FT-Z-OE: A Fault Tolerant and Low Overhead Routing Algorithm on TSV-based 3D Network on Chip Links

Hoda Naghibi Jouybari
College of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

Karim Mohammadi
College of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

ABSTRACT

Three-dimensional Network-On-Chips (3D NOC) are the most efficient communication structures for complex multi-processor System-On-Chips (SOC). Such structures utilize short vertical interconnects in 3D ICs together with scalability of NOC to improve performance of communications in SOCs. By scaling trends in 3D integration, probability of fault occurrence increases that leads to low yield of links, especially TSV-based vertical links in 3D NOCs. In this paper, FT-Z-OE (Fault Tolerant Z Odd-Even) routing, a distributed routing to tolerate permanent faults on vertical links of 3D NOCs is proposed. FT-Z-OE is designed to have low overhead because of no need to any routing table or global information of faults in the network. The proposed routing is evaluated using a cycle-accurate network simulator and compared to planar-adaptive routing for a 3D mesh-based network. It is shown that FT-Z-OE significantly outperforms planar-adaptive in the terms of latency and throughput under synthetic traffic patterns.

General Terms

Embedded Systems, Network on Chips, Routing Algorithm, Fault Tolerance.

Keywords

Three Dimensional Network on Chip, 3D Integrated Circuits, System on Chip, Fault Tolerant Routing, TSV-based Links.

1. INTRODUCTION

Due to the exponential growth of circuit integration and increasing complexity of System on Chips (SOC), Network on Chip (NOC) has been proposed as scalable and efficient on chip communication architecture for SOCs [1, 2].

Based on ITRS prediction, improvement in integrated circuits in deep sub-micron technologies is being saturated, so moving to novel technologies like 3D integration is essential. Using wafer bonding process 3D ICs will be manufactured by stacking of multiple active layers connected with vertical Through Silicon Via (TSV). TSVs are shorter than horizontal links, so in contrast to traditional 2D ICs, currently emerging 3D integration technology provides short interconnects, improves performance and packing density of chip and reduces on-chip power consumption [3]. Short vertical interconnects in 3D ICs together with scalability of network on chip has led to appearance of 3D NOCs that has been most efficient interconnection structure for current and future high performance SOCs.

On the other hand, scaling trends in integrated circuits have led to some problems such as manufacturing defects, misalignment of TSVs, increasing of heat, random open defects and some other faults that cause system failure. A promising solution for increasing the reliability of system is the use of design methods which introduces tolerance against possible faults in an integrated circuit. In NOC based SOCs, fault tolerant routing is essential to increase reliability of connections between devices on a chip.

As low TSV yield is one of the most important challenges in current 3D fabrication process [4], in this paper, a routing algorithm is proposed for 3D mesh NOC with the ability of fault tolerance of permanent fault on TSV based vertical links.

The rest of paper is organized as follows. In section 2 prior works on 2D and 3D fault tolerant routing are reviewed. Section 3 describes the proposed routing algorithm in details. Simulation results are presented and analyzed in section 4 and finally section 5 offers conclusion.

2. RELATED WORKS

Routing algorithm is one of the most important aspects of NOC design. Existing works have proposed many fault tolerant routing algorithms for 2D NOCs. In this section, some of important ones are reviewed in 3 categories [12].

1. Stochastic routing algorithms such as probabilistic gossip flooding [5], directed flooding and N random walk [6] tolerate faults through redundancy of data and probabilistically replication of packets and sending them over different paths [7]. But these routings have the drawback of high energy consumption, high storage and computational overhead and high probability of livelock and deadlock.

2. Fully adaptive routings such as [8,9] require routing table in each router to reflect the runtime state of NOC and need to update tables based on fault occurrence and route information frequently. These algorithms need thousand cycles to access to global fault and route information to support runtime faults. Also they lose scalability (especially in large size NOCs), increase energy consumption and difficulty to avoid deadlock and livelock.

3. Partial adaptive fault tolerant routing schemes that are based on turn models (north-last, west-first, negative-first and Odd-Even) [10, 11] that restrict special turns to avoid deadlock situations. For example [12] supports fault tolerance through packet duplication in high fault rates and sends each of them based on one turning model (Odd-Even and inverse Odd-Even). A few routings use convex or rectangular faulty regions to tolerate faults using turn models [13-18]. Main drawback of these schemes is disabling some healthy components of network to build convex faulty blocks, besides they have limitation on fault location and shape of faulty regions.

