CoBRA: Low Cost Compensation of TSV failures in 3D-NoC

Ronak Salamat*, Masoumeh Ebrahimi†, Nader Bagherzadeh* and Freek Verbeek‡

*University of California, Irvine
† KTH Royal Institute of Technology and University of Turku, Finland
‡ Open University of The Netherlands, Radboud University, Nijmegen

Abstract—**3D-NoC has emerged to provide fast and power efficient connection between the layers of 2D-NoCs using Through-Silicon-Vias (TSV). Thermal stress, warpage, impurities and misalignment during the manufacturing process make these expensive TSVs vulnerable to faults. Chips with faulty TSVs should be either discarded or utilized by providing a proper fault-tolerant method. In this paper, we target designing a reconfigurable fault-tolerant routing algorithm capable of tolerating fabrication-time or run-time TSV failures. The proposed algorithm ensures a fault-free communication between any two nodes in the presence of TSV failures. Experimental results show that the proposed fault-tolerant routing algorithm provides 100% reliability as long as there is one healthy TSV in the eastmost or westmost column. The reliability of the counterpart algorithm, the Elevator-first routing algorithm, drops to 75% and 45% in presence of one and two faulty TSVs, respectively.**

1. Introduction

The positive correlation between the length of long global wires and performance bottlenecks such as delay and power consumption, has driven the research toward vertically stacking multiple dies using TSVs [1]. The main issues regarding the use of TSVs are the large area overhead of TSV pitches, the yield reduction caused by the large number of TSVs, and TSV cost. Moreover, the manufacturing process of TSVs has become a main challenge due to the variability of the manufacturing process [2] [3]. The cost of high yield TSV manufacturing process is only justifiable in presence of a practical solution to counteract TSV related effects (such as voids in TSVs, TSV pinch-off, oxide defects such as pinholes, thermo-mechanical stress, cracks in micro-bumps, chip warpage, and impurities [4]) which may render the entire chip useless [5]. Also TSV failures may be introduced at run-time because of e.g. thermal issues and Single Event Effect (SEE) [6]. For all these reasons, TSVs are vulnerable and prone to failure. Instead of discarding the entire chip with defective TSVs, measures are taken to enhance the fault tolerance of the 3D-NoC for dealing with TSV failures. A comprehensive list of both fabrication and runtime TSV failure sources have been presented in [6].

Employing traditional routing algorithms such as XYZ in vertically partially connected 3D-NoCs ensues performance degradation due to lack of mechanisms to handle faulty TSVs. Hence, a routing algorithm capable of spotting healthy TSVs in the presence of defective TSVs enhances the fault tolerance2 of 3D-NoC considerably.

In this paper we propose a reconfigurable fault-tolerant algorithm, named Column-Based Routing algorithm (Co-BRA). CoBRA is an adaptive routing algorithm which has two virtual channels along the Y dimension, called $Y0$ and $Y1$. CoBRA suggests a dynamic elevator assignment, meaning that the elevator selection is performed at each router while the packet is forwarded toward the destination. The routing decision is made based on the partial knowledge of routers about the location of TSV failures. This partial knowledge avoids packets from being forwarded to faulty TSVs. Therefore, performance improves and faults are tolerated.

2. Related work

Routing is responsible to deliver packets to their destinations. Generally routing algorithms are divided into static and adaptive [7] [8] where adaptive routing is a better candidate for a fault-tolerant routing due to its inherent path redundancy.

Examples of fault-tolerant routing algorithms in 3D-NoCs have already been presented in [9] and [10]. Hamiltonian path strategy [9] tolerates one faulty links either horizontal or vertical in 3D mesh NoCs without using any virtual channels. [10] deals with multiple faults in the 3D-NoC architecture with a limited quantity of TSVs; the authors apply deadlock recovery schemes rather than avoiding deadlock. Elevator-first [11] is a routing algorithm which has been proposed for partially connected 3D-NoCs. The elevators for vertical transmission are planned at design time based on the location of TSVs. Thereby, the algorithm cannot be reconfigured by any TSV failure which have been occurred after the manufacturing process or at runtime. Elevator-first is a deterministic routing algorithm having two virtual channels along the X and Y dimensions.

