LBDR3D: Fault Tolerant Routing scheme for 3D NoCs

Priyanka Mitra*
*Department of Computer Science and Engineering
Malaviya National Institute of Technology,
Jaipur, Rajasthan, India, 302017
priyanka.mitra.vit@gmail.com

***Abstract*—The reliable and scalable design for Network on chips (NoCs) usually uses mesh topologies. But the increase of the number of cores integration on a single chip increases network diameter and introduces power and area constraints in designing system-on-chips. Thus 3D NoCs an alternating interconnect technology has been introduced which sustains growth and performance with increased cores integration. But 3D NoCs faces fault issues due to fabrication defects, component failures, power saving schemes, heterogeneous cores and technology integration on different layers which may lead to irregular topologies. Thus to design efficient routing algorithms for such irregular 3D NoCs becomes a challenge. This paper provides Logic Based Distributed Routing for 3D NoCs (LBDR3D) which extend the capabilities of Logic Based Distributed Routing of 2D NoCs for handling faults in 3D NoCs. The Logic Based Distributed Routing is topology agnostic in nature and works efficiently for handling faults in 2D mesh topology but fails to work in 3D NoCs. The paper provides a new circuit LBDR3D that provides routing implementation and eliminates the need of routing tables for routing and handling faults in regular 3D mesh NoCs. Experimental results show that LBDR3D mimic the performance of 3D NoCs routing algorithms and provide fault tolerance without any routing tables.**

***Index Terms*—interconnect; distributed; fault; routing; bits**

I. Introduction

The increasing demand for integrating large number of cores on a single chip has paved the way for designing interconnect fabric. Network on chips (NoCs) has provided the promising solution to provide communication infrastructure to multicore systems. But the traditional interconnects face the challenge of increased power and area consumption with increased wire delay. To overcome the bottleneck of wired 2D mesh network of large number of cores, 3D NoCs have been designed where multiple silicon layers are vertically stacked and connected through die-to-die high speed interconnects [1] such as Through Silicon Vias (TSVs).

3D NoCs have combined the advantage of NoCs and 3D ICs [2] to reduce the average wire length in a mesh network and thus provided high performance solution with reduced wire delay and power consumption. But NoC interconnects running with GHz clock frequency and designed with deep submicron semiconductor technology are prone to failures [3]. Although TSVs offer faster, shorter and power efficient interconnect than horizontal wired links but it carries issues of poor yield, traffic balancing and thermal challenges. Moreover TSVs [4] fabrication leads to low reliability due to misalignment, bonding failures, temperature defects, dislocation etc. which may lead to transient and permanent faults in 3D NoCs.

In many researches, various fault tolerance routing algorithms has been proposed for tackling faults in 3D NoCs such as Dimension Order Routing [5], turn model routing [6] and 4NP-First [7] routing. The traditional fault tolerant routing algorithms uses detour methods [8] to reroute the packets. However rerouting of packets increases routing distance and hence network latency and traffic congestion. Pasricha et al. [7] designed partially adaptive routing algorithm for 3D NoCs based on turn model but it lacks in achieving full adaptivity due to routing restrictions of turn model.

Ebrahimi et al. [9] introduces 3D-FT to provide tolerance from both faulty links and nodes. It requires additional overhead of large area due to use of two, four and two virtual channels along X, Z and Y dimensions respectively. HamFa [10] eliminates the requirement of virtual channels and provide tolerance against faulty links. The approach does not require any overhead bits in the header and routing table. It uses Hamilton path strategy to deliver the packets at several destinations. But this approach does not provide tolerance for multiple horizontal and vertical faulty links.

AFRA [11] uses ZXY routing algorithm in the absence of faults and XZXY in the presence of faults. The authors in [12] proposed HARS, which is deadlock free routing scheme and extends DyAD of 2D routing to 3D scenario to support multi-fault and single-fault models. [13] proposed turn guided routing scheme TURO, which is deadlock free and lightweight and does not require any virtual channels. It provides higher performance and improved adaptivity as compared to other 3D turn models. Thus designing fault tolerant routing algorithm having high network performance has become a critical challenge in NoC applications.