As mentioned above, all of these routings have been proposed for 2D NOCs, but just a few fault tolerant routings are presented to support 3D NOCs. Planar-adaptive routing [19] makes use of adaptation in 2D plains and avoids deadlock through three virtual channels (VC). This method needs to build convex faulty regions, so some healthy nodes will be disabled. In [20] intermediate nodes are used to tolerate faults and a table in each router is required to find intermediate node. Routing scheme in [21] uses faulty cube to tolerate faults. HamFA routing scheme [22] utilizes Hamiltonian Path to tolerate unidirectional one-faulty links without virtual channels.

Most of proposed schemes have large overhead (virtual channels or large routing tables) and complexity that is not suitable for on-chip communication.

On the other hand, existing 3D routings usually consider horizontal and vertical links the same way, but actually probability of fault occurrence in TSV based links is more than horizontal links due to low yield of TSVs. So the proposed algorithm focuses on permanent faults over vertical links. In authors' knowledge just [23] addressed vertical links to tolerate faults, but it also needs some global information about the faulty links of other routers in each router, so some overhead is added to the routers to store fault information. Besides, to support runtime permanent faults, this information should be updated frequently that takes some time during which network may be unstable.

FT-Z-OE routing can simply tolerate faults on TSV-based vertical links without using redundant packets, routing table or global information of the network and location of all faulty links (each router only must be aware about its own faulty links). So it can support both design time and runtime permanent fault. Besides it can avoid deadlock by using only one extra virtual channel for bidirectional faults and without any VCs for the case that all of faults have the same direction.

3. FAULT TOLERANT ROUTING ALGORITHM

In this section, the proposed fault tolerant routing algorithm, FT-Z-OE (Fault Tolerant Z Odd-Even) is described that is designed to tolerate permanent faults on TSV based vertical links. Then deadlock freeness of the algorithm is explained in details.

3.1 Preliminaries

A Three-Dimensional mesh NOC has N^3 nodes which N is the number of nodes in every dimension. A node is identified by its coordinate in these three dimensions (node_x, node_y, node_z) and has six neighbors in six directions of up, down, north, south, east and west. Figure 1 demonstrates these six directions that are assumed in this paper. Each link is a bidirectional channel that consists of two unidirectional physical channels.

Permanent faulty links are assumed as completely broken and disabled to transfer data. When one direction of a link is faulty, another direction works properly. Although in the case of bidirectional faulty links, both direction of a link are broken.

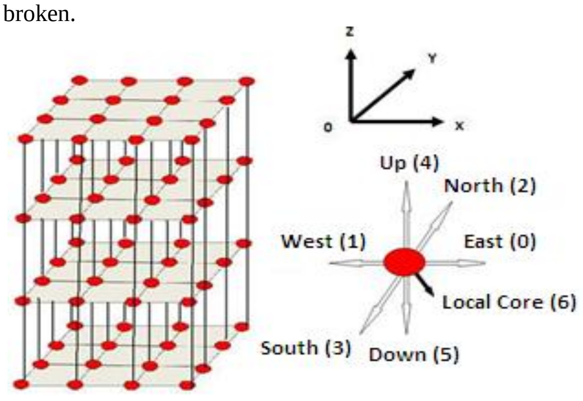


Fig.1: A 3D mesh with six directions

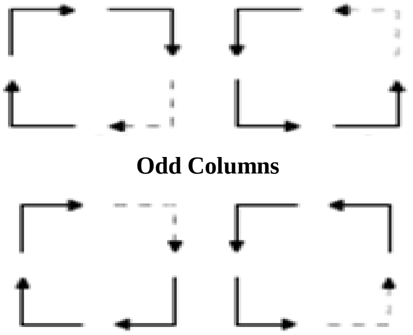
3.2 FT-Z-OE Routing

Odd-Even turn model for partial adaptive routing was proposed by Chiu [11]. The Odd- Even turn model applies two rules to restrict the locations at which certain turns can occur to ensure deadlock freedom in 2D routing:

Rule 1: Any packet is not allowed to take an EN (East-North) turn at any nodes located in an even column, and it is not allowed to take an NW (North-West) turn at any nodes located in an odd column.

