12] Jose Miguel Montañana, MichihiroKoibuchi, Hiroki Matsutani, Hideharu Amano, “Balanced Dimension-Order Routing for k-ary n-cubes”, (2009), Vienna, 499-506, DOI=<http://dx.doi.org/10.1109/ICPPW.2009.64>
- [13] EmadAbuelrub , “A Comparative Study on the Topological Properties of Hyper-Mesh Interconnection Network ,” Proceedings of World Congress on Engineering (2008), London, UK, Vol. 1 , 9-5
- [14] BehroozParhami , “Intorudction to parallel processing”, Springer(1999); 77, DOI=<http://doi.acm.org/10.1145/72551.72553>
- [15] G. Min and M. Ould-Khaoua, “A performance model for wormholeswitched interconnection networks under self-similar traffic,” IEEE Trans. Comput.(May 2004), vol. 53, no. 5,601–613, DOI=<http://dx.doi.org/10.1109/TC.2004.1275299>
- [16] S. Murali and G. De Micheli, “Bandwidth-constraint mapping of cores onto NoC architectures,” in Proc. Des. Autom. Test Eur.Conference(Feb. 2004), 896–901, DOI= <http://dx.doi.org/10.1109/DATE.2004.1269002>
- [17] L. Ni and P. Mckinley, “Survey of wormhole routing techniques in direct networks,” IEEE Trans. Computer (Feb. 1993), vol. 26, no. 2, 62–76, DOI= <http://dx.doi.org/10.1109/2.191995>
- [18] U. Y. Ogras and R. Marculescu, “Analytical router modeling for networks-on-chip performance analysis,” in Proc. Des. Autom. Test Eur. Conference (Apr. 2007), 1096–1101, DOI= <http://dx.doi.org/10.1109/DATE.2007.364440>
- [19] U. Y. Ogras and R. Marculescu, “It’s a small world after all NoC performance optimization via long-range link insertion,” IEEE Trans. Very Large Scale Integr. Syst.-Special Section Hardw./Softw. Codesign Syst. Synthesis(Jul. 2006) , vol. 14, no. 7, pp. 693–706, DOI=<http://dx.doi.org/10.1109/TVLSI.2006.878263>
- [20] K. Pawlikowski, “Steady-state simulation of queuing processes: A survey of problems and solutions,” ACM Comput. Surv.(Jun. 1990) , vol. 22, no. 2, 123–170, DOI= <http://dx.doi.org/10.1145/78919.78921>
- [21] L. S. Peh and W. J. Dally, “A delay model for router microarchitectures,” IEEE Micro, (2001), vol. 21, no. 1, 26–34, DOI=<http://dx.doi.org/10.1109/40.903059>
- [22] D. Rahmati, A. E. Kiasari, S. Hessabi, and H. Sarbazi-Azad, “A performance and power analysis of WK-recursive and mesh networks for network-on-chips,” in Proc. Int. Conf. Comp. Design (Oct. 2006), 142–147. DOI=<http://dx.doi.org/10.1109/ICCD.2006.4380807>

[23] H. Sarbazi-Azad, M. Ould-Khaoua, and L. Mackenzie, "An analytical modeling of wormhole routed k-ary n-cubes in the presence of hotspot traffic," IEEE Trans. Comput (Jul. 2001), vol. 50, no. 7, 623–634, DOI= <http://dx.doi.org/10.1109/12.936230>

[24] A. E. Kiasari, D. Rahmati, H. Sarbazi-Azad, and S. Hessabi, "A Markovian performance model for networks-on-chip," in Proc. Parallel Distrib. Network. Process (Feb. 2008), 157–164, DOI=<http://dx.doi.org/10.1109/PDP.2008.83>

[25] M. Kim, D. Kim, and E. Sobelman, "Network-on-chip link analysis under power and performance constraints," in Proc. IEEE Int. Symp. Circuits System (May 2006), 4163–4166. DOI= <http://dx.doi.org/10.1109/ISCAS.2006.1693546>

[26] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches," ACM Trans. Des. Autom. Electron. Syst., vol. 12, no. 3, pp. 1–20, Aug. 2007. DOI=<http://dx.doi.org/10.1145/1255456.1255460>

[27] Wang Zhang, LigangHou, Jinhui Wang, ShuqinGeng, Wuchen Wu "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip" Global Congress on Intelligent Systems. (GCIS 2009), China, 329 - 333 DOI=<http://dx.doi.org/10.1109/GCIS.2009.110>

Appendix A.