Logic Based Distributed Routing (LBDR) [14] is a distributed routing logic for 2D mesh topology. The scheme is implementation efficient and can be implemented for any deadlock free routing algorithm for irregular networks. It does not need any routing tables but some bits to define routing decisions based on relative position of current node and destination node. This scheme has provided fault tolerance in 2D mesh topologies, but complexity and inefficiencies of the routing scheme increases with the increase in network sizes.

978-1-4673-6540-6/15/$31.00

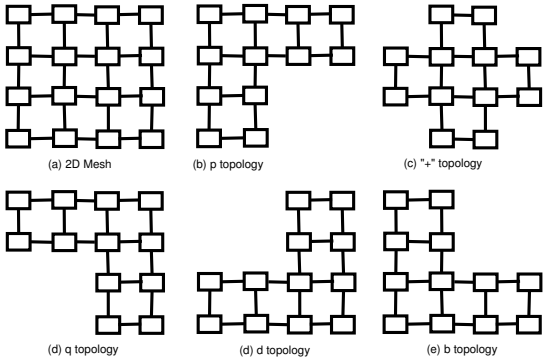

Fig. 1: Supported topologies of LBDR Logic for 2D NoCs

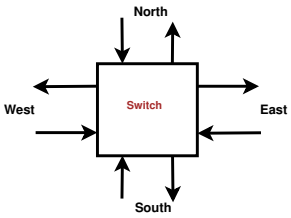

Fig. 2: 2D NoC Switch

In this paper, we have extended logic based distributed routing for handling permanent faults in 3D NoCs. The proposed scheme LBDR3D is a cost-efficient, reconfigurable routing scheme that overcomes the bottleneck of traditional routing methods. The basic idea for LBDR3D define routing logic for 3D NoC that handle faults using routing bit and connectivity bit. LBDR3D put routing restrictions and does not include the faulty path in the route computation which ensures high performance of NoC.

LBDR3D route the packet using horizontal and vertical channels depending on the routing algorithm routing restrictions to ensure deadlock freedom of the implemented algorithm. The scheme considers fault information of one hop distance in advance and makes routing decision accordingly. The rest of the paper is described as follows. Section 2 describes basics of LBDR and its computation logic for 2D NoCs. Section 3 describes extension of LBDR to proposed method LBDR3D for 3D NoCs. Section 4 describes the scheme analysis and comparison with traditional XYZ algorithm using NIRGAM [15] simulator. Section 5 concludes the paper.

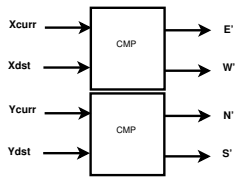

Fig. 3: First part of LBDR routing logic for 2D NoCs

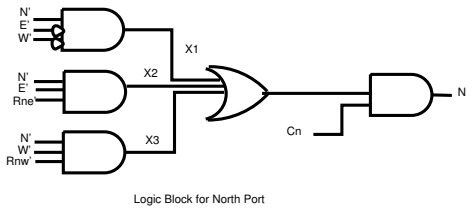

Fig. 4: Second part of LBDR routing logic for 2D NoCs

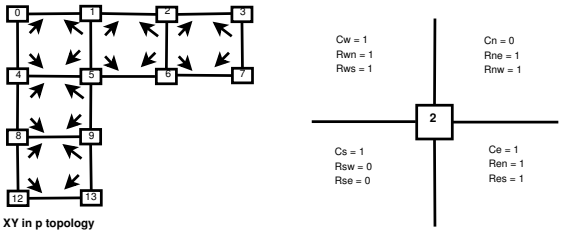

Fig. 5: (a) XY routing restriction in p topology for 2D NoCs (b) Quadrants of LBDR

II. LBDR Basics

A. *LBDR Logic for 2D NoCs*

The increase in the number of cores integration on a single chip and VLSI advancements for deep sub-micron technology has increased the probability of link failures. Thus 2D mesh topologies face some manufacturing defects and faults which may lead to failure of NoCs.