Rule 2: Any packet is not allowed to take an ES (East-South) turn at any nodes located in an even column, and it is not allowed to take an SW (South-West) turn at any nodes located in an odd column. (Figure 2)

This model is used in proposed algorithm to route packets in each 2D layer of network.



Odd Columns

Even Columns

Fig.2: The allowed (solid lines) and forbidden (dashed lines) turns in Odd-Even turn model

As shown in Figure 3, in the absence of fault(s), FT-Z-OE first routes packets in Z direction (if necessary), according to relative address of source and destination nodes, until it reaches to the layer of destination node. Then using Odd-Even turn model the packet moves toward destination node.

```
// current router (cur_x,cur_y,cur_z)
// destination router ( dest_x,dest_y,dest_z)
if (cur_x=dest_x and cur_y=dest_y and
    cur_z=dest_z)
  Deliver the packet to local node and exit;
else if (cur_z = dest_z)
  goto 2D routing toward destination node;
else if (cur_z  < dest_z){
  if(upward link is not faulty){
    out_port = 4; // upward link
    misrouting bit = 0; }
  else if ((upward link is faulty and
          dest_y=cur_y and dest_x=cur_x) or
          misrouting bit=1){
    misrouting bit=1;
    if (current node is not on the west edge of
       mesh)
      out_port=1;    // to the west
    else if ( current node is on the west edge
           and not on the north edge of
           mesh)
      out_port=2;   //to the north
    else if ( current node is on north-west
             corner of mesh)
      out_port=0;}  //to the east

  else if (upward link is faulty and misrouting
          bit =0)
   goto 2D routing toward (dest_x,
   dest_y,cur_z);} /* to the image of
      destination node in
      the current layer */
else if (cur_z  > dest_z){
  if(downward link is not faulty){
    out_port = 5; // downward link
    misrouting bit = 0;}
  else if ((downward link is faulty and
             dest_y=cur_y and  dest_x=cur_x) or
            misrouting bit=1){
    misrouting bit=1;
    if (current node is not on the west
            edge of mesh)
      out_port=1;    // to the west
    else if ( current node is on the west edge
           and not on the north edge of
           mesh)
      out_port=2;   //to the north
    else if ( current node is on north-west
             corner of mesh)
      out_port=0;}  //to the east
  else if (downward link is faulty and
           misrouting bit=0)
      goto 2D routing toward (dest_x,
      dest_y,cur_z);} /* to the image of
           destination node in
           the current layer*/
2D routing:
Preferred ports based on (dest_x,dest_y ) and
(cur_x,cur_y  and    Odd-Even    model   are
determined.
if(the number of preferred ports is one)
    out_port= preferred port;
else if (the number of preferred ports is two)
    out_port  =  the  one  that  has  the  less
congestion in the buffer of downstream router;
```

Fig. 3: FT-Z-OE routing algorithm

In Odd-Even routing when two output ports are preferred, proposed algorithm selects an output port due to proximity congestion information defined in [24] (the number of occupied cells in all input buffers of the downstream router).

In the presence of fault(s), the packet first moves in the Z direction toward destination layer, when it encounters to a faulty link in this direction, it is routed in that layer toward the image of destination node in that layer, based on Odd-Even model (and according to congestion information of neighbors in the case of two preferred ports) until it reaches to a router that has safe vertical link toward the destination layer. Then it continues moving in Z direction until the packet reaches to destination layer. At this time packet is routed to the destination node using Odd-Even model. Note that image of router A in a specified layer is router B which has the same X and Y coordinate as router A in that specified layer.

In the FT-Z-OE algorithm there is an exception, that is when source (or current node) and destination nodes has the same X and Y coordinate but are in different layers (in other words source and destination coordinates differs only in Z). In this case, when vertical links toward destination is faulty, as the image of destination node in the current layer is current node, Odd-Even cannot be used and misrouting should be done to reach a node with safe vertical link toward destination.

In the misrouting phase, FT-Z-OE algorithm routes packet to the west neighbor of current router (except for the routers on the west edge of mesh that should route packets to the north neighbor and at the north-west corner of mesh, packets should be routed to east direction). If this node has faulty vertical link to destination layer too, packet will be routed to west neighbor of this router again. This method will be continued until the packet reaches nearest node with safe vertical link. From this node normal routing of FT-Z-OE is done. Note that WN (West-north) turn may occur in the west edge of mesh in the misrouting phase and as the west edge is always an even column (column 0), this turns is allowed based on Odd-Even turn model. So there will be no cycle dependency in 2D layers of network.

