Fault-Tolerant Circular Routing Algorithm for 3D-NoC

Razieh Alizadeh, Mohsen Saneei
Department of Electrical Engineering
ShahidBahonar University of Kerman
Kerman, Iran
ra.alizadeh@eng.uk.ac.ir;msaneei@uk.ac.ir

Masoumeh Ebrahimi
Department of IT; ES Department
University of Turku; KTH
Turku, Finland; Stockholm, Sweden
masebr@utu.fi; mebr@kth.se

***Abstract*—ExpandingNetworks-on-Chip (NoCs) to the third dimension (3D-NoC) has been known as a promising solution for the latency challenges of future many-core Systems-on-Chip. 3D-NoC may take advantages of TSVsfor vertical links which are shorter and faster than horizontal ones. Faults may occur in TSVs as well as the horizontal links though faults inTSVs are more costly. In this paper, we present a fault-tolerant routing algorithm targeting faults in both TSVs and horizontal links. The proposed routing algorithm is based on defining some circular routing paths which offers a deadlock-free routing for packets in mesh-based topologies. In addition to tolerating faults, these circular pathshelp in reducing congestion in the centralpart of the network at high injection rates. The proposed circular routing algorithm is able to tolerate all one-faulty links. In addition, it is shown that its performance is better than those of traditional methods.**

I. INTRODUCTION

Networks-on-Chip has been proposed as a solution to address the scalability problem of System-on-Chip designs[1][2]. Scalability and reusability characteristics of NoC may result in reducing design time and shortening time to market for new products. NoC consists of an interconnection of many switches to enable a large number of cores to communicate with each other. In a mesh-based NoC, each core is connected to a switch by a local network interface. Cores can communicate with each other by propagating packets through switches in the network. Each switch is connected to its neighbors through bidirectional links. Typically in NoCs, the resources are scarce regarding the ever increasing demand for a high performance communication among cores [3].

The 3D integration paradigm addresses the interconnect problem and area consumption of 2D-based NoCs[4]. 3D NoCs are composed of 2D layers stacked on top of each other and connected by Through-Silicon Vias (TSVs)[5][6]. TSVs have different characteristics than horizontal ones as they offer shorter, faster and more power efficient interconnects than horizontal links[6].

Since the 3D technology is new, it carries also new challenges such as poor yield, thermal issues and traffic balancing[8][9]. Low yield for current TSV fabrication process seriously affects the reliability of 3D NoCs[10]. This low reliability is because of different factors such as misalignment, dislocation, void formation, failure on bonding, defects due to temperature, and random open defects[11][12]. These factors can lead to permanent and transient faults on different elements of NoCs. In this paper, we address permanent faults in bidirectional and unidirectional TSVs as well as the horizontal links. In this direction, we propose circular routing which is a deadlock free fault-tolerant routing algorithm for 3D-mesh topologies based on circular paths.

Implementing deadlock-free fault-tolerant algorithms are usually complicated in a 2D mesh network while the complexityincreases in a 3D mesh network. This is due to the fact that in a 2D mesh network, the algorithms were concerned with twoabstract cycles in a XY plane while in a 3D network the cycles should be prevented within each layer (i.e XY, XZ, and YZ) andbetween layers.

Adaptive routing algorithms can be either minimal or non-minimal. In minimal routing algorithms, packets use only shortest paths for transmission. In non-minimal routing algorithms, packets can take longer paths and temporarily move away from the destination[13].

In low traffic loads, minimal adaptive routing algorithms achieve better overall performance, while in high traffic loads non-minimal adaptive algorithms perform better. This is due to the fact that minimal routing methods are very limited to avoid hotspots during periods of high network activity as they cannot misroute packets around congested regions.

Circular routing is a non-minimal routing algorithm which can be implemented with negligible hardware overhead in NoCs. Its non-minimal routing characteristic provides the potential to achieve low congestion at the central part of the network at high injection rates.

The proposed algorithm satisfies three main goals as simplicity, performance, and robustness. To obtain simplicity, in the circular routing approach, each router needs to know the fault information of its adjacent links and no other extra links, routing table or overhead bits in packets is required. The proposed circular routing satisfies performance by avoiding packets through central parts and thus reducing congestion at central regions. The circular routing algorithm improves the performance in both faulty and non-faulty networks.