3. CoBRA Routing Algorithm

In this algorithm, routers do not have a global knowledge about the location of TSVs as it is the case in the Elevator-first routing algorithm. The only information that routers maintain is the presence of any healthy TSV in the same column (with the same X value) as the current router. The details on the propagation of TSV statuses are elaborated in Section 4.

CoBRA uses two virtual channels along the Y dimension to guarantee the freedom from deadlock. For this purpose, the network is virtually partitioned into two disjoint sub-networks: Subnetwork1 $(X^+, Y0^*, Z^+)$ and Subnetwork2

$(X^-, Y1^*, Z^-)$ where +, - represent channels along positive and negative directions respectively, while * stands for both directions. Based on this partitioning, packets can use the channels of either Subnetwork1 or Subnetwork2. In addition packets can move from Subnetwork1 to Subnetwork2 or vice versa but switching is allowed only in one direction at a time to avoid deadlock. The deadlock-freedom is elaborated in Section 5.

In CoBRA, the default transition occurs from Subnetwork1 to Subnetwork2. If all TSVs fail in the eastmost column, the algorithm is reconfigured to route packets first in the Subnetwork2 and then in Subnetwork1. This reverses all the conditions and thus TSV failures are tolerated again as long as there is one healthy TSV in the westmost column.

In more details, CoBRA routing algorithm can be described as follows:

Current and destination are on the same layer:

If the destination is on the east side of the current node, Subnetwork1 is used to deliver the packet (i.e. channels X^+ and $Y0^*$). On the other hand, if the destination is on the west side of the current node, the channels of Subnetwork2 are used (i.e. channels X^- and $Y1^*$).

Current and destination are not on the same layer:

As it was mentioned, the routers are aware of TSV statuses in their columns. If no elevator is found in the column, the packet is forwarded one hop to the east and the process is repeated until a healthy TSV is found. In the worst case, if no TSV is found at the eastmost column, the packet has to be dropped. This implies that a healthy elevator in the eastmost column guarantees the delivery of packets to all destinations no matter how many elevators are available in the network or disabled at runtime. Upon the loss of all TSVs in the eastmost column, CoBRA routing algorithm is reconfigured to deliver the packets toward the west direction. After this reconfiguration, every router forwards the packet one hop to the west if there is no healthy elevator in the same column.

Figure 1 shows a $4 \times 4 \times 2$ network where the source node 20 sends a packet to the destination node 6. Since there is no TSV at the source column, packets are forwarded toward east through Subnetwork1. In the next column (with $X = 1$), there are two available TSVs, located at node 17 and 25. Since the node 6 is in the upper Y-half plane, the elevator located at the node 17 is a better choice to deliver the packet. The routing path is as follows: $20 \rightarrow 21 \rightarrow 17 \rightarrow 1 \rightarrow 2 \rightarrow 6$, or alternatively the packet can take the path $20 \rightarrow 21 \rightarrow 17 \rightarrow 1 \rightarrow 5 \rightarrow 6$. When the source node 10 targets the destination node 21, the elevator at the node 2 is used since the destination is in the upper Y-half plane.