Comparison of Quadrant-based XYZ Algorithm and Simple XYZ over 128-Test Vector for Asymmetric Torus(4 x 4 x 8)

Case #	Source	Dest.	Cost (Quadrant, XYZ)	Case #	Source	Dest.	Cost (Quadrant, XYZ)	Case #	Source	Dest.	Cost (Quadrant, XYZ)
1	3, 3, 1	0,0,0	(3, 7)	44	3, 3, 1	1,1,3	(6, 6)	87	3, 3, 1	2,2,6	(5, 7)
2	3, 3, 1	0,0,1	(2, 6)	45	3, 3, 1	1,1,4	(7, 7)	88	3, 3, 1	2,2,7	(4, 8)
3	3, 3, 1	0,0,2	(3, 7)	46	3, 3, 1	1,1,5	(8, 8)	89	3, 3, 1	2,3,0	(2, 2)
4	3, 3, 1	0,0,3	(4, 8)	47	3, 3, 1	1,1,6	(7, 9)	90	3, 3, 1	2,3,1	(1, 1)
5	3, 3, 1	0,0,4	(5, 9)	48	3, 3, 1	1,1,7	(6, 10)	91	3, 3, 1	2,3,2	(2, 2)
6	3, 3, 1	0,0,5	(6, 10)	49	3, 3, 1	1,2,0	(4, 4)	92	3, 3, 1	2,3,3	(3, 3)
7	3, 3, 1	0,0,6	(5, 11)	50	3, 3, 1	1,2,1	(3, 3)	93	3, 3, 1	2,3,4	(4, 4)
8	3, 3, 1	0,0,7	(4, 12)	51	3, 3, 1	1,2,2	(4, 4)	94	3, 3, 1	2,3,5	(5, 5)
9	3, 3, 1	0,1,0	(4, 6)	52	3, 3, 1	1,2,3	(5, 5)	95	3, 3, 1	2,3,6	(4, 6)
10	3, 3, 1	0,1,1	(3, 5)	53	3, 3, 1	1,2,4	(6, 6)	96	3, 3, 1	2,3,7	(3, 7)
11	3, 3, 1	0,1,2	(4, 6)	54	3, 3, 1	1,2,5	(7, 7)	97	3, 3, 1	3,0,0	(2, 4)
12	3, 3, 1	0,1,3	(5, 7)	55	3, 3, 1	1,2,6	(6, 8)	98	3, 3, 1	3,0,1	(1, 3)
13	3, 3, 1	0,1,4	(6, 8)	56	3, 3, 1	1,2,7	(5, 9)	99	3, 3, 1	3,0,2	(2, 4)
14	3, 3, 1	0,1,5	(7, 9)	57	3, 3, 1	1,3,0	(3, 3)	100	3, 3, 1	3,0,3	(3, 5)
15	3, 3, 1	0,1,6	(6, 10)	58	3, 3, 1	1,3,1	(2, 2)	101	3, 3, 1	3,0,4	(4, 6)
16	3, 3, 1	0,1,7	(5, 11)	59	3, 3, 1	1,3,2	(3, 3)	102	3, 3, 1	3,0,5	(5, 7)
17	3, 3, 1	0,2,0	(3, 5)	60	3, 3, 1	1,3,3	(4, 4)	103	3, 3, 1	3,0,6	(4, 8)
18	3, 3, 1	0,2,1	(2, 4)	61	3, 3, 1	1,3,4	(5, 5)	104	3, 3, 1	3,0,7	(3, 9)
19	3, 3, 1	0,2,2	(3, 5)	62	3, 3, 1	1,3,5	(6, 6)	105	3, 3, 1	3,1,0	(3, 3)
20	3, 3, 1	0,2,3	(4, 6)	63	3, 3, 1	1,3,6	(5, 7)	106	3, 3, 1	3,1,1	(2, 2)
