

Cluster Based Hierarchical Routing Algorithm for Network on Chip

U. Saravanakumar¹, R. Rangarajan², R. Haripriya², R. Nithya², K. Rajasekar¹

¹Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, India

²Department of Electronics and Communication Engineering, Indus College of Engineering, Coimbatore, India

Email: saran.usk@gmail.com, profrr@gmail.com

Received July 12, 2013; revised August 12, 2013; accepted August 19, 2013

Copyright © 2013 U. Saravanakumar *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This paper presents a new logical mechanism called as Cluster Based Hierarchical Routing (CBHR) to improve the efficiency of NoC. This algorithm comprises the following steps: 1) the network is segmented logically into clusters with same size or different sizes; 2) algorithms are assigned for internal and global routing; 3) routers working functions are modified logically to support local and global communication. The experiments have conducted for CBHR algorithm for two dimensional mesh and torus architectures. The performance of this mechanism is analyzed and compared with other deterministic and adaptive routing algorithms in terms of energy, throughput with different packet injection ratios.

Keywords: System on Chip; Network on Chip; Deterministic and Adaptive Routing Algorithms; Mesh; Torus

1. Introduction

A number of processors in bus based System on Chip (SoC) are increased continuously and they face design challenges in different aspects [1]. This bus architecture has faced bottleneck problem when more processors integrated into single chip. To avoid bottleneck, bus architecture is replaced with network architecture which is similar to the data networks. This new technology is known as Network on Chip (NoC) and it is widely accepted as a solution for communication issues in SoC. Data communication between the processors is packetized and transmitted throughout the entire network [2, 3]. The basic components of NoC are processors, memories, routers and physical links. All the processors, memory blocks and other cores are connected to routers using physical links. The routers are interconnected to each other directly or through other intermediate routers. The role of router is to make decision where the data is to be transmitted based on destination address in the header flit of message packet [4-6]. A routing algorithm plays a major role in NoC that helps to communicate one processor to other processors or memory. This paper presents different routing algorithms such as XY—routing algorithm, OE—turn routing algorithm, and Pseudo adaptive routing algorithm. Additionally new algorithm has also proposed in this paper to achieve better performance for different NoC architectures. These routing algorithms

have implemented on different NoC architectures such as two Dimensional Mesh and Torus.

The rest of this paper is organized as follows: In Section 2, NoC architectures are described; the deterministic and adaptive routing algorithms and proposed CBHR algorithms are explained in Sections 3 and 4 respectively; Sections 5 deals with experimental results and discussions of the algorithms.

2. NoC Architectures

The different network topologies like mesh, ring, star, and torus are used in MPSoCs to overcome the communication issues. Additionally, some hybrid topologies have also proposed by VLSI designers especially for multi-processor SoCs. This section deals with popular network architectures.

A mesh-shaped network consists of m columns and n rows. The 2D mesh architecture is shown in **Figure 1(a)** which consists of 16 Processing Elements arranged in 4×4 matrix structure. The routers are situated in the intersections of two wires and the computational resources are near routers. Addresses of routers and resources can be easily defined as (x, y) coordinates in mesh. Regular mesh network is also called as Manhattan Street network. A Torus network is an improvement of basic mesh network. A simple torus network is a mesh in which the heads of the columns are connected to the tails of the

columns and the left sides of the rows are connected to the right sides of the rows [7].

Torus network has better path diversity than mesh network, and it has more minimal routes. Torus architecture is shown in **Figure 1(b)**. Torus is same as regular mesh except additional links in every row and column (red colour links in **Figure 1(b)**). In mesh, edge switches are connected only to two neighboring switches, the torus architecture uses wrap-around channels in order to connect the switches at the edges to the switches at the opposite edge. The number of switches is equal to the number of IP blocks and every switch has five ports. Due to the long wrap-around channels the packet transmission delay may become significantly long and require usage of repeaters. Folding is done by shifting all nodes in even rows to the right and all nodes in even positions of each row down, next connecting all the neighbouring nodes in newly gained rows and columns then pair-wise connecting edge nodes in rows and columns. The wraparound links are significantly shorter and link propagation delays fit within a single clock cycle [8].