4. Providing Partial Knowledge

Propagation of TSV statuses locally enhances the reliability of 3D-NoC significantly. Providing global information about the location of healthy and faulty elevators in a network may improve the performance but in turn it consumes more resources. In CoBRA, routers in the same column share the TSV statuses with each other. For this purpose, a router is equipped with two signals as it is shown in Figure 2, one transferring the TSV status from north to south (i.e. called signal A) and another one from south to north (i.e. called signal B). Figure 2 represents how these
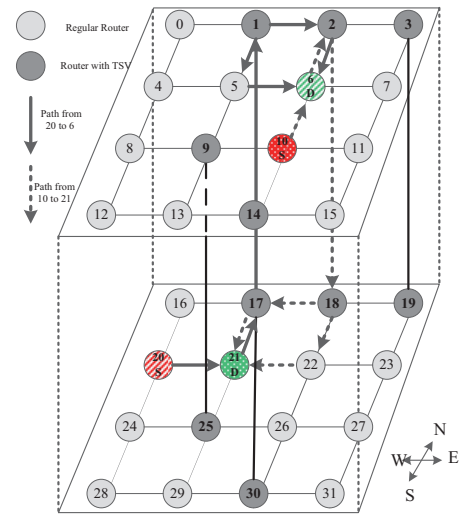

Figure 1: An example of a 3D-NoC

signals have been connected among four routers located in the same column. According to the figure, the signal A reflects the fault information on the north neighbors of a router. If this signal value is one, it means that there is at least one healthy elevator on the north direction of the node. Similarly, the signal B propagates the fault information in the south direction. The signal A and the TSV information of the current router are ORed together to form the signal A that should be sent to the next router. Therefore, if $A = 1$ or the current node has an elevator, the signal A of the next router gets the value of one representing the existence of a healthy elevator in the north direction of a node. The same trend is applied to the signal B.

Figure 2 (a) shows the value of signals when there are two TSVs at the routers 4 and 12. As it is clear, the router at the node 4 does not have any elevator at the north direction while there is one at the south direction which is indicated by the value of the signals $A = 0$ and $B = 1$, respectively. The router at the node 4 does not exactly know where the healthy elevator is located or how many healthy elevators there are on the south direction. The router just knows that there are healthy elevators in the south direction that can be used for vertical transmission.

If a fault disconnects one of the TSVs during runtime, the new status is propagated in the column through the wires. The ORed signals will be updated and routers will adapt themselves to TSV failures. Figure 2 (b) shows the value changes on the signal A when the TSV at the router 4 becomes faulty.

5. Discussion of Deadlock Freedom

We have already mentioned that the network will be deadlock free if packets use either the channels of Subnetwork1 or Subnetwork2 in addition to the possibility of switching from Subnetwork1 to Subnetwork2. Based on subnetwork definition, there are no circular dependencies in each subnetwork. Moreover, when packets are switched from Subnetwork1 to Subnetwork2, there cannot exist any circular path since the direction of moving along X and Z, as well as the virtual channel index along Y, change upon
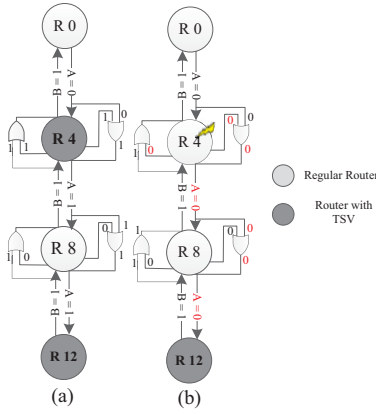
(a)   (b)

Figure 2: Implementation of TSV status propagation.

subnetwork switch. In this section, we use formal methods to verify three properties. The routing logic has to ensure *deadlock-* and *livelock-freedom*. Additionally, even in the presence of faulty TSVs, the routing logic should always be *connected*. In other words, for any pair of source and destination, there must be at least one possible route. All these properties depend on the assumption that between each pair of layers, there is at least one non-faulty TSV.

Let a *configuration* be an assignment of TSVs (faulty or not) to nodes. Each configuration induces a new channel dependency graph as each configuration causes the routing logic to make different choices. Let x, y and z be the dimension of the mesh and let t be the number of TSVs. The total number of configurations is:

$$\sum_{f \leq t} \binom{xyz}{e} \cdot \binom{e}{f} \qquad (1)$$

For example, in a $4 \times 4 \times 4$ mesh with 6 TSVs there are $512,512$ different configurations. It is infeasible to run simulations for all these configurations or perform a manual proof.