Figure 4 shows an example of routing packet using FT-Z-OE algorithm. Node A is source and node I is the destination of a packet. In node A because of faulty upward link, FT-Z-OE goes to misrouting phase, after two hops routing packet to the west, it is routed to the D node and normal routing of FT-Z-OE is done until the packet is delivered to the destination node.

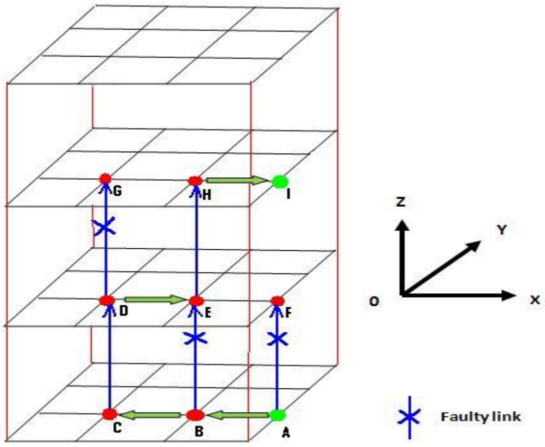


Fig.4: An example of FT-Z-OE routing

To implement misrouting, one bit (named misrouting bit) is considered in the header flit of packet that is 0 in the normal routing, but in that exceptional case, this bit changes to 1 and as long as misrouting bit is 1, the packet can only move to the west direction until that reaches to a node with safe vertical link. At this time the misrouting bit changes to 0 again. This misrouting bit helps to avoid backtracking that may lead to livelock situations.

3.3 Deadlock and Livelock Freeness

As mentioned, FT-Z-OE uses Odd-Even turn model to route packets in 2D layers, hence no deadlock cycle will be occurred in routing packets in 2D layers of network.

In the case of unidirectional faulty links (only upward or downward links are faulty) there is no cycle between layers too. For example in the case of just upward faulty links, packets may have XZ^+ , YZ^+, Z^+X , Z^+Y, Z^-X, Z^-Y turns, but to create a cycle between layers XZ^- or YZ^- turns are needed that do not occur in FT-Z-OE algorithm in such case. The similar discussion is correct for just downward faulty links. Note that the notation of XZ^+ is defined as West-Up or East-Up turns and other notations will be defined the same way according to Figure 1.

To avoid cycle creation between layers in the case of two directional faulty links (upward and downward), FT-Z-OE uses only one additional virtual channel. One VC is considered for routing packets that are moving upward (destination_z > cur_z) and another for moving downward packets (destination_z < cur_z). Hence in the case of multiple faulty vertical links (upward and downward links), there is no cycle in the network by using two virtual channels.

Since minimal routing is used in the normal state of FT-Z-OE toward destination, livelock situations will be avoided. In misrouting phase by using of misrouting bit (as discussed in previous section), any backtracking is prevented and after moving just a few hops away the minimal path, normal and minimal routing will be done to the destination node.

4. SIMULATION RESULTS

4.1 Simulation Setup

The performance of FT-Z-OE routing algorithm is evaluated using Booksim [25], a cycle accurate network simulator. FT-Z-OE routing and planar-adaptive routing scheme are compared for a 4-ary 3-mesh network in the terms of average packet latency, throughput and power consumption under synthetic traffic patterns. In simulations, three VCs are considered with the buffer size of 5 flits, and each injected packet consists of 5 flits. The simulations are based on wormhole switching. One cycle for each of routing delay, VC allocation delay, switch allocation delay and vertical link delay is considered. For power estimation, dynamic power model for NOC components from Orion 2.0 [26] is incorporated into the simulator. Clock frequency of 1GHz and technology of 65nm is assumed.

Each simulation was initially run for 20,000 cycles to allow stabilization of network and then it was executed for 40,000 cycles to measure packet latency and throughput. Faulty links were inserted randomly in each simulation and the same set of random faults was used for both algorithms every time. For each simulation of network under N faulty links, to be sure that simulation results are independent of location of faulty links, 20 different combinations of N faults (but the same for two algorithms) were considered and the results are the average of these 20 scenarios.