The rest of the paper is organized as follows. Related work is given in Section II. Section III describes the

circular routing and fault-tolerant model. Experimental results are presented and discussed in Section IV, followed by the conclusion in Section V.

II. Related Work

In 3D NoC, faults have been studied in many different aspects to secure the network against faults.Using bypass links to bypass faulty links or faulty nodes is introduced by designing resilient TSVs in [9]. In this paper, it has been suggested to duplicate every TSV to provide a better fault tolerant network. Although this approach is highly reliable, it is at the cost of adding additional TSVs which imposes large footprints.In another approach, more wires (e.g. M) are used in addition to the N wires. Thereby, the resulting link uses M+N wires to support at most M faults. These links require the circuits to distribute bits over M+N wires in order to extract the N original bits[14].

In many other researches, fault problems are tackled with routing algorithms. AFRA [15] is an algorithm presented for the 3D mesh network. AFRA uses two simple routing ZXY and XZXY in the absence and presence of faults, respectively. AFRA is only able to tolerate single faults on TSV links and each router needs to know about vertical links of all routers in the same row. Another routing algorithm is proposed in[16]. This method is suitable for 2D and 3D NoCs but because of its large area overhead, due to using tables, it is not a proper approach for large networks.

In[17], authors presented a fault-tolerant method based on deflection routing. This method suffers from poor scalability as a routing table of size n*4 is needed at each router. Minimal routing algorithms based on the concept of faulty blocks have been presented in [18][19]. These proposals just focus on faulty nodes but not faulty links.

3D-FT[3] deals with both faulty nodes and links, but it uses two, two, and four virtual channels along the X, Y, and Z dimensions, respectively which leads to a large area overhead.

HamFa[20]tolerates faulty links and it does not require any virtual channels. This method uses the Hamilton path strategy when a packet has several destinations to be delivered. This routing algorithm does not require any routing table and does not impose overhead bits in the header. However, it cannot tolerate multiple vertical and horizontal faulty links.

In this paper, we propose a circular routing algorithm for 3D NoCs which is able to tolerate multiple faults in horizontal and vertical links. This shows its advantages over HamFa[20] which is able to tolerate single faults. In contrast to 3D-FT[3], thecircular routing algorithm is light-weight as it does not require any virtual channels. Unlike AFRA[15], the presented algorithm works based on a limited knowledge about the location of faults and tolerates faults on horizontal links as well as vertical links.

III. Circular Routing and Fault Model

In this section, we discuss about the circular routing algorithm in 3D NoCs given several examples and pseudo codes.We first describe the circular routing algorithm in details and the labeling mechanism.Then we investigate how this algorithm tolerates faults in both vertical and horizontal links. Finally, we prove that this algorithm is deadlock-free.

A. Circular routing algorithm

The circular routing algorithm is proposed for both 2D and 3D mesh topologies. Based on this algorithm, the packets are routed through circles to reach their destinations. Let us explain the operation of circular routing in a 6*6 2D mesh. This can be considered as the first layer of a 6*6*z 3D network, where z is the number layers.

The Number of circles in a networkis dependent on the number of nodes in that network. In the a*b network if b is equal or less than a, the number of circles can be calculated by the following equations:

Number of circles = $b \div 2$ where b: even (1)

Number of circles = $(b + 1) \div 2$where b: odd (2)

As shown inFigure , each node is assigned a label in a circular manner from 0 to N-1 in which N is the number of nodes in the network.

Number of nodes in each circle in an a*b mesh, can be calculated by following sequence:

Nodes in circle-0 = 2(a+b) -4
Nodes in circle-1 = 2(a+b-4) -4
Nodes in circle-2 = 2(a+b-8) -4
Nodes in circle-n = 2(a+b-4*n) -4 (3)

The total nodes incircle-0 to X^{th} circle is:

$$Total\ nodes\ X\ = \sum_{n=0}^{X}(2(a+b-4*n)-4) \quad (4)$$

In an a*b*c mesh, each node is presented by an ordered triple (X, Y, Z). The following equations show assigning the (X, Y) labels. Cir is the circle number in which the node stands on it.