To address this issue, we have used DCI2 to formally verify CoBRA for all of the above properties [12]. DCI2 takes as input a model of the routing logic in the form of a function $R :: N \times N \mapsto P$, i.e., a function R that takes as input the current node and the destination node and produces as output the port to which the packet is routed. DCI2 enumerates all configurations and generates the corresponding channel dependency graphs. Based on these graphs, it checks a necessary and sufficient condition for deadlock-free adaptive routing.

We integrated DCI2 and AccessNoxim and instead of analyzing a separate model, DCI2 has been applied to the exact same routing code as was used for AccessNoxim.

When given a $4 \times 4 \times 4$ mesh with 6 TSVs, all $512,512$ configurations are generated. Among all, $208,252$ configurations satisfy the assumption that there is at least one healthy elevator at the eastmost column. All these configurations are formally proven to be deadlock- and livelock-free and to be connected. Other configurations are not considered. We have verified CoBRA for any number of elevators from 0 up to and including 6. The total verification time is about 90 minutes on a 4 core 2 GHz Intel Core i7 machine. Table 1 reports the required verification time for 1, 2, 4, and 6
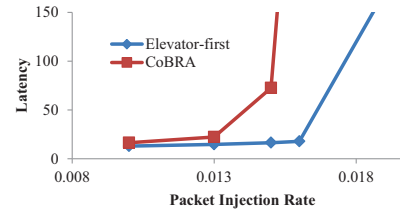

Figure 3: Performance under random traffic for 4 TSVs

elevators. A configuration is an assignment of locations to elevators.

6. Results and Discussion

The efficiency of the proposed routing algorithm under different number of faults has been evaluated using Access-Noxim simulator [13].

A $4 \times 4 \times 4$ mesh NoC has been considered for experiments. All the routers have 5-flit FIFO and the packet size is 8 flits. The simulator is warmed up for 1000 cycles and then the reliability is evaluated over another $20,000$ cycles. The defective TSV is modelled as an open fault. Therefore, if a TSV or a bundle of TSVs are faulty, the entire vertical connection is considered broken.

To evaluate the reliability of the proposed routing algorithm against available routing algorithms, Elevator-first is implemented in AccessNoxim alongside CoBRA. It is necessary to mention that there are few algorithms in literature tolerating faults in partially connected 3D-NoCs. For this reason, the performance of CoBRA cannot be compared with the commonly used routing algorithms, such as XYZ which is proposed for the fully connected 3D-NoCs. The measure of reliability defined in this article is the percentage of flits successfully delivered to the target destinations.

In order to model run-time TSV failures, faults are injected at every 5000 cycles. This value is selected to ensure that the network is stabilized before injecting a new fault. Moreover, results for different traffic patterns including synthetic and real traffic scenarios are reported [14] [15].

Three architectures have been used to evaluate the efficiency of the CoBRA routing algorithm. The first one has four elevators at four corners located at nodes 0, 3, 12 and 15 based on the numbering given in Figure 1. The second one has eight TSVs at nodes 0, 2, 5, 7, 8, 10, 13 and 15. Finally, the third one has five TSVs located at nodes 0, 2, 7, 8 and 10. Figures 3, 4 and 5 illustrate the latency comparison for the fault-free Elevator-first and CoBRA routing algorithms for the three architectures under random, real traffic, and shuffle, respectively.

According to Figure 3, under the uniform random traffic pattern and by the availability of four TSVs, Elevator-first outperforms CoBRA. According to Figure 4, CoBRA and Elevator-first perform relatively close under the real traffic. As it is clear in Figure 5, CoBRA outperforms Elevator-first under shuffle traffic if there are five elevators in the network at nodes 0, 2, 7, 8 and 10. Therefore, the number and the location of elevators affect the performance of CoBRA and Elevator-first.