3. Deterministic and Adaptive Routing Algorithms

3.1. Deterministic Algorithm

The traditional XY routing algorithm for multi-computers is first proposed by Intel Corporation and adapted for NoC by Wang Zhang *et al.* The working function of XY and YX routing algorithm is described in **Figure 2**. In XY algorithm is performed the routing function by increasing the x coordinate value of routers in the network from source until reaching the column of destination router and then start to increase the value of y coordinate until reaching destination router. After reaching the destination router, the message forwards to corresponding processor. In case of YX routing algorithm, the packet is routed by increasing y coordinate value first and increasing x coordinate value next. The XY/YX algorithm is simple to implement in any kind of network those are having mesh structure [9-11].

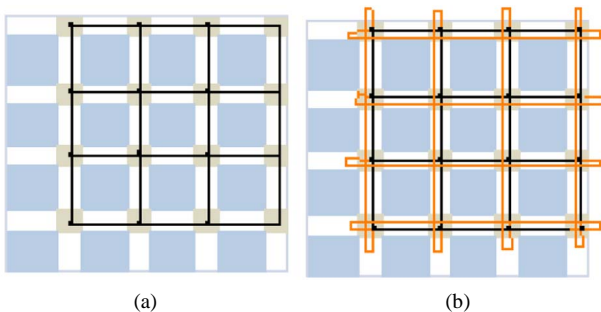


Figure 1. NoC Architectures: (a) 2D Mesh; (b) Torus.

3.2. Adaptive Routing Algorithm

The Odd-Even turn algorithm is proposed by Chiu for two dimensional mesh networks with no virtual channels. **Figure 3** shows the possible routing path based on adaptive algorithm. It is a kind of distributed adaptive routing algorithm and the main advantage of this algorithm is deadlock free by restricting some of the turns. This algorithm is also suitable for torus network. In a two-dimension mesh with dimensions $n \times m$ each node is identified by its coordinate (x, y) . A column is called even if its x dimension element is even numerical column and odd if its x dimension element is an odd number. A turn is a 90-degree turn in the following description. There are eight types of turns, according to the travelling directions of the associated I/O ports. A turn is called an ES turn if it involves a change of direction from East port to South port. Similarly, other seven types of turns are defined as, EN, WS, WN, SE, SW, NE, and NW turns, where E, W, S, and N indicate East, West, South, and North, respectively [12,13].

The OE turn algorithm performs the routing function based on two conditions and they are described in the **Figure 4**.

Condition 1: Any packet is not allowed to take a turn from East to North at any routers located in an even column, and it is not allowed to take a turn from North to West at any nodes located in an odd column

Condition 2: Any packet is not allowed to take a turn from East to South at any routers located in an even column, and it is not allowed to take a turn from South to West at any nodes located in an odd column

The adaptive odd—even turn routing algorithm is more complex than classic XY routing algorithm but it provides deadlock free condition.

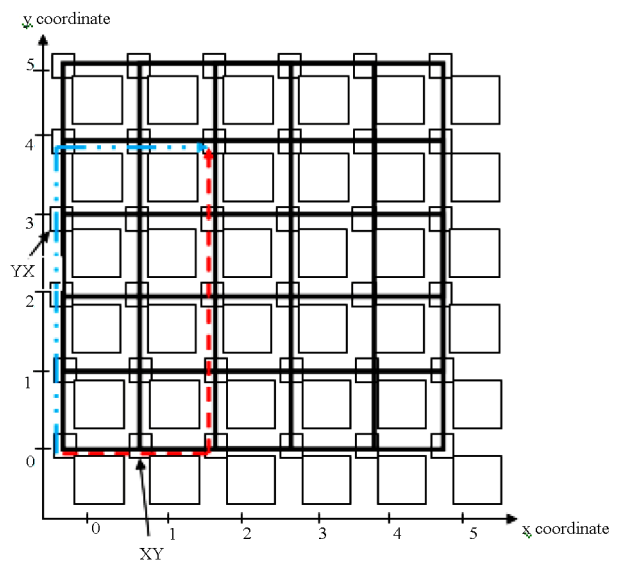


Figure 2. Deterministic algorithms for mesh NoC.