4.2 Performance Evaluation

FT-Z-OE and planar adaptive routings are evaluated in term of average packet latency. Figure 5 demonstrates the average packet latency of FT-Z-OE and planar-adaptive routings in the fault free, 1 fault (1% of all vertical links) and 5 faults (5% of all vertical links) situations under uniform, bit-complement and transpose traffic patterns. FT-Z-OE outperforms planar-adaptive routing in all situations and all traffic patterns.

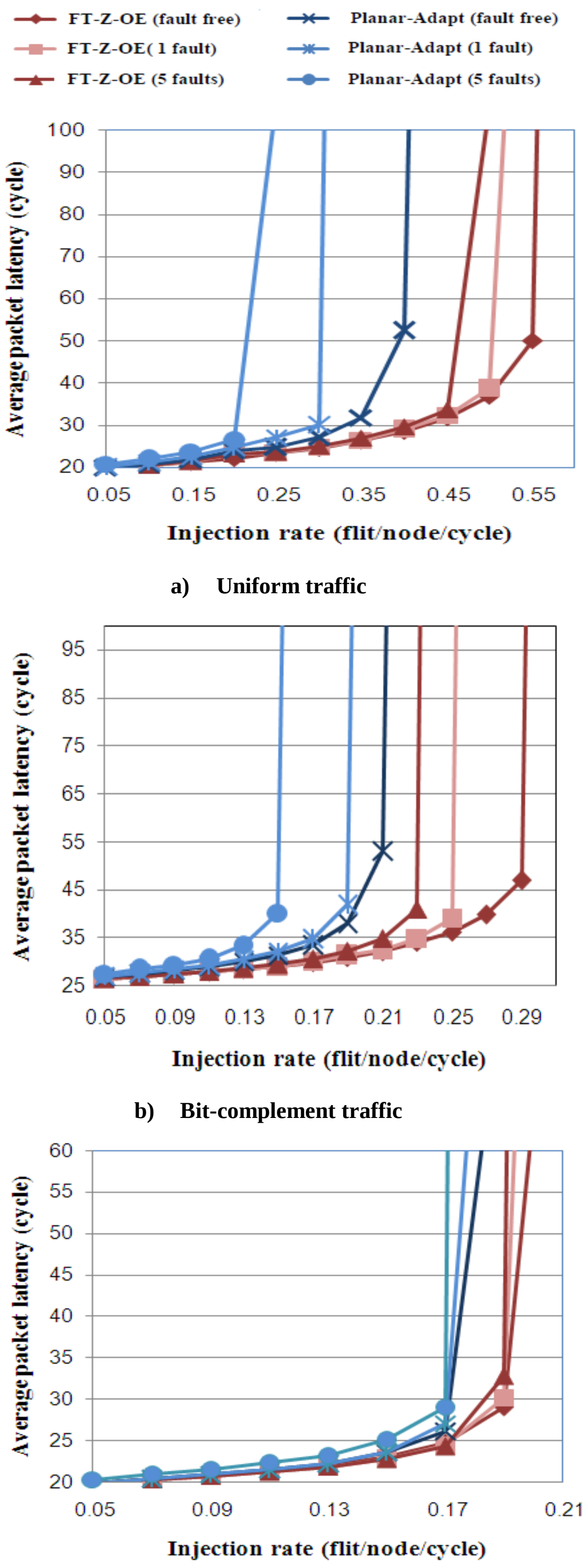
In uniform traffic, FT-Z-OE improves saturation injection rate by 37% (from 0.4 to 0.55 flit/node/cycle), 66% and 125% for 0 fault, 1 fault and 5 faults respectively. In bit-complement traffic, we see improvement of 38%, 31% and 53% in these three situations and finally Figure 5-c demonstrates improvement in saturation injection rate by 11% for all situations of fault free, 1 fault and 5 faults situations under transpose traffic.

As mentioned, FT-Z-OE does not need any VC to avoid deadlock in the case of one directional faults, so each packet can use any of VCs and uses the VCs to improve performance rather than deadlock avoidance. However, planar-adaptive uses all three VCs to avoid deadlock. So as is shown in Figure 5, FT-Z-OE provides less latency and significant improvement of saturation injection rate in contrast to planar-adaptive routing.

Figure 6 shows average throughput across all nodes of network with respect to injection rate for FT-Z-OE and planar-adaptive routings under uniform traffic. Note that throughput evaluation in this simulation is based on throughput definition in [27] that is accepted traffic (the rate at which packets are delivered to the destination by routing algorithm). According to Figure 6, FT-Z-OE algorithm shows better throughput compared with planar-adaptive routing.