Number of elevators	Number of faults	Total number of config	Number of eastmost config	Verification time (Sec)
1	0	16	4	2
2	0	120	54	3
	1	240	60	3
4	0	1820	1325	28
	1	7280	4420	93
	2	10920	4914	113
	3	7280	1280	46
6	0	8008	7084	127
	1	48048	39336	640
	2	120120	87450	1355
	3	160160	97240	1655
	4	120120	54054	980
	5	48048	12012	248

TABLE 1: Verification results
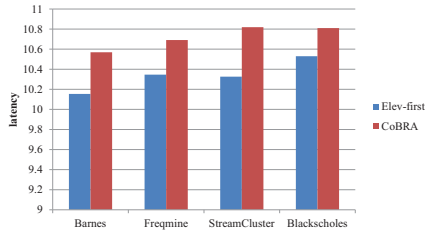
Figure 4: Latency comparison under real traffic for 8 TSVs

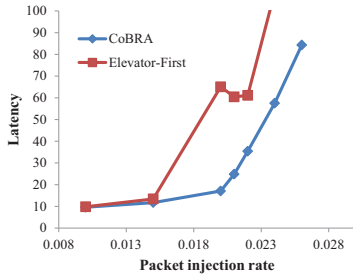Figure 6: Reliability under single faults for 4 TSVs

Figure 5: Performance under shuffle traffic for 5 TSVs

6.1. Reliability Comparison under Synthetic Traffic

The reliability comparison for the architecture with four TSVs at four corners have been represented in this section. For this architecture, the effects of single, double and triple faults have been assessed. According to the results (Figure 6), CoBRA provides full reliability in the presence of a single fault. The reliability of Elevator-first drops to nearly 85% and 70% under random and transpose traffic, respectively. Since the transpose traffic is based on vertical transmission for every pair of source and destination, a single fault has more severe effect on this traffic. As it is clear, Elevator-first can not adapt itself to faults at runtime.

The effect of changing the location of double faults is presented in Figure 7. CoBRA is fully fault-tolerant as long as there exists one healthy TSV at the eastmost column. When both TSVs in the eastmost column fail, CoBRA is reconfigured to switch from Subnetwork2 to Subnetwork1. By routing packets to the west, CoBRA will be able to tolerate faults as long as there exists at least one healthy TSV in the westmost column. For all of the presentend fault scenarioa, the reliability of Elevator-first falls within 47% to 78%.
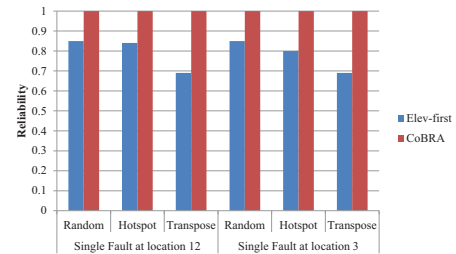
Three failure scenarios have been considered in the Figure 7, where in each scenario two TSVs have been disconnected:

1) Faulty TSVs at nodes 0 and 12: under this scenario CoBRA supports full reliability since it dynamically seeks for the healthy elevators by forwarding the packet to the east direction.
2) Faulty TSVs at nodes 3 and 15: In this case the reliability of CoBRA decreases considerably as no elevator can be found in the east direction. The reconfiguration, to the contrary, provides full reliability due to the existence of healthy elevators in the westmost column. It should be noted that some packets are dropped in the reconfiguration phase until the network backs to its stable condition again. Elevator-first drops 30% of flits under this condition.
3) Faulty TSVs at nodes 0 and 15: CoBRA provides full reliability because of one healthy elevator at the node 3.