3.3. Pseudo Adaptive Routing Algorithm

The pseudo adaptive routing algorithm is proposed by Dehyadgari *et al.* This algorithm is developed based on both deterministic and adaptive approach with respect to the network load. If the network load is low, the packet is routed using classic XY routing algorithm (deterministic) else the packet is routed using adaptive mode. The congestion in the routing path can be identified by setting the

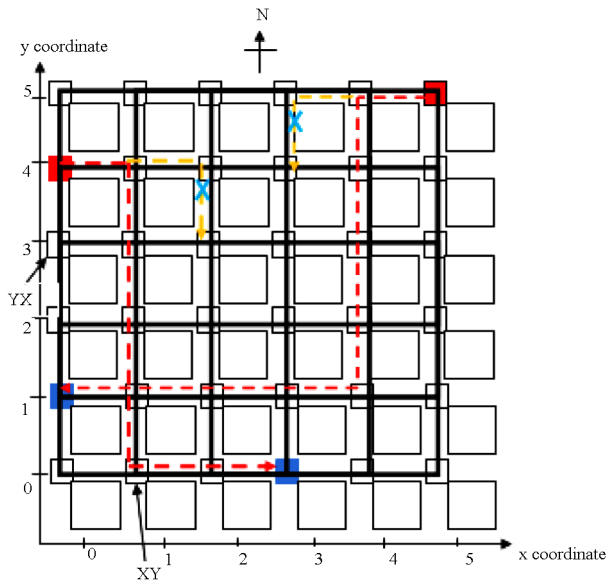


Figure 3. Adaptive routing algorithm for mesh NoC.

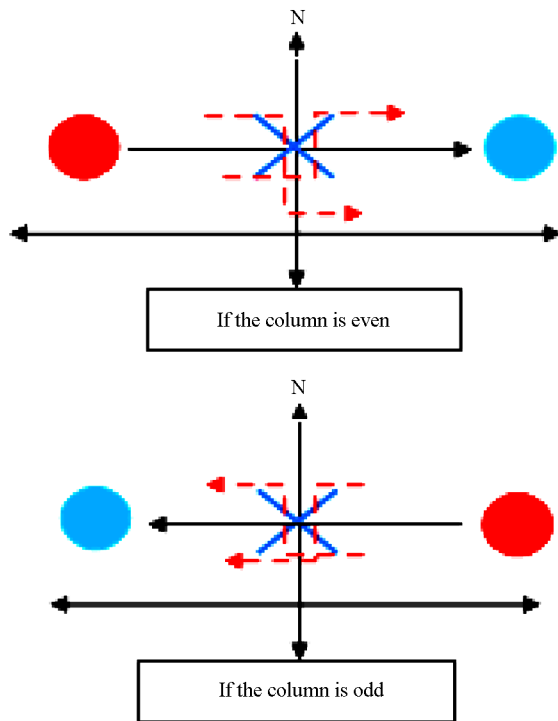


Figure 4. Conditions to perform adaptive algorithm.

threshold level for input buffer in the router. The threshold value is fixed as 100% free (ready to receive a data), 75% free (assume buffer is loading with data), 50% free (assume buffer is full) and 100% busy (can't ready to receive a data). Based on the threshold value, the router decides the port where the data to be routed. This algorithm offers possible paths from source to destination with low traffic load before receiving the status of heavy traffic [9].