If Y=Cir:
(X,Y) label = Total nodes (Cir-1) + (X-Cir) (5)
If X=a-Cir:
(X,Y) label = Total nodes (Cir-1) + (X-Cir + Y-Cir) (6)
If X=Cir:
(X,Y) label = Total nodes Cir – (Y-Cir) (7)
If Y=a-Cir:
(X,Y) label = Total nodes Cir – (Y-Cir + X-Cir) (8)

These equations can be extended to third dimension by adding the follow statement:

Nodes number = (a*b*Z) + (X, Y) labels (9)

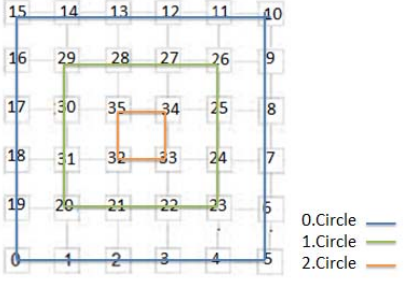


Figure 1.6*6 mesh physical network with the label assignment; three circles which are shown in different colors

```
1: IFDCir == CCirTHEN
2:     IF (Distance between Did and Cid >Diameter) THEN
3:          NextNode = Lid-1;
4:     ELSIF (Distance between Did and Cid <Diameter) THEN
5:    NextNode = Lid+1;
6:     ELSIF (Distance between Did and Cid =Diameter) THEN
7:          Packet choose ascending path or ascending path randomly;
8:     END IF;
9:  ELSE
10:    Route to the nearest node to current and destination node;
11: END IF;
```

Figure 1. Pseudo code of Circular routing algorithm

According to these equations, there are some nodes which have equal X and Y while the labels of these nodes are different. These nodes are located in the same position but in different layers. We call these nodes as shadow nodes.

Here we explain an example of calculating nodes number in a 6*6*6 mesh network, a node that is presented by triple ordered (2, 1, 2). This node is located in circle-1. Since Y = circle number = 1, we use first equation to calculating (X,Y) label of the node:

$$Total\ nodes\ 0\ = \sum_{n=0}^{0}(2(6+6-4*n)-4) = 20$$

(X,Y) label = 20 + (2-1) = 21

Node 21 is shadow node of the (2, 1, 2) node :

Nodes number = (a*b*Z) + (X, Y) labels = (6*6*2)+21

Nodes number = 93

The numbers of the circles in a layer depends on the dimension of that layer. Three circles exist in a 5*5 meshnetwork (Figure 2). The first circle, called Circle-0,contains the nodes (0,1,2,…, 15) while Circle-1 and Circle-2 contains the nodes (16, 17, 18,…, 23) and (24), respectively. Two paths are constructed by the labeling: ascending path and descending path. A packet is called moving along the ascending path when the label of the neighboring node is greater than the current node. On the other hand, the packet is called moving along the descending path if the label of neighboring node is smaller than the current node. There is one exception and this is where a packet moves from the node with the smallest number in a circle (e.g. the node 0 in the circle-0 to the largest number in that circle (e.g. the node 15 in the circle-0). In this case by moving from the node 0 to the node 15, the packet is still counted as moving in the descending path. For an example the path P:{2,1,0,15, and 14} is a descending path.

Figure 1shows the process of the circular routing algorithm. We denote the destination circle and current circle with DCir and CCir respectively, while the current node ID and destination node ID are represented with Cid and Did.

According to this algorithm, in the case when both of source and destination nodes are in the same circle and the distance between the label of the destination node and label of the source node is greater than the diameter of the circle, the packet route to its destination using the descending path. Otherwise, the packet moves in the ascending path to reach its destination. If the distance between two nodes in a circle is equal to the diameter of the circle, either path can be chosen. The diameter of a circle with the size of a×b is defined as a+b-2. As an instance in Figure 2, the diameter of 0-circle with size of 5×5 is 8.

In Figure 2assume a casewhere the node 1 sends a packet to the node 7. In this case, since the distance between two nodes (i.e. 6) is smaller than the diameter size in the circle-0 (i.e. 8), the ascending path will be chosen. On the other hand, if the source and destination are in different circles, the packet will be routed to the nearest node to destination in a destination's circle. For example in Figure 2, the node 23 sends a packet tothe node 8. Since these nodes belong to different circles, the packet should be first sent from the circle-1 to the circle-0 as soon as possible, so the next node can be the node 14 or the node 11. The node 11, along the Y direction is the nearest nodes to the node 8 of the circle- 0. As another example, if a corner node like the node 0 wants to send a packet to the node 17, the packet has to be delivered to circle-1. The node 16 is the nearest node to destination, so in the second step one of the nodes in X or Y should be chosen. If there is no difference between the node 1 and the node 15 in the hop count, the direction with less congestion will be chosen.