Logic Based Distributed Routing (LBDR) [14] has been proposed to handle irregular topology of 2D mesh derived from manufacturing defects as shown in Fig. 1. The routing scheme does not require any routing tables but some routing bits to compute routing decisions. LBDR can be applied to all the supported topologies having the property that all end-nodes should communicate to other nodes through minimal paths of the original mesh topology and every router should know its own coordinates. The paper [14] has shown LBDR application for XY and other topology agnostic routing algorithms like SRh and UD.

The routing logic of LBDR uses three bits per switch output port maintained under two sets as follows.

1) Connectivity Bits: It represents connectivity of the router with its neighbours in the current network topology. It is represented at each output port as C_x where x represents the connectivity of the current router with the neighbour router in x direction. Thus, the four ports of the router are having connectivity bits as $C_n, C_s, C_e,$ and C_w for representing connectivity in north, south, east and west directions respectively. The set bit of C_x represents connectivity through direct link with the neighbour router in x direction.

2) Routing Bits: It represents which routing directions can be taken depending on the routing restrictions of the

routing algorithm. It takes the routing decision based on one hop visibility from the switch. Routing bit can be represented as R_{xy} which shows whether the packet can take turn for y direction on the neighbour router from x direction of the current switch or not. The set bit of R_{xy} indicates specified turn can be taken on the neighbour router. For example: R_{ns} and R_{ne} indicates routing bit for North output port to represent whether South port or East port can be taken at the neighbour switch respectively to route packets from current switch.

LBDR thus, consists of one connectivity bit and two routing bits in 2D mesh topology. Each switch consists of four output ports as shown in Fig. 2, thus requires 12 bits per switch.

B. Routing Logic

The routing logic for LBDR consists of two parts:

a) First part of routing logic as shown in Fig. 3 computes relative position of the destination's switch using two comparators with X_{curr} and X_{dst} as input to first comparator and Y_{curr} and Y_{dst} as input to second comparator. The comparator output signal indicates the quadrant of the destination switch as shown in Fig. 5(b).

b) Second part of routing logic as shown in Fig. 4 constitutes four logic units. Each logic unit describes each output port. Since logic unit of each port is similar thus logic for south output port for next hop is taken to explain the satisfying conditions of LBDR. These routing logic conditions should be satisfiable according to the routing restrictions of routing algorithm. Fig. 5(a) shows the routing restrictions for p topology and its corresponding bits are shown in Table 1. The south port cannot be taken to route the packets if the following conditions are not satisfied:

1. if destination is in south direction $(S'.\overline{E'}.\overline{W'})$.

2. if destination is in south east quadrant and R_{se} = 1 $(S'.E'.R_{se})$, or

3. if destination is in south west quadrant and R_{sw} = 1 $(S'.W'.R_{sw})$

and if current switch has set connectivity bit in south direction (C_s = 1)

Thus second part of the logic is as follows:

- N"=N'.$\overline{E'}$.$\overline{W'}$ + N'.E'.R$_{ne}$ + N'.W'.R$_{nw}$.
- E"=E'.$\overline{N'}$.$\overline{S'}$ + E'.N'.R$_{en}$ + E'.S'.R$_{es}$.
- W"=W'.$\overline{N'}$.$\overline{S'}$ + W'.N'.R$_{wn}$ + W'.S'.R$_{ws}$.
- S"=S'.$\overline{E'}$.$\overline{W'}$ + S'.E'.R$_{se}$ + S'.W'.R$_{sw}$.
  - N=N".C$_n$.
  - W=W".C$_w$.
  - E=E".C$_e$.
  - S=S".C$_s$.

where N', S', E' and W' represents the north, south, east and west signals computed from the first part of the logic. N'', S'', E'' and W'' represents the intermediate signals computed in the second part of the logic to get signals for output port N, S, E and W respectively.