Figure 8 illustrates the effect of triple faults. According to this figure, only one healthy elevator located at the node 15 guarantees the delivery of all packets to destinations in CoBRA while Elevator-first delivers only 45% and 15% of packets to destinations under random and transpose traffic respectively. Moreover, triple faults at locations 0, 3 and 15 provide a reliability of 50% for the Elevator-first. On the other hand, no healthy elevator at the eastmost column drops the reliability by 55% and 70% for the random and transpose traffic in CoBRA, respectively compared to full reliability support in the first triple fault scenarios. Again, reconfiguration solves the problem.
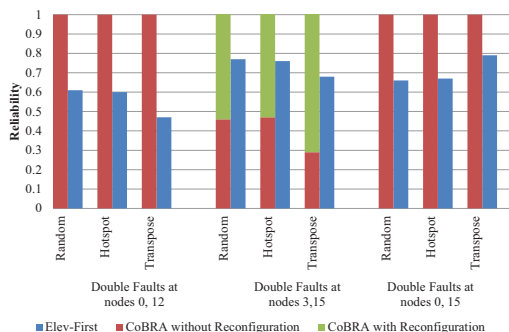
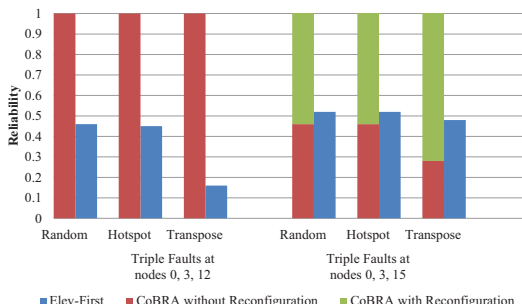Figure 7: Reliability under double faults for 4 TSVs

Figure 8: Reliability under triple faults for 4 TSVs

6.2. Reliability Comparison under Real Traffic

The fault tolerance of CoBRA versus Elevator-first for single, double and four faults have been evaluated for the real traffic Barnes and Freqmine. The Streamcluster and Blackscholes perform relatively close to Freqmine so they have been omitted due to the lack of space. Based on Figure 9, CoBRA delivers all the packets to destinations unless double faults are located at nodes 7 and 15. Figure 10 illustrates the reliability comparison under 4 faults.

6.3. Power and Area Comparison

In this section, the power consumption of CoBRA and Elevator-first are compared under different fault scenarios for the random traffic. The power reports are extracted from AccessNoxim which accumulates energy upon flit reception/transmission at a router.

Table 2 reveals that the power consumption of CoBRA is higher than Elevator-first in most cases. This is because
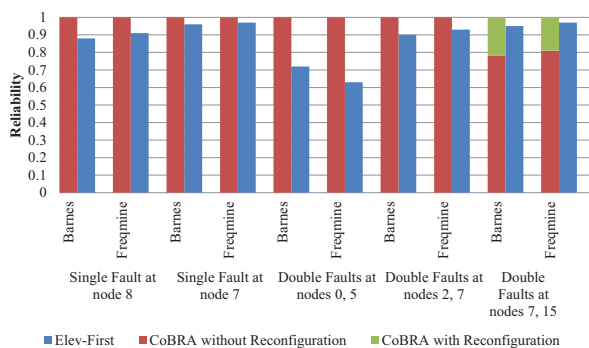

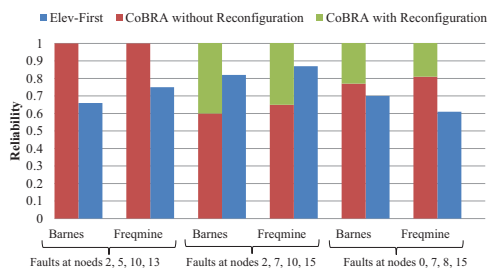Figure 9: Single and double faults comparison for 8 TSVs

Figure 10: Reliability comparison for 8 TSVs

Elevator-first drops packets when a TSV is faulty. Therefore, network congestion reduces and the power consumption decreases as well. It is in contrast with CoBRA where packets are rerouted when they encounter a faulty TSV. In a fault-free network, the energy consumption of CoBRA might be higher or lower than Elevator-first depending on the number and location of TSVs. It is worth noting that Elevator-first consumes more static power due to the extra buffers in the north and south input ports. However, static power consumption is not considered in the results as it is independent of the routing algorithm.