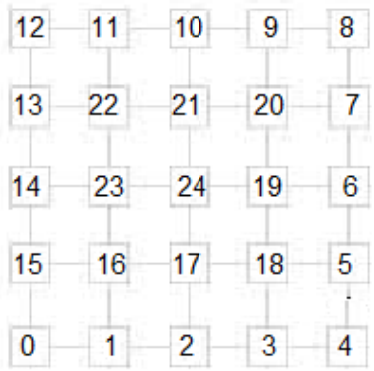


Figure 2.5*5 mesh physical network with the label assignment (the first layer of a 5*5*5 mesh)

```
1: IFzD = zc
2:    IF C=D THEN
3:          Deliver to local IP;
4:    ELSE
5:          Circular 2D Route to D;
6:    END IF;
7:  ELSE
8:    WHILEzD<>zC
9:          IF (Vertical and Horizontal link are healthy) THEN
10:               IFzD>zC THEN
11:                     NLink = UP;
12:               ELSE
13:                     NLink = Down;
14:               END IF;
15:   ELSIF (Vertical link was broken) THEN
16:               Circular route to shadow node;
17:         ELSIF (Horizontal link was broken) THEN
18:               Route to up or down;
19:         END IF;
20:   ENDWHILE;
21: END IF;
```

Figure 3. Global view of 3D routing

B. Fault Model

A 3D NoC is composed of several layers connected by vertical links. Our fault-tolerant routing algorithm is presented upon the following general assumptions.

- The packets are not allowed to select the input port as output (180-degree turn is not allowed).
- We consider faults as completely broken links (permanent faults).
- The faulty links can be unidirectional or bidirectional.
- Between each two adjacent layers, there are at least two non-faulty vertical links connecting the non-faulty nodes at both ends.
- All non-faulty nodes on the same layer are reachable from one another.
- All the links that connected router to its core are non-faulty.

The main idea of the circular routing algorithm is to move the packet to the destination layer first (if it is not already there). This is due to the fact that the possibility of finding a healthy link decreases as the packet gets closer to the destination.

Thereby, in the presented algorithm, the priority of routing packets along the Z dimension is higher than X or Y. However, if the UP or DOWN link was not healthy, the current node sends packet along the X or Y dimension toward the destination shadow. As it is already mentioned, the shadow node is a node with the same position but in a different layer. If the current node is the shadow node of the destination node, the packet will be sent to the next circles in order to find a vertical healthy link. The pseudo code of the algorithm is shown in Figure 3.

Let us explain the routing algorithm using an example illustrated inFigure4. In this figure, assume that the vertical links (C1-D1) and (C2-A) are faulty (red lines) and the nodes C1 and C2 want to send a packet to the node D1. Faults on the link (C1-D1) disconnect C1 and D1 in the Z dimension so that the packet has to be routed to the neighboring node (i.e. node 23). The upward link of the node 23 is safe, thereby the packet will be sent to the destination layer through this link. Finally, the packet reaches the destination layer and will be routed to the D1 node by moving along the circle-1.

Now, let us follow the path of the packet from the node C2 to node D1. According to this algorithm since the layer of the current node is different from the destination node, the possibility of sending the packet through the Z dimension is checked. Since the link (C2-A) is faulty, the packet is routed toward the destination shadow (C1). Along this path, the packet is sent to the Z dimension as soon as finding a non-faulty vertical link (e.g. node N in this example). By reaching the destination layer, the packet is forwarded to the destination node.

If a fault occurs in one of the horizontal links, the vertical links are used to support this fault. In Figure4, let us assume that the horizontal link (C3-D2) is faulty and the node C3 wants to send a packet to D2. The node C3 sends the packet to the upward link. In this layer, the packet is routed to the destination shadow and finally the packet is delivered to the destination.