TABLE I: Routing and Connectivity bits of LBDR implementation for xy routing of p topology of 4x4 mesh

Switch	R$_{ne}$	R$_{nw}$	R$_{en}$	R$_{es}$	R$_{wn}$	R$_{ws}$	R$_{se}$	R$_{sw}$	C$_n$	C$_e$	C$_w$	C$_s$
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0

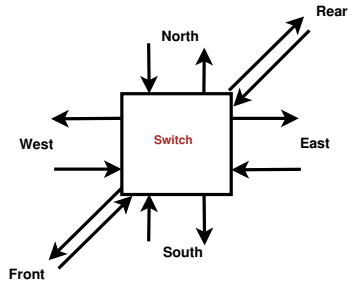

Fig. 6: 3D NoC Switch

III. Proposed Method: LBDR3D

A. LBDR Logic for 3D NoCs

LBDR for 2D mesh network is efficient in handling faults as the algorithms are mainly concentrated on XY plane but its complexity increases when the network size increases. Thus with the increase in the number of cores integration 3D NoCs has provided a promising solution. But in 3D NoCs the algorithm needs to consider XYZ plane and thus handling faults in XY, XZ and YZ plane using LBDR for 2D NoCs does not work efficiently. This section extends the LBDR approach for handling faults in 3D NoCs. The routing switch consists of 6 output ports namely N, S, E, W, F and R showing the direction of North, South, East, West, Rear and Front respectively as shown in Fig. 6.

The routing logic of LBDR3D uses the same concept as LBDR for 2D NoCs but instead of three bits, LBDR3D uses five bits per switch output port. These five bits can be categorized as 4 Routing bits R_{xy} and 1 Connec-

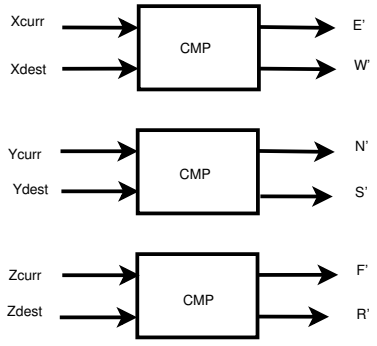

Fig. 7: First part of LBDR routing logic for 3D NoCs

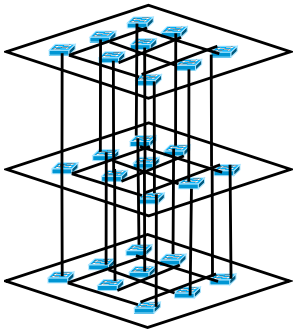

Fig. 8: 3D NoC

tivity bit C_x. Each switch consists of six output ports, thus requires 30 bits per switch. LBDR consists of four quadrants but LBDR3D consists of eight quadrants namely $NWF, NEF, SWF, SEF, NWR, NER, SWR, SER$. The two sets of bits of LBDR3D can be classified as follows:

1) Connectivity Bits: The connectivity bits C_x definition remain same but the number of directions for x add two additional directions connectivity: rear and front to provide connectivity to upward layer and downward layer of 3D NoCs respectively in Z direction from current layer router.

2) Routing Bits: The routing bits R_{xy} in LBDR3D represents the routing restriction for moving from the current router in additional directions of rear and front besides LBDR directions of north, south, east and west. The routing bits indicating routing restriction are set to zero while other bits are set to one including the one not existing in the topology.

The combinations of routing and connectivity bit for each output port are represented as follows:

- At North port:
  - Routing bits : R_{ne}, R_{nw}, R_{nf}, R_{nr}.
  - Connectivity bit : C_n.
- At South port:
  - Routing bits : R_{se}, R_{sw}, R_{sf}, R_{sr}.
  - Connectivity bit : C_s.
- At East port:
  - Routing bits : R_{en}, R_{es}, R_{ef}, R_{er}.
  - Connectivity bit : C_e.
- At West port:
  - Routing bits : R_{wn}, R_{ws}, R_{wf}, R_{wr}.
  - Connectivity bit : C_w.
- At Front port:
  - Routing bits : R_{fn}, R_{fs}, R_{fe}, R_{fw}.
  - Connectivity bit : C_f.
- At Rear port:
  - Routing bits : R_{rn}, R_{rs}, R_{re}, R_{rw}.
  - Connectivity bit : C_r.