4. CBHR Algorithm

The deadlock free is the main concept in any network and different routing algorithms have proposed to achieve the same. One of the method is, hierarchical routing scheme proposed by Holsmark *et al.* In their work, each subnet works perform the routing function using internal routing algorithm and each subnets are interconnected with global routing algorithm. In this chapter, a cluster based hierarchical routing logic is introduced. The entire network is divided into several clusters logically and its size can be varied depends on network size as shown in **Figure 5**. The clusters in the network are standalone network and the routers do not bother about other clusters [14].

In this work, the two different network sizes 4×4 and 8×8 are considered. For 4×4 network architecture, it is divided into four clusters and each having four routers. For 8×8 network architecture, it is divided into four cluster with 16 routers or 16 clusters with 4 routers.

4.1. CBHR Protocol

The packets are routed to the destination in the same cluster or other clusters in the network using internal

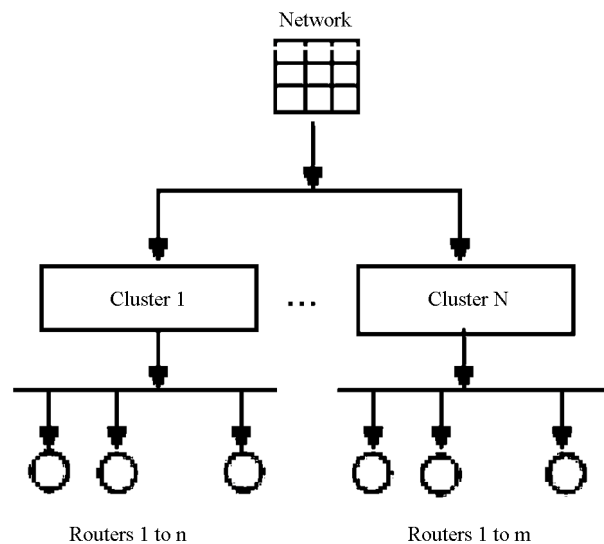


Figure 5. CBHR concept in NoC architectures.

cluster or other clusters in the network using internal routing or external routing algorithm. If the destination address is in the same cluster, internal routing can be done by tagging the additional information of *cluster id* and *destination router id*. If the destination address is in the different cluster, the routing can be done through boundary nodes and some additional information of *cluster id*, *boundary router id* and *destination router address*. **Figure 6** shows the dedicated packet format for CBHR based NoC.

4.2. Routing Function

In the CBHR, the routing function first takes the details of header flit in the packet which contains the destination router address and cluster id if the destination in the same

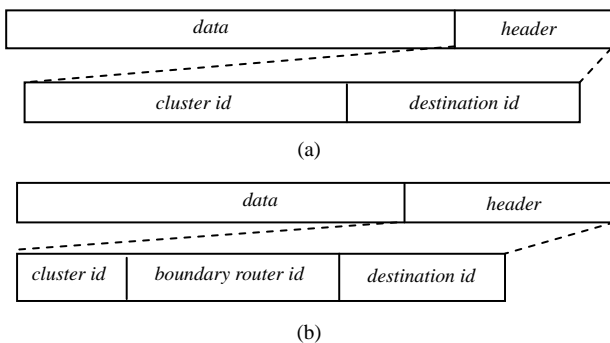


Figure 6. Packet format if (a) destination in same cluster (b) destination address in different cluster.

cluster or cluster id, boundary router id and destination router id if the destination router in the different network. Consider the case 1: from the header flit information, if both the cluster id and router address are equal then the corresponding port is set to the local PE. Otherwise, internal routing function (in this case XY for domain 1 which consist of clusters 1 and 3 or OE for domain 2 which consist of clusters 2 and 4) is called with *destination router address*. Consider the case 2: if the cluster ids are different, the external routing function (in this case pseudo adaptive is selected in order to identify the network load and avoid congestion) is invoked with *cluster id* and *boundary router id*. The working functions of two cases are clearly described in **Figure 7**. The boundary routers are designed using logical concept to adopt both internal and external routing algorithms.