As packets will not be dropped in FT-Z-OE and planar-adaptive routing anyway, so before saturation point, routing algorithms will deliver all packets to their destination and throughput is almost equal to injection rate. But after the network is saturated, some packets will be remained in the buffers of routers and cannot move toward destination. Therefore throughput is constant in high injection rates when the network is saturated. As shown in Figure 6, FT-Z-OE routing provides much throughput in high injection rates compared to planar-adaptive routing, in the situations of fault free, 1 fault and 5 faults.

Figure 7 and Figure 8 respectively demonstrate average packet latency and dynamic power consumption of FT-Z-OE and planar-adaptive routing algorithms with respect to fault rate in injection rate of 0.05 (flit/node/cycle). As fault rate increases, number of hops that a packet takes, increases. Correspondingly, latency and power consumption increase. As it is shown in Figure 7 and 8, there is very slight increasing of latency and power consumption in FT-Z-OE algorithm because FT-Z-OE utilizes minimal paths to route packets toward destination as much as possible and just in the case of no safe minimal path, after just a few hops misrouting, returns to minimal path. So there is small increase of average number of hops in FT-Z-OE by increasing fault rate. Correspondingly, latency and power consumption increase slightly. For example when fault rate increases from 1% to 20%, we can see increasing of 44(mW) in power consumption of planar-adaptive routing but just 4(mW) for FT-Z-OE routing.



a) Uniform traffic

b) Bit-complement traffic

c) Transpose traffic

Fig. 5: Average packet latency of FT-Z-OE and planar adaptive routing under synthetic traffic patterns

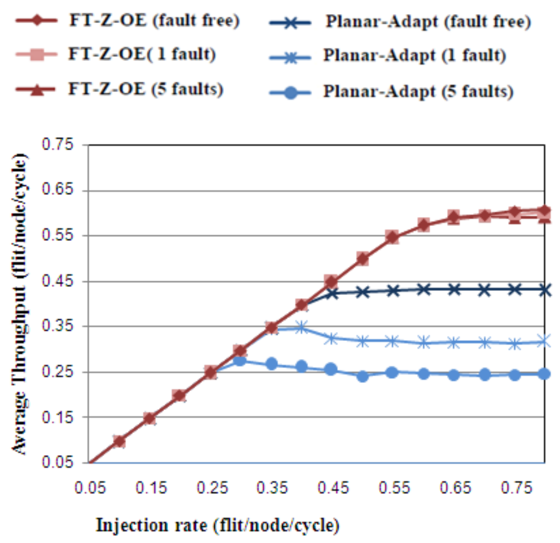


Fig. 6: Average throughput of FT-Z-OE and planar adaptive routings under uniform traffic

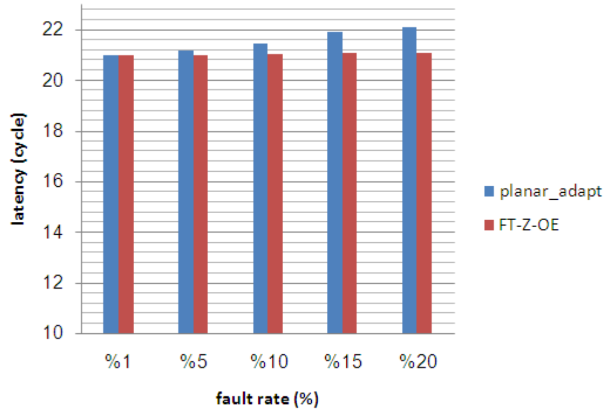


Fig. 7: Average packet latency of FT-Z-OE and planar adaptive routings under uniform traffic

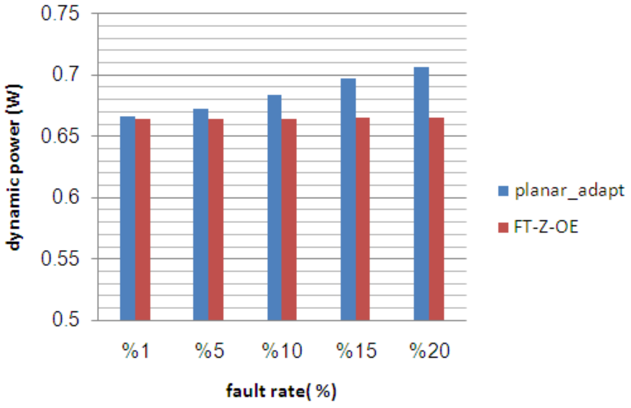


Fig. 8: Dynamic power of FT-Z-OE and planar adaptive routings under uniform traffic