B. Routing Logic

The routing logic for LBDR3D consists of two parts:

a) First part of routing logic computes relative position of the destination's switch using three comparators with X_{curr} and X_{dst} as input to first comparator and Y_{curr} and Y_{dst} as input to second comparator and Z_{curr} and Z_{dst} as input to third comparator as shown in Fig. 7. The signals from the first part having three comparator's indicates the quadrant of destination switch out of 8 quadrants of 3D NoCs. Fig. 8 shows layered architecture of 3D NoCs.

b) Second part of routing logic describes each output port to take next hop depending on the routing restrictions of the algorithm implemented. The second part take the routing signals from first part of LBDR3D to satisfy following conditions of LBDR3D. The conditions for North output port is explained as it remains similar for other ports as well.

- The packets destination is on the same column (N'.$\overline{E'}$.$\overline{W'}$.$\overline{F'}$.$\overline{R'}$).
- The packets destination is on the NE quadrant and the packet can take the E port at the next switch through the N port (N'.E'.R_{ne}).
- The packets destination is on the NW quadrant and the packet can take the W port at the next switch through the N port (N'.W'.R_{nw}).
- The packets destination is on the NF quadrant and the packet can take the F port at the next switch through the N port (N'.F'.R_{nf}).
- The packets destination is on the NR quadrant and the packet can take the R port at the next switch through the N port (N'.R'.R_{nr}).

The satisfying conditions for different output port can be defined as follows:

- N" = N'.$\overline{E'}$.$\overline{W'}$.$\overline{R'}$.$\overline{F'}$ + N'.E'.R_{ne} + N'.W'.R_{nw} + N'.R'.R_{nr} + N'.F'.R_{nf}
- E" = E'.$\overline{N'}$.$\overline{W'}$.$\overline{R'}$.$\overline{F'}$ + E'.S'.R_{es} + E'.N'.R_{en} + E'.R'.R_{er} + E'.F'.R_{ef}
- W" = W'.$\overline{N'}$.$\overline{W'}$.$\overline{R'}$.$\overline{F'}$ + W'.S'.R_{ws} + W'.N'.R_{wn} + W'.R'.R_{wr} + W'.F'.R_{wf}
- S" = S'.$\overline{E'}$.$\overline{W'}$.$\overline{R'}$.$\overline{F'}$ + S'.E'.R_{se} + S'.W'.R_{sw} + S'.R'.R_{sr} + S'.F'.R_{sf}
- F" = F'.$\overline{E'}$.$\overline{W'}$.$\overline{N'}$.$\overline{S'}$ + F'.E'.R_{fe} + F'.W'.R_{fw} + F'.N'.R_{fn} + N'.F'.R_{fs}
- R" = R'.$\overline{E'}$.$\overline{W'}$.$\overline{N'}$.$\overline{S'}$ + R'.E'.R_{re} + R'.W'.R_{rw} + R'.N'.R_{rn} + F'.S'.R_{rs}

N = N".C_n; E = E".C_e; W = W".C_w;
S = S".C_s; R = R".C_r; F = F".C_f;

where N', S', E', W', R' and F' represents the north, south, east, west, rear and front signals computed from the first part of the logic. N'', S'', E'', W'', R'' and F'' represents the intermediate signals computed in the second part of the logic to get signals for output port N, S, E, W, R and F respectively.