The router to support CBHR is designed with two additional concepts, one is comparator to compare the destination address and current address with cluster ids and another one is multiplexer to select routing function to be done whether it is internal or global. The routers in the boundary regions are designed with threshold value as discussed earlier to identify the congestion in the network. As like in mesh NoC, the CBHR algorithm is applied on torus NoC but the cluster size has varied. In **Figure 8**, the brown colour routers are identified as boundary routers, that are support both deterministic and adaptive algorithm. Here also, the classic XY algorithm and OE algorithm are used for local routing in cluster 1 and 2 respectively. The pseudo adaptive algorithm is

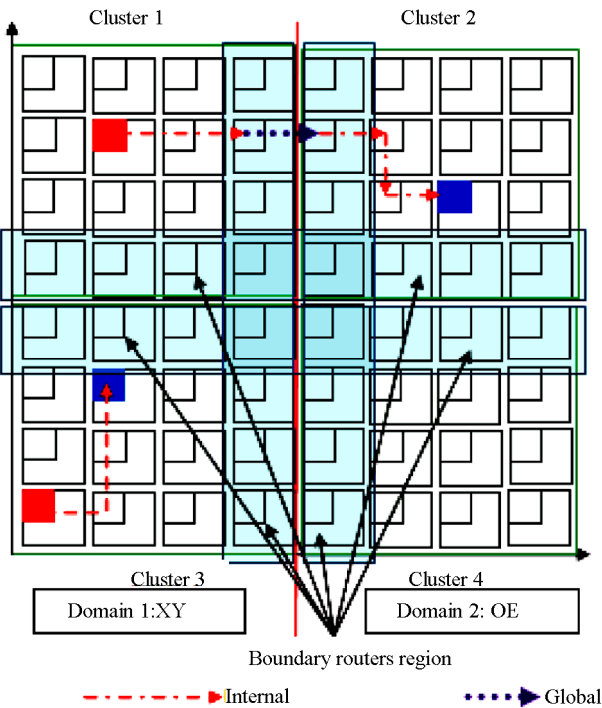


Figure 7. CBHR for mesh NoC.

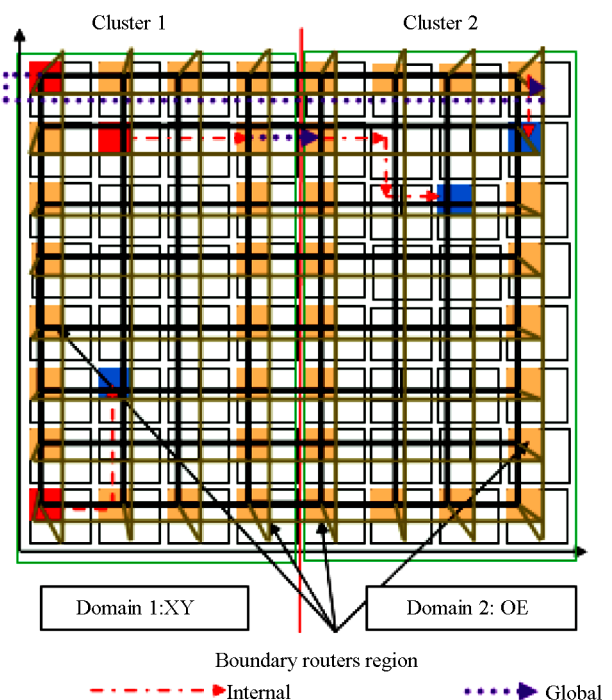


Figure 8. CBHR for torus NoC.

used as a global routing mechanism for data transmission from one cluster to another cluster.

5. Simulation Results and Discussions

The two dimensional mesh and torus architectures with 16 and 64 PEs are considered for the evaluation of XY/YX, OE and CBHR algorithms. These algorithms are simulated in Network Simulator under Linux environment. To understand the effective of routing algorithms on NoCs, throughput and energy are assumed as evaluation metrics. The experiments are conducted for routing algorithms with two different packet sizes 210 and 512 in 60 seconds' simulation time. For simulation purpose, the distance between source router and destination routers are considered in three categories named as minimum (Number of links in 16 PEs: 1 and 64 PEs: 1), moderate (Number of links in 16 PEs: 4 and 64 PEs: 7) and maximum (Number of links in 16 PEs: 6 and 64 PEs: 14).