Note that dynamic power is the power that is consumed in five phases of: writing in buffers, crossbar, physical links, switch arbiter and virtual channel arbiter. Table 1 demonstrates dynamic power consumption of every router components in different fault rates under uniform traffic and injection rate of 0.05 (flit/node/cycle). This low injection rate is considered to avoid saturation in high fault rates in planar-adaptive routing. So power increases slightly by increasing fault rates. Based on table 1, Crossbar, buffer and link power are more affected by increasing fault rate. The latency-power product that gives an approximation of energy consumption of routing algorithms, for FT-Z-OE algorithm is less than planar-adaptive routing too.

As mentioned, FT-Z-OE has the ability of tolerating unidirectional faults without VC, like HamFA routing scheme [22]. Also FT-Z-OE has the ability of tolerating bidirectional faulty links by allocating one virtual channel. FT-Z-OE can tolerate not only one fault situations (just vertical links) with 100% reliability, but also multi fault situations with high reliability. Table 2 shows reliability of FT-Z-OE with respect to number of faults in network. 10,000 iterations with random inserted vertical faulty links were considered for calculating reliability of network. A network is reliable if all of injected packets reach to their destination successfully. Based on this definition, in this paper, reliability is defined as number of successfully reached packet divided by total injected packets. As Table 2 demonstrates, for example in the case of 2 faults in network, FT-Z-OE can deliver 98% of injected packets to their destinations successfully.

Table 1: Dynamic power of router components of FT-Z-OE and planar adaptive routings under uniform traffic

Fault		Crossbar dynamic power	Buffer dynamic power	Switch allocator dynamic power	VC allocator dynamic power	Link dynamic power	Total dynamic power(W)
1%	FT-Z-OE	0.145	0.195	0.043	0.061	0.22	0.664
	Planar	0.146	0.196	0.043	0.061	0.22	0.666
5%	FT-Z-OE	0.145	0.196	0.043	0.061	0.22	0.665
	Planar	0.148	0.198	0.043	0.061	0.22	0.671
10%	FT-Z-OE	0.146	0.196	0.043	0.061	0.22	0.666
	Planar	0.15	0.203	0.043	0.061	0.23	0.687
15%	FT-Z-OE	0.146	0.196	0.043	0.061	0.23	0.667
	Planar	0.15	0.204	0.043	0.061	0.24	0.698
20%	FT-Z-OE	0.146	0.196	0.043	0.061	0.23	0.668
	Planar	0.151	0.205	0.043	0.061	0.25	0.71

Table 2: Reliability of FT-Z-OE routing under uniform traffic

Number of vertical faulty link	Reliability of FT-Z-OE
1	100%
2	98%
3	95%
5	91%

5. CONCLUSION

In this paper FT-Z-OE, a low overhead and fault tolerant routing is presented that needs no routing table to store global fault and route information and no restriction on fault location and shape. FT-Z-OE is a distributed routing in which every router decides about next hop of packet, so computational overhead is reduced. This algorithm has the ability to tolerate design time and runtime permanent faults on TSV based vertical links in 3D mesh network on chip, with high reliability for both one fault situations and multi fault situations. Simulation results demonstrate significant improvement of FT-Z-OE algorithm in terms of latency, saturation injection rate and throughput compared to planar-adaptive routing in fault free, single fault and multiple fault situations under synthetic traffic patterns, as FT-Z-OE uses minimal paths in the presence of fault(s) and saves VCs for performance improvement rather than deadlock avoidance.

For future work, to support transient fault tolerance, error control coding (ECC) techniques can be added. Also node failures can be modeled as its six output links are faulty. Besides, an appropriate 2D fault tolerant routing scheme can be utilized in FT-Z-OE to support horizontal faulty links too.

6. REFERENCES

[1] Dally, W. J. and Towles, B., 2001, Route packets, not wires: on-chip interconnection networks, In Proceeding of the Design Automation Conference, pp.684-689.