LBDR realize deadlock free routing techniques and ensures deadlock freedom in LBDR implementation by enforcing routing restriction of routing algorithm. The routing restriction prohibits the packet for the next hop while traveling from source to destination if it leads to deadlock. The next section

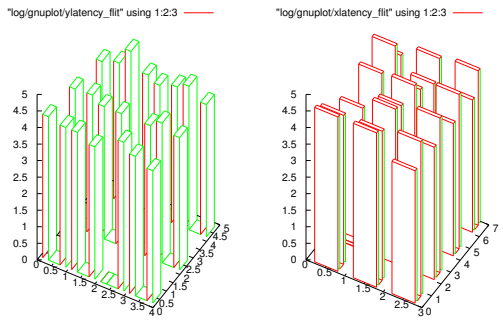

Fig. 9: Latency (in clock cycles) through XYZ algorithm without faults in uniform traffic

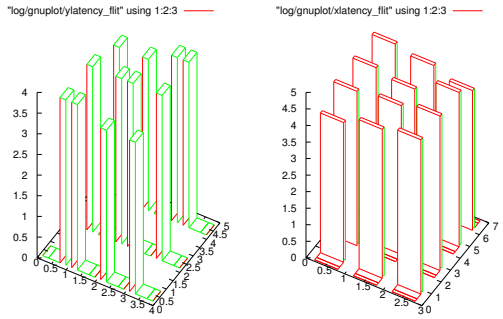

Fig. 10: Latency (in clock cycles) through XYZ algorithm without faults in transpose traffic

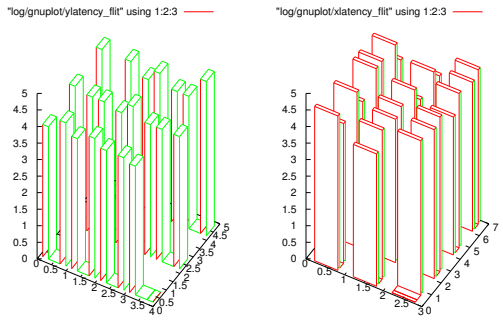

Fig. 11: Latency (in clock cycles) through LBDR3D implementation with faults in uniform traffic

implements the routing restriction of XYZ for four layered 3D NoCs with each layer having 4*4 mesh.

IV. Results Analysis

In this section we evaluate the performance of LBDR3D. The performance of LBDR3D can be evaluated by comparing its implementation results with that of routing algorithms implemented with routing tables. We have used cycle accurate NoC simulator NIRGAM [15] to evaluate LBDR3D. NIRGAM has been modified for link failures and performance of LBDR3D has been compared to routing table's usage performance. All simulations on NIRGAM are performed on 4x4 mesh having four layers with each layer of size 4x4 supporting wormhole switching technique. The packet size of 8 flits and input port buffers of size 6 flits have been taken. Each simulation runs for 1000 cycles and generates traffic for 100 cycles with warm-up sessions of 5 cycles. XYZ routing algorithm results have been compared with LBDR3D implemented XYZ routing algorithm.

For the performance metric, communication latency per channel has been used for channels X axis and Y axis as it represents the mesh layer of 3D NoCs. The latency can be defined as the number of cycles required to departure a packet

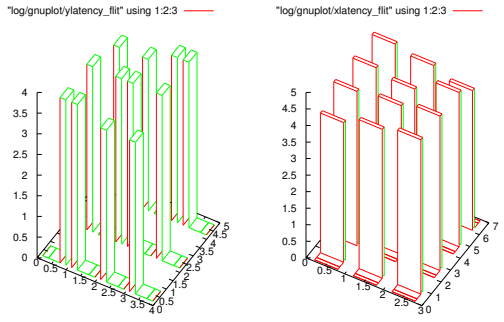

Fig. 12: Latency (in clock cycles) through LBDR3D implementation with faults in transpose traffic

from the source router to arrival of packet at the destination router. The performance of LBDR3D is evaluated for XYZ routing algorithm for uniform and transpose traffic.

1) Uniform Traffic Model: In the uniform traffic model, each IP core generates packet with specific packet injection rate and distributes to every other node with equal probability. Fig. 9 shows average latency per channel for XYZ routing algorithm without faults and Fig. 11 shows average latency for XYZ routing based LBDR3D routing scheme. For uniform traffic, LBDR3D routing scheme based XYZ mimics the performance of table based routing algorithm even in case of faults.