The packet injection rates for the fault scenarios are adjusted to be on the saturation threshold. It can be observed from the table that these PIRs are different for each fault scenario. Specifically, the PIRs of the first (faults at 0 and 15) and the third fault scenario (faults at 0, 3 and 12) are noticeably lower that the PIR of the second scenario (faults at 3 and 15). Also, the PIR of the third faulty scenario (faults at 0, 3 and 12) is slightly lower than the PIR of the first case (faults at 0 and 15). This observation is based on the fact that for example when faults are presented at 3 and 15, the drop rate is higher compared to the other two cases, which allows the nodes to inject more packets into the network before reaching the saturation point.

Regarding the area, Elevator-first occupies relatively larger area compared to CoBRA since it uses two virtual channels along the X and Y dimensions while CoBRA has just two virtual channels along the Y dimension.

6.4. Temperature Distribution

The traffic-thermal mutual-coupling cosimulation platform [16] has been used to compare the thermal distribution of CoBRA and Elevator-first. The tile geometry and power model are based on Intel 80-core chip [17]. The physical floorplans and power traces (generated during the network traffic simulation) are used as inputs of thermal simulation.

The thermal distribution of the two routing algorithms are compared for a $4 \times 4 \times 4$ mesh NoC under the shuffle traffic whose latency comparison was shown in Figure 5. The thermal distribution is extracted under the injection rate of 0.022. Based on Figure 11, the high temperature nodes are located in the east side of the network for CoBRA, while they are located in the left for Elevator-first. The distribution of hot spot nodes, which represent points of higher traffic, is determined by the routing algorithm and the method by which elevators are assigned to different nodes.

|	Random Traffic					
	Average power (Whole network) ($\mu J/cycle$)			Average power per router ($nJ/cycle$)		
Routing algorithm	Double Faults 0, 15 PIR = 0.007	Double Faults 3, 15 PIR = 0.017	Triple Faults 0, 3, 12 PIR = 0.006	Double Faults 0, 15 PIR = 0.007	Double Faults 3, 15 PIR = 0.017	Triple Faults 0, 3, 12 PIR = 0.006
CoBRA without Reconfiguration	1.68	2.22	1.52	26.23	34.6	23.8
Elevator-first	1.15	2.78	0.9	18	43.5	14.12

TABLE 2: Power consumption comparison
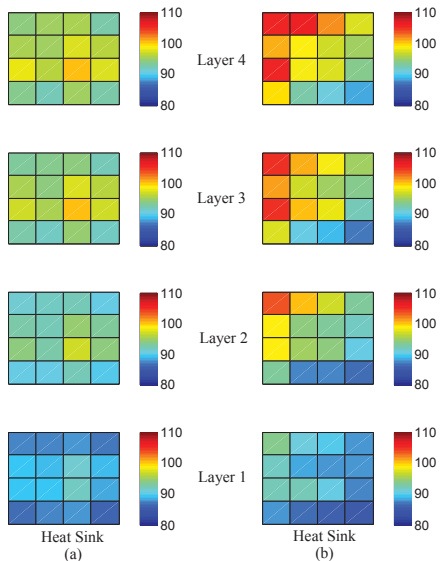
Figure 11: Thermal distribution a) CoBRA b) Elevator-first

7. Conclusion

TSVs are applied to stack multiple layers of 2D-NoCs to provide fast and power efficient 3D architectures. The vulnerability of TSVs during manufacturing process makes these interconnections susceptible to faults. TSV failures might occur during or after manufacturing process. Either the TSV yield should be increased or chips with faulty TSVs should be discarded but both solutions are costly. We proposed a reconfigurable routing algorithm, called CoBRA, to tolerate TSV failures during runtime and manufacturing process. First, the routing algorithm dynamically searches for a healthy elevator in the same column. If no TSV is found, the packet moves to east and and if the packet reaches the eastmost column and fails to find a healthy TSV, the network is reconfigured to find an elevator at the west direction. Simulation results indicate that CoBRA enhances reliability considerably compared to Elevator-first.