21	3, 3, 1	0,2,4	(5, 7)	64	3, 3, 1	1,3,7	(4, 8)	107	3, 3, 1	3,1,2	(3, 3)
22	3, 3, 1	0,2,5	(6, 8)	65	3, 3, 1	2,0,0	(3, 5)	108	3, 3, 1	3,1,3	(4, 4)
23	3, 3, 1	0,2,6	(5, 9)	66	3, 3, 1	2,0,1	(2, 4)	109	3, 3, 1	3,1,4	(5, 5)
24	3, 3, 1	0,2,7	(4, 10)	67	3, 3, 1	2,0,2	(3, 5)	110	3, 3, 1	3,1,5	(6, 6)
25	3, 3, 1	0,3,0	(2, 4)	68	3, 3, 1	2,0,3	(4, 6)	111	3, 3, 1	3,1,6	(5, 7)
26	3, 3, 1	0,3,1	(1, 3)	69	3, 3, 1	2,0,4	(5, 7)	112	3, 3, 1	3,1,7	(4, 8)
27	3, 3, 1	0,3,2	(2, 4)	70	3, 3, 1	2,0,5	(6, 8)	113	3, 3, 1	3,2,0	(2, 2)
28	3, 3, 1	0,3,3	(3, 5)	71	3, 3, 1	2,0,6	(5, 9)	114	3, 3, 1	3,2,1	(1, 1)
29	3, 3, 1	0,3,4	(4, 6)	72	3, 3, 1	2,0,7	(4, 10)	115	3, 3, 1	3,2,2	(2, 2)
30	3, 3, 1	0,3,5	(5, 7)	73	3, 3, 1	2,1,0	(4, 4)	116	3, 3, 1	3,2,3	(3, 3)
31	3, 3, 1	0,3,6	(4, 8)	74	3, 3, 1	2,1,1	(3, 3)	117	3, 3, 1	3,2,4	(4, 4)
32	3, 3, 1	0,3,7	(3, 9)	75	3, 3, 1	2,1,2	(4, 4)	118	3, 3, 1	3,2,5	(5, 5)
33	3, 3, 1	1,0,0	(4, 6)	76	3, 3, 1	2,1,3	(5, 5)	119	3, 3, 1	3,2,6	(4, 6)
34	3, 3, 1	1,0,1	(3, 5)	77	3, 3, 1	2,1,4	(6, 6)	120	3, 3, 1	3,2,7	(3, 7)
35	3, 3, 1	1,0,2	(4, 6)	78	3, 3, 1	2,1,5	(7, 7)	121	3, 3, 1	3,3,0	(1, 1)
36	3, 3, 1	1,0,3	(5, 7)	79	3, 3, 1	2,1,6	(6, 8)	122	3, 3, 1	3,3,1	(0, 0)
37	3, 3, 1	1,0,4	(6, 8)	80	3, 3, 1	2,1,7	(5, 9)	123	3, 3, 1	3,3,2	(1, 1)
38	3, 3, 1	1,0,5	(7, 9)	81	3, 3, 1	2,2,0	(3, 3)	124	3, 3, 1	3,3,3	(2, 2)
39	3, 3, 1	1,0,6	(6, 10)	82	3, 3, 1	2,2,1	(2, 2)	125	3, 3, 1	3,3,4	(3, 3)
40	3, 3, 1	1,0,7	(5, 11)	83	3, 3, 1	2,2,2	(3, 3)	126	3, 3, 1	3,3,5	(4, 4)
41	3, 3, 1	1,1,0	(5, 5)	84	3, 3, 1	2,2,3	(4, 4)	127	3, 3, 1	3,3,6	(3, 5)

42	3, 3, 1	1,1,1	(4, 4)	85	3, 3, 1	2,2,4	(5, 5)	128	3, 3, 1	3,3,7	(2, 6)
43	3, 3, 1	1,1,2	(5, 5)	86	3, 3, 1	2,2,5	(6, 6)				