2) Transpose Traffic Model: In transpose traffic model, the coordinates of the source node are transposed to obtain the coordinates of the destination node. In case of transpose traffic LBDR3D mimics the performance of XYZ which can be seen in Fig. 10 and Fig. 12 for evaluating latency per channel for X axis and Y axis. The results show that in the presence of faults, the proposed method LBDR3D works as XYZ routing. LBDR3D is able to tolerate faults and achieve better performance than traditional implementation of routing tables.

V. Conclusion

In this paper we have presented cost efficient and fault tolerant LBDR3D routing scheme for implementing existing routing algorithms of 3D NoCs. LBDR3D based routing algorithms are less costlier than table based distributed routing as table based algorithms require larger silicon area for large irregular NoCs as compared to logic based distributed algorithms. LBDR3D requires only four routing bit and one connectivity bit to provide fault tolerance with one or multiple single link faults in each 2D mesh layer of 3D NoCs. Experimental results compares the performance of table based XYZ routing algorithm with LBDR3D based XYZ routing implementation. In the absence and in the presence of faults, LBDR3D mimics the performance of table based routing algorithm. Future work extends LBDR3D mechanism for supporting complex algorithms such as Elevator Routing algorithm.

References

[1] W.R. Davis et al. Demystifying 3d ics: the pros and cons of going vertical. *Design Test of Computers, IEEE*, 22(6):498–510, Nov 2005.

[2] Ying Wang. Economizing tsv resources in 3-d network-on-chip design. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 23(3):493–506, March 2015.

[3] Wen-Chung Tsai. A fault-tolerant noc scheme using bidirectional channel. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 918–923, June 2011.

[4] H.C. Chuang et al. Fabrication of through-silicon-via (tsv) by copper electroplated in an electrolyte mixed with supercritical carbon dioxide. In *Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS), 2015 Transducers - 2015 18th International Conference on*, pages 464–467, June 2015.

[5] M.M.H. Rahman, A. Shah, and Y. Inoguchi. A deadlock-free dimension order routing for hierarchical 3d-mesh network. In *Computer Information Science (ICCIS), 2012 International Conference on*, volume 2, pages 563–568, June 2012.

[6] Ge-Ming Chiu. The odd-even turn model for adaptive routing. *Parallel and Distributed Systems, IEEE Transactions on*, 11(7):729–738, Jul 2000.

[7] S. Pasricha and Yong Zou. A low overhead fault tolerant routing scheme for 3d networks-on-chip. In *Quality Electronic Design (ISQED), 2011 12th International Symposium on*, pages 1–8, March 2011.

[8] M. Valinataj et al. A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip. *Electronics and Communication (AEU), IEEE Journal of*, 65(7):630–640, July 2011.

[9] M. Ebrahimi et al. Fault-tolerant method with distributed monitoring and management technique for 3d stacked meshes. In *Computer Architecture and Digital Systems (CADS), 2013 17th CSI International Symposium on*, pages 93–98, Oct 2013.

[10] Masoumeh Ebrahimi et al. Fault-tolerant routing algorithm for 3d noc using hamiltonian path strategy. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1601–1604, March 2013.

[11] S. Akbari et al. Afra: A low cost high performance reliable routing for 3d mesh nocs. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 332–337, March 2012.

[12] Jun Zhou et al. Hars: A high-performance reliable routing scheme for 3d nocs. In *Proceedings of the 2014 IEEE Computer Society Annual Symposium on VLSI*, ISVLSI '14, pages 392–397, Washington, DC, USA, 2014. IEEE Computer Society.

[13] Jun Zhou et al. Turo: A lightweight turn-guided routing scheme for 3d nocs. In *Low-Power and High-Speed Chips (COOL CHIPS XVIII), 2015 IEEE Symposium in*, pages 1–3, April 2015.

[14] J. Flich and J. Duato. Logic-based distributed routing for nocs. *Computer Architecture Letters*, 7(1):13–16, Jan 2008.

[15] L. Jain et al. Nirgam: a simulator for noc interconnect routing and application modeling. In *Design, Automation and Test in Europe Conference (DATE), 2007 Workshop on Diagnostic Services in Network-on-Chips*, pages 1–2, Sept 2007.