References

[1] W. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. Sule, M. Steer, and P. Franzon, "Demystifying 3d ics: the pros and cons of going vertical," *Design Test of Computers, IEEE*, vol. 22, no. 6, pp. 498–510, Nov 2005.

[2] I. Loi, F. Angiolini, S. Fujita, S. Mitra, and L. Benini, "Characterization and implementation of fault-tolerant vertical links for 3-d networks-on-chip," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 1, pp. 124–134, Jan 2011.

[3] C. Metzler, A. Todri, A. Bosio, L. Dilillo, P. Girard, and A. Virazel, "Through-silicon-via resistive-open defect analysis," in *Test Symposium (ETS), 2012 17th IEEE European*, May 2012, pp. 1–1.

[4] K. Chakrabarty, S. Deutsch, H. Thapliyal, and F. Ye, "Tsv defects and tsv-induced circuit failures: The third dimension in test and design-for-test," in *Reliability Physics Symposium (IRPS), 2012 IEEE International*, April 2012, pp. 5F.1.1–5F.1.12.

[5] K. Tu, "Reliability challenges in 3d ic packaging technology," *Microelectronics Reliability*, vol. 51, no. 3, pp. 517–523, 2011.

[6] A. Eghbal, P. Yaghini, N. Bagherzadeh, and M. Khayambashi, "Analytical fault tolerance assessment and metrics for tsv-based 3d network-on-chip," *Computers, IEEE Transactions on*, vol. 64, no. 12, pp. 3591–3604, Dec 2015.

[7] M. Ebrahimi, X. Chang, M. Daneshtalab, J. Plosila, P. Liljeberg, and H. Tenhunen, "Dyxyz: Fully adaptive routing algorithm for 3d nocs," in *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, Feb 2013, pp. 499–503.

[8] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, P. Liljeberg, and J. Plosila, "Q-learning based congestion-aware routing algorithm for on-chip network," in *Networked Embedded Systems for Enterprise Applications (NESEA), 2011 IEEE 2nd International Conference on*, Dec 2011, pp. 1–7.

[9] M. Ebrahimi, M. Daneshtalab, and J. Plosila, "Fault-tolerant routing algorithm for 3d noc using hamiltonian path strategy," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, March 2013, pp. 1601–1604.

[10] X. Jiang and T. Watanabe, "A novel fully adaptive fault-tolerant routing algorithm for 3d network-on-chip," in *TENCON 2013 - 2013 IEEE Region 10 Conference (31194)*, Oct 2013, pp. 1–4.

[11] F. Dubois, A. Sheibanyrad, F. Petrot, and M. Bahmani, "Elevator-first: A deadlock-free distributed routing algorithm for vertically partially connected 3d-nocs," *Computers, IEEE Transactions on*, vol. 62, no. 3, pp. 609–615, March 2013.

[12] F. Verbeek and J. Schmaltz, "A decision procedure for deadlock-free routing in wormhole networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 1935–1944, 2014.

[13] http://access.ee.ntu.edu.tw/noxim/index.html.

[14] J. P. Singh, W.-D. Weber, and A. Gupta, "Splash: Stanford parallel applications for shared-memory," *ACM SIGARCH Computer Architecture News*, vol. 20, no. 1, pp. 5–44, 1992.

[15] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM, 2008, pp. 72–81.

[16] K.-Y. Jheng, C.-H. Chao, H.-Y. Wang, and A.-Y. Wu, "Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip," in *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, April 2010, pp. 135–138.

[17] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-ghz mesh interconnect for a teraflops processor," *Micro, IEEE*, vol. 27, no. 5, pp. 51–61, Sept 2007.