Finally the performances of these three algorithms are compared for mesh and torus NoC architectures with different packet sizes. The energy consumption for data transmission from source to destination using XY, OE and CBHR algorithms are described in Figure 9 and the values are listed in Table 1 under packet size of 210 and 512. For both mesh and torus NoC architectures, the CBHR algorithm works very well. The CBHR algorithm consumes the energy of 36.345 J and 28.881 J to transmit

za packet size of 512 for maximum distance range in mesh NoC and torus NoC respectively. Therefore, the CBHR algorithm consumes 16.8% less energy than OE algorithm and 11.1% less than classic XY routing algorithm for mesh NoCs. For torus NoC, the CBHR algorithm consumes less energy 22.7% than OE and 15.9% than classic XY routing algorithm. The throughput of algorithms for 8 × 8 mesh and torus NoCs with 210 and 512 packet sizes are described in Table 2. From the Table 2, the throughput of the CBHR algorithm on torus NoC is higher than other algorithms on different NoC architectures.

The throughput of routing algorithms for NoC in terms of Kbps is compared in Figure 10 with 512 as a packet size. As a result, the new logic CBHR on torus NoC is more efficient than XY and OE algorithms on mesh and torus NoC architectures. But this new algorithm performs very well in NoC with more number of processors than NoC with less count of processors.

The simulation results show that, there is no major improvement in 16 PEs NoC but it shows better result in 64 PEs. The estimation from ITRS in 2011, the number of processors is increased gradually. For those cases, this type of hybrid routing logics helps to improve the performance of NoC architectures.

6. Conclusion

In this paper, both deterministic and adaptive routing algorithms have been discussed for NoC architectures. A new logical approach called Cluster Based Hierarchical

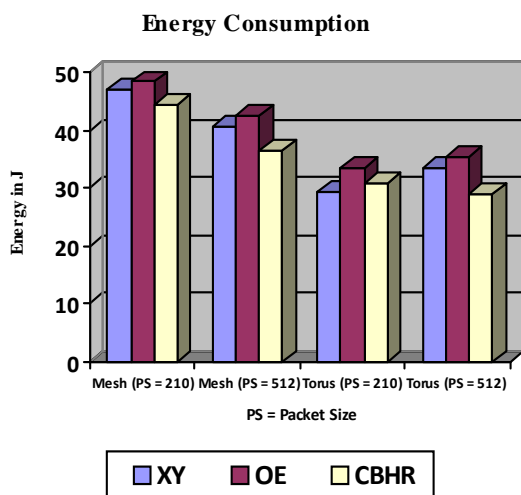


Figure 9. Energy comparison of routing algorithms with 210 and 512 packet sizes.

Table 1. Energy consumption of routing algorithms.

PS	Energy Consumption (J)					
	8 × 8 Mesh			8 × 8 Torus		
	XY	OE	CBHR	XY	OE	CBHR
210	46.9	48.2	44.23	29.3	33.5	30.9
512	40.5	42.4	36.3	33.5	35.4	28.9

Table 2. Throughput of routing algorithms for mesh and torus NoC.

PS	Throughput (packets/60 seconds)					
	8 × 8 Mesh			8 × 8 Torus		
	XY	OE	CBHR	XY	OE	CBHR
210	4737.8	4933.03	6214.1	4875.4	5239.1	6712.4
512	4199.1	4649.06	5673.4	4307.3	4927.2	5990.2

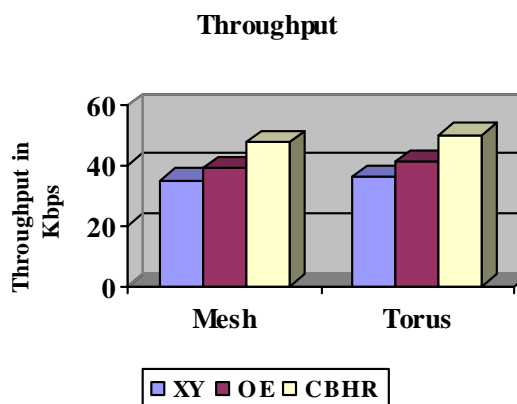


Figure 10. Throughput comparison.