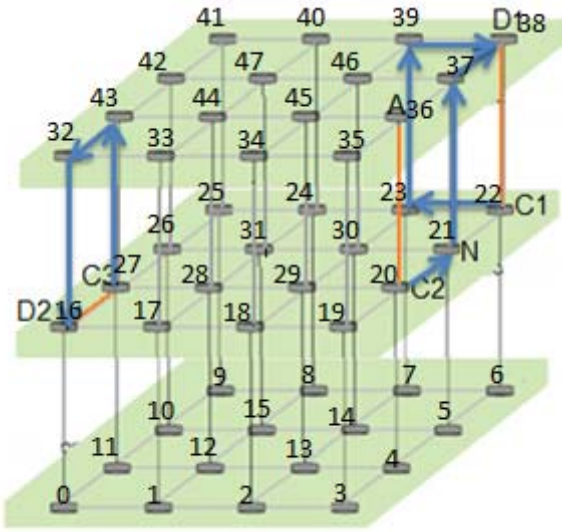


Figure4.4*4*3 mesh with circular labeling

C. *The circular algorithm is deadlock-free*

To show that the algorithm is deadlock-free, we prove that the deadlock cannot occur in absence and presence of faults.

In the absence of faults, if the source and destination are in different layers, when a packet left a layer never uses or goes back to the previous layer.Therefore, a cycle can never be formed between layers. On the other hand the algorithm is deadlock free in a layer according the following proofs.

A: The network is deadlock free between circles.

Proof: According to the algorithm, when the source and destination nodes are in different circles, by leaving a circle, the packet never uses or goes back to that circle again. This proves that there is no deadlock among circles.

B: The network is deadlock free within circles.

Proof: A cycle can be formed if packets can take both ascending and descending paths. However, in the proposed algorithm, packets can be either routed in the ascending or descending path.

A fault maybe occurs on a horizontal or vertical link. In both cases we have to change the layer to tackle these faults.

A: If a fault occurs on a vertical link, the network remains deadlock free.

Proof: When source and destination are in different layers and a broken vertical link cuts the path, packet will be sent to the destination layer through another links and it never goes back to the previous layer.

B: If a fault occurs on a horizontal link, the network remains deadlock free.

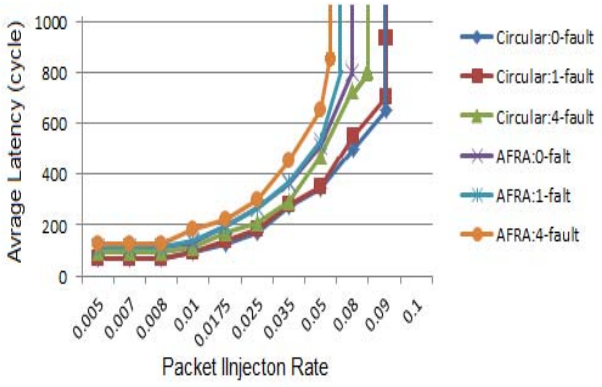
Proof: To tackling a faulty horizontal link, the packet must be sent to up or down layers (which are less congested). Packet routes toward the shadow node in the up or down layer. But the direction of routing path is not changed. It means that if the packet was moved in ascending/descending path in the original layer, ascending /descending path would be chosen in the up or down layer as well. As a result, the packet never goes back and a cycle is not formed.

IV. Simulation Results

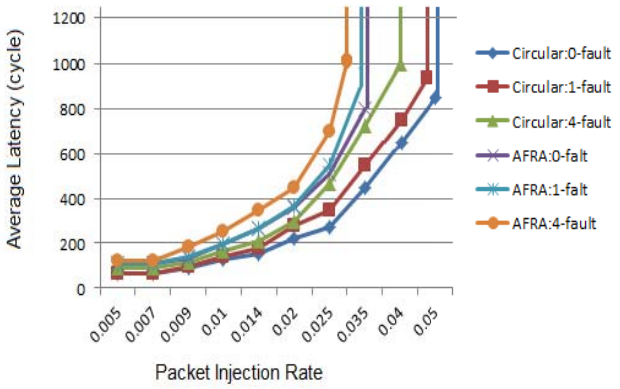
We evaluate the performance of the routing algorithm using a cycle-accurate NoC simulator developed with OMNET++ to model all components of the on chip network. Each message has 2 packets where these packets contain 2 flits with the width of 32 bits. Delay that considered for the links is 2ns. As a performance metric, we measure the average packet latency under the uniform and hotspot traffic patterns. The packet latency T is calculated by Tn+Tq where Tn and Tq are the number of cycles that a packet moves inside network and waits in the source nodes, respectively. We perform the experiments on a 6*6*3 3D mesh network. The average performance is measured over 100,000 cycles. We compare our proposed algorithm and AFRA under different traffic patterns.