[2] Benini, L. and De Micheli, L., 2002, Networks on chips: a new SoC paradigm, Computer 35 (1), pp.70-78.

[3] A.W.Topol, et al., Three-dimensional integrated circuits, IBM J. RES. & DEV 50(4.5) (2006) 491-506.

[4] Loi, I., et al., 2008, A Low-overhead Fault Tolerance Scheme for TSV-based 3D Network on Chip Links, In Proceeding of ICCAD, pp. 598-602.

[5] Dumitras, T. and Marculescu, R., 2003, On-chip stochastic communication, In Proceeding of DATE, pp. 790-795.

[6] Pirretti, M., Link, G. M., Brooks, R. R., Vijaykrishnan, N., Kandemir, M., Irwin, M.J., 2004, Fault tolerant algorithms for network-on-chip interconnect, In Proceedings of IEEE Computer society Annual Symposium on VLSI, pp. 46 – 51.

[7] Zhu, H., Pande, P. P. and Grecu, C., 2007, Performance evaluation of adaptive routing algorithms for achieving fault tolerance in NOC fabrics, In Proceeding of ASAP, pp.42-47.

[8] Young Bok, K. and Yong-Bini, K., 2007, Fault Tolerant Source Routing for Network-on-chip, In Proceeding of 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI systems, pp.12-20.

[9] Schonwald, T., et al., 2007, Fully adaptive fault tolerant routing algorithm for network on chip architecture, In Proceeding of DSD, pp. 527-534.

[10] Glass, M. J., Ni, L. M., 1994, The turn model for Adaptive Routing, Journal of ACM 41(5), pp. 874-902.

[11] Chiu, G., 2000, The odd-even turn model for Adaptive Routing, IEEE Transactions on Parallel and Distributed Systems 11 (7) , pp.729-738.

[12] Pasrecha, S., et al., 2010, OE+IOE: A Novel Turn Model Based Fault Tolerant Routing Scheme for Networks-on-Chip, In Proceeding of ISSS, pp. 24-29.

[13] Andres, M., et al., 2009, Region based: a mechanism to support efficient routing algorithms in NOCs, In Proceeding of IEEE TVLSI, pp. 356-369.

[14] Hu, J. and Marculescu, R., 2004, Dyad- smart routing for networks on chip, in: Proceeding of DAC, pp. 260-263.

[15] Jovanovic, S., Tanougast, C., Weber, S., Bobda, C., 2009, A new deadlock-free fault-tolerant routing algorithm for NoC interconnections, In Proceeding of FPLA, pp. 326 – 331.

[16] Rezazadeh, A., Fathy, M., Hassanzadeh, A., 2009, If-cube3: An improved fault-tolerant routing algorithm to achieve less latency in NoCs, In Proceeding of IACC, pp. 278-283.

[17] Wu, J., 2003, A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model,” IEEE Transactions on Computers 52 (9), pp. 1154-1169.

[18] Boppana , R. V. and Chalasani, S., 1995, Fault-tolerant wormhole routing algorithms for mesh networks, IEEE Transactions on Computers 44 (7), pp. 848–864.

[19] Chien, A. A. and Kim, J. H., 1992, Planar-adaptive routing: low-cost adaptive networks for multiprocessors, In Proceeding of the 19th Annual International symposium on computer architecture, pp. 268-277.

[20] Nordbotten, N.A., et al., 2004, A fully adaptive fault-tolerant routing methodology based on intermediate nodes, In Proceeding of IFIP International Conference on Network and Parallel Computing, vol. 3222, pp. 341-356.

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

[22] Ebrahimi, M., Daneshtalab, M., Plosila, J., 2013, Fault-tolerant routing algorithm for 3D NoC using hamiltonian path strategy, Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.1601-1604.

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

[24] Li, M., Zeng, Q. A. and Jone, W. B., 2006, DyXY - A proximity congestion aware deadlock-free dynamic routing method for Network on Chip, In Proceeding of Design Automation Conference, pp. 849-852.

[25] https://nocs.stanford.edu/

[26] Kahng, A. B., Li, B., Peh. L.S., and Samadi, K., 2009, ORION2: A Fast and Accurate NOC Power and Area Model for Early-Stage Design Space Exploration, In Proceeding of DATE, pp. 423-428.

[27] Dally, W.J. and Towels, B., 2004, Principles and Practices of Interconnection Networks, Morgan Kaufmann.