Routing (CBHR) algorithm for NoC is introduced. To evaluate the performance of routing algorithms, two different NoC architectures such as two dimensional mesh and torus are considered with various sizes. This CBHR helps to reduce the energy consumption considerably in torus architecture. The simulation results also show that, throughput is increased for NoC with more processors.

REFERENCES

- [1] ITRS, International Technology Roadmap for Semiconductors, 2012.
<http://www.itrs.net/Links/2012ITRS/Home2012.htm>
- [2] T. N. Theis, "The Future of Interconnection Technology," *IBM Journal of Research and Development*, Vol. 44, No. 3, 2000, pp. 379-390. [doi:10.1147/rd.443.0379](https://doi.org/10.1147/rd.443.0379)
- [3] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," *Proceedings of the 38th Design Automation Conference (DAC)*, Las Vegas, 18-22 June 2001.
- [4] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys*, Vol. 38, No. 1, 2006.
[doi:10.1145/1132952.1132953](https://doi.org/10.1145/1132952.1132953)
- [5] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *Computer*, Vol. 35, No. 1, 2002, pp. 70-78. [doi:10.1109/2.976921](https://doi.org/10.1109/2.976921)
- [6] J. Nurmi, "Network-on-Chip: A New Paradigm for System-on-Chip Design," *Proceedings of International Symposium on System-on-Chip*, Tampere, 17 November 2005, pp. 2-6.
- [7] J. Henkel, W. Wolf and S. T. Chakradhar, "On-Chip Networks: A Scalable, Communicationcentric Embedded System Design Paradigm," *Proceedings of 17th International Conference on VLSI Design*, Mumbai, 5-9 January 2004, p. 845. [doi:10.1109/ICVD.2004.1261037](https://doi.org/10.1109/ICVD.2004.1261037)
- [8] W. J. Dally and C. L. Seitz, "Torus Routing Chip," *Distributed computing*, Vol. 1, No. 4, 1986, pp. 187-196.
[doi:10.1007/BF01660031](https://doi.org/10.1007/BF01660031)
- [9] M. Dehyadgari, M. Nickray, A. Afzali-kusha and Z. Navabi, "Evaluation of Pseudo Adaptive XY Routing Using an Object Oriented Model for NOC," *Proceedings of the 17th International Conference on Microelectronics*, Islamabad, 13-15 December 2005.
- [10] Intel Corporation, "A Touchstone DELTA System Description," Intel Corporation, Portland, 1991.
- [11] S. Lillevik, "The Touchstone 30 Gigaflop DELTA Prototype," *Proceedings of 6th Distributed Memory Computing Conference*, Portland, 28 April-1 May 1991.
- [12] G.-M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 7, 2000, pp. 729-738.
[doi:10.1109/71.877831](https://doi.org/10.1109/71.877831)
- [13] W. Zhang, L. Hou, J. Wang, S. Geng and W. Wu, "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip," *Proceedings of the WRI Global Congress on Intelligent Systems (GCIS '09)*, Washington DC, 19-21 May 2009, pp. 329-333.
[doi:10.1109/GCIS.2009.110](https://doi.org/10.1109/GCIS.2009.110)
- [14] R. Holsmark, S. Kumar, M. Palesi and A. Mejia, "HiRA: A Methodology for Deadlock Free Routing in Hierarchical Networks on Chip," *Proceedings of 3rd ACM/IEEE International Symposium on Networks-on-Chip*, San Diego, 10-13 May 2009, pp. 2-11.