A. *Performance Analysis*

We evaluated the performance of the circular routing algorithm is compared with the AFRA routing algorithm. Figure 5shows the average message latency of both circular routing and AFRA in three different situations (i.e., fault free, single fault, and four faults) and under the uniform and hotspot traffic profile. To reach different packets injection rate we increase the number of packets traveling the network at the same time, by decreasing the interval between sending packets to network, and see the impact of congestion on the performance of both algorithm with different traffic patterns. As observed from the results in Figure 5, circular routing performs better than AFRA under all conditions of the uniform and hotspot traffic pattern and in fault free and faulty cases. The reason is that, circular routing routes packets through periphery nodes while AFRA routes packet through central parts. According to this figure, circular routing improves the saturation point by 28.5%, and 60.6% under uniform and hotspot traffic patterns, respectively when there is a single fault in the network. Under similar conditions, the improvement of 26.8% and 33.33% is obtained when there are four faults in the network. Additionally, in the absence of faults, circular routing improves the performance by 28.58% and 52.7% under uniform and hotspot traffic patterns, respectively.



(a)



(b)

Figure 5.Performance analysis of circular routing in 6*6*3 3D-mesh network (a) uniform traffic (b) hotspot traffic in fault-free, 1-faulty and 4-faulty cases

B. *Reliability Evaluation*

In this section we calculate the reliability of the network, meaning that sources are able to deliver packets to every destination in the network without any packet drop. In the other words, if there is even one packet drop, the network will be counted as unreliable [20].All faulty links are selected using a random function. The results are obtained using 8,000 iterations. We evaluated and compared the reliability of circular routing and AFRA under the uniform traffic pattern. The number of faulty links increases from 1 to 3 in a 6*6*3 mesh network. As shown inFigure 6, circular routing can tolerate one, two and three faulty links by 100%, 67%, 23% reliability, respectively which is obviously better than the reliability offered by AFRA.

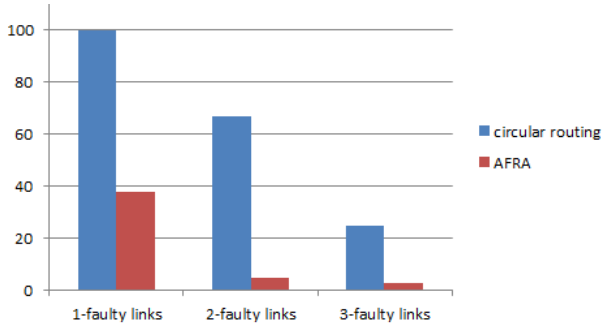


Figure 6. Reliability Analysis

C. *Hardware Analysis*

In this subsection, the overhead of theour proposed 3D routing is assessed, in terms of supplementaryfields that must be stored at each node and in each message, andthe complexity of the routing algorithm.

To implement the 3D routing algorithm, the nodes in the NoCdo not need to keep global information (e.g.routing tables or lists of failed nodes or links), so the memory overhead is minimized. For the fault tolerant 3D routing,a small and constant amount of data (related to nodes links) needsto be kept at each node. Thus, the routing algorithm can scale toany number of nodes in each layer and any number of layers. And there is no need for extra bits in packets header to carrying information about faults and topologies. Therefore our proposed algorithm equipped minimized area.

V. CONCLUSION

In this paper, we have proposed circular routing, a low-cost, highly efficient, reliable routing algorithm for 3D mesh NoCs. Circular routing tolerates fault on vertical and horizontal links, either bidirectional or unidirectional.Ourproposed algorithm is able to tolerate one, two and three faulty links by 100%, 67%, 23% reliability in 3D NoCs without using any virtual channel.

The Circular routing algorithm is simple, does not need any distribution mechanism, look up tables or additional information in the header flit.According to the results, circular routing improves the performance over AFRA over all conditions in uniform and hotspot traffic profiles and under non-faulty, single and four faults.

REFERENCES

[1] M. Palesi and M. Daneshtalab (Eds.), "Routing Algorithms in Networks-on-Chip", Springer 2014, ISBN: 978-1-4614-8273-4.

[2] Pavlidis, V. F. and E. G. Friedman. "3-D topologies for networks-on-chip." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, p.p.1081-1090.2007.

[3] M. Ebrahimi et al.,"Fault-tolerant Method with Distributed Monitoring and Management Technique for 3D stacked mesh", in Proc. of 17th CSI International Symposium on Computer Architecture & Digital Systems, (CADS), pp. 93-95, 2013.

[4] K Banerjee,SJ.Souri, P. Kapur, KC.Saraswat. "3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration." In Proceedings of the IEEE, 2001,p.p 602-633.

[5] B. S. Feero and P. P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A PerformanceEvaluation," IEEE Transactions on Computers, vol. 58, no. 1, pp. 32 –45, 2009.

[6] H. Matsutani, M. Koibuchi, and H. Amano, "Tightly-Coupled Multi-Layer Topologies for 3-D NoCs,"in Proceedings of 41st International Conference on Parallel Processing, 2007, p. 75.

[7] S. Spiesshoefer, et al." Z-axis interconnects using fine pitch, nanoscale through-silicon vias". In Proc. ECTC,p.p.466-471. 2004

[8] Igor Loi, Subhasish Mitra, Thomas H. Lee, Shinobu Fujita, Luca Benini, "A lowoverhead fault tolerance scheme for TSV-based 3D network on chip links", in Proc. IEEE/ACM Int. Conf. Computer-Aided Design,, IEEE Press, Piscataway, NJ, USA, 2008, pp. 598–602.

[9] Ayse K. Coskun, Jose L. Ayala, David Atienza, Tajana Simunic, Yusuf Leblebici, "Dynamic thermal management in 3D multicore architectures",In Proc. of DATE,p.p.1410-1415. 2009.

[10] I. Loi, S. Mitra, T. H. Lee, S. Fujita, and L. Benini, "A low-overhead fault tolerance scheme for tsv-based 3d network on chip links," in Proc. IEEE/ACM Int. Conf. Computer-Aided Design, pp. 598–602. 2008.

[11] A. Hsieh, et al. "TSV redundancy: architecture and design issues in 3D-IC". In Proc of DATE. Leuven, Belgium,p.p. 711-722. 2010.

[12] I. Loi, et al. "Characterization and Implementation of Fault-Tolerant Vertical Links for 3-D Networks-on-Chip". IEEE TCAD, 2011.

[13] M. Ebrahimi, M.Daneshtalab, J.Plosila,, “LEAR – A Low-weight and Highly Adaptive Routing Method for Distributing Congestions in On-Chip Networks”, in Proc. of PDP, pp. 520-524, 2012.

[14] U. Kang et al., “8 Gb 3-D DDR3 DRAM Using Through-Silicon-ViaTechnology” . IEEE JSSC, January 2010.

[15] S. Akbari, A.Shafiee, M.Fathy, R. Berangi, “AFRA: A low cost high performance reliable routing for 3D mesh NoCs,” In Proc. of DATE, pp.332-337, 2012.

[16] D. Fick, et al.”A Highly Resilient Routing Algorithm for Fault-Tolerant NoCs”. In DATE, p.p.21-26. 2009.

[17] Chaochao Feng et al.,” A Low-overhead Fault-aware Deflection Routing Algorithm for 3D Network-on-Chip”. Computer Society Annual Symposium on VLSI,p.p.598-602. IEEE 2011.

[18] J. Wu. “A simple fault-tolerant adaptive and minimal routing approach in 3-D meshes”. In Journal of Computer Science and Technology,p.p.1-13. January 2003.

[19] D. Xiang, et al. “Fault-tolerant routing in meshes/tori using planarly constructed fault blocks”. In ICPP,p.p.577-584. 2005.

[20] M.Ebrahimi, M.Daneshtalab, J.Plosila,”Fault-tolerant routing algorithm for 3D NoC using Hamiltonian path strategy”..In Proc. of DATE,p.p.1601-1604. 2013.