Improving Reliability in Application-Specific 3D Network-on-Chip

Farnoosh Hosseinzadeh, Nader Bagherzadeh, Ahmad Khademzadeh, Majid Janidarmian, Fathollah Karimi Koupaei

Abstract— **Three-dimensional integrated circuits (3D ICs) offer an attractive solution for overcoming the barriers to interconnect scaling, thereby offering an opportunity to continue performance improvements using CMOS technology, with smaller form factor, higher integration density, and the support for the realization of mixed-technology chips.**

As feature sizes shrink, faults occur in on-chip network become a critical problem. At the same time, many applications require guarantees on both message arrival probability and response time. We address the problem of router failures by means of designing fault-tolerant architecture. The proposed architecture not only is able to recover from routers failure, but also improves the average response time of the system. In this design, in order to avoid adding a port in a router, a new component is also developed to reduce hardware overhead.

Index Terms—3DNetwork on Chip, fault-tolerance, By-pass, spare router, application-specific

I. Introduction

For implementing emerging complex applications, integrated system designs should be able to communicate tens to hundreds of cores [1]. An efficient data transfers among these cores can be achieved through innovations of the on-chip communications strategies [2]. Network-on-Chip (NoC) has been proposed as a promising solution to structure the design of the on-chip communications in multi core SoCs [3]-[5].

As the end of scaling the CMOS transistor comes in sight the third dimension may come to the rescue of the industry to allow for a continuing exponential growth of integration during the next decade [6]. As such 3D stacking may the key technology to sustain growth until more exotic technologies such as nanowire, quantum dot devices and molecular computers become sufficiently mature for deployment in main stream application areas.

Furthermore, the emerging three dimensional (3D) integration and process technologies allow the design of multi-level Integrated Circuits (ICs) [7]. This new design exhibits two major advantages, namely, higher performance and smaller [8].

However, a major challenge facing the design of such highly integrated 3D-ICs in deep submicron technologies is the increased likelihood of failure due to permanent and intermittent faults caused by a variety of factors that are becoming more and more prevalent. Permanent faults occur due to manufacturing defects, or after irreversible wear out damage due to electro migration in conductors, negative bias temperature instability, dielectric breakdown, etc [9], [10]. Intermittent faults on the other hand, occur frequently and irregularly for several cycles, and then disappear for a period of time [11], [12]. These faults commonly arise due to process variations combined with variation in the operating conditions, such as voltage and temperature fluctuations. This argument strengthens the notion that chips need to be designed with some level of built-in fault tolerance. Furthermore, relaxing of 100% correctness in the operation of various components and channels profoundly reduces the manufacturing cost as well as cost incurred by test and verification [13].

The remainder of this paper is organized as follows: In Section II, an overview of some fault-tolerant research efforts in NoC is given. Section III illustrates the basic concepts of application-specific NoC design, and a new fault-tolerant architecture is introduced in section IV. The results are given in section V followed by the concluding remark in Section VI.

II. Related work

Interconnect scaling has become one of the most crucial challenges in chip design, and is only expected to get worse in the future. 3D integration and NoC design methodologies are expected to overcome many of these challenges [14].
Pavlidis and Friedman [15] have compared 2D mesh NoC with its 3D counterpart by analyzing the zero-load latency and power consumption of each network. In the work of [16], a performance analysis method based on network calculus has been proposed for 3DNoC. In [17], the performance of several alternative vertical interconnection topologies has been studied. In [14], the authors proposed a dimension decomposition scheme to optimize the cost of 3DNoC switches, and presented some area and frequency figures derived from a physical implementation.

Fault-tolerant design is a design that enables a system to continue operation, possibly at a reduced level, rather than failing completely, when some part of the system fails.
Fault-tolerant routing algorithms should be able to find a path from source to destination in presence of the faults in NoC with a certain degree of tolerance [18]. Many algorithms in this area have been proposed which follow their own optimization aims.

Farnoosh Hosseinzadeh is with Computer Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, Iran, (phone: 98-9127246812; e-mail: fh.zadeh@srbiau.ac.ir).
Nader Bagherzadeh is with UCI, Irvine, (e-mail: nader@uci.edu).
Ahmad Khademzadeh is with Iran Telecommunication Research Center, Tehran, Iran, (e-mail: zadeh@itrc.ac.ir).
Majid Janidarmian is with Computer Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, Iran, (e-mail: jani@ieee.org).
Fathollah Karimi Koupaei is with Computer Engineering Department, Arak Branch, Islamic Azad University, Arak, Iran, (e-mail: karimi.fathollah@gmail.com).

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In [19], presents the scalable and fault-tolerant distributed routing (SFDR) mechanism. It supports three routing modes including corner-chains routing, boundary-chains routing and fault-ring routing.

A fault-tolerant mesh-based NoC architecture with the ability of recovering from single permanent failure is presented in [20]. This method adds a redundant link between each core and one of its neighboring routers, resulting in significant improvement in reliability while has little impact on performance. In [21], a hardware and performance aware design for the fault-tolerant 2DNoC architecture is presented.

In the 3D domain, [22] present an application specific 3DNoC synthesis algorithm that is based on a ripup-and-reroute procedure for routing flows, where the traffic flows are ordered in the order of increasing rate requirements so that smaller flows are routed first, followed by a router merging procedure. Murali *et al.* [23] propose a 3D NoC topology synthesis algorithm, which is an extension to their previous 2D work [24].

In this paper, a new fault-tolerant 3D application-specific network-on-chip was proposed which is able to tolerate routers failure and guarantees the 100% packet delivery.

III. Perquisites of Application-Specific NoC Design

A. Mapping Problem

To formulate mapping problem in a more formal way, we need to first introduce the following two concepts borrowed from [25]:

Definition 1: The core graph is a directional graph $G(V, E)$, whose each vertex $v_i \in V$ shows a core, and a directional edge $e_{i,j} \in E$ illustrates connection between v_i and v_j. The weight of $e_{i,j}$ that is shown as $comm_{i,j}$, represents the communication volume from v_i to v_j. The IP core along with a router connected to it by Network Interface (NI) is displayed as a tile.

Definition 2: The NoC architecture graph is a directional graph $A(T, L)$, whose each vertex $t_i \in T$ represents a tile in the NoC architecture, and its directional edge that is shown by $l_{i,j} \in L$ shows a physical link from t_i to t_j.

The core graph mapping $G(V, E)$ on NoC architecture graph $A(T, L)$ is defined by a one to one mapping function.

$$map: V \to T, s.t. map(v_i) = t_j, \forall v_i \in V, \exists t_j \in T, |V| \leq |T|$$

Mapping algorithms are mostly focused on mesh topology (Architecture Graph) which is the most popular topology in NoC design due to its suitability for on-chip implementation and low cost. The definitions are presented in Fig.1 and Fig.2.

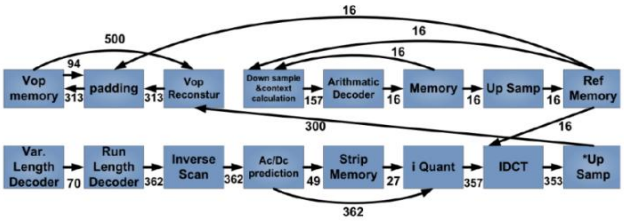

Fig. 1. The VOPD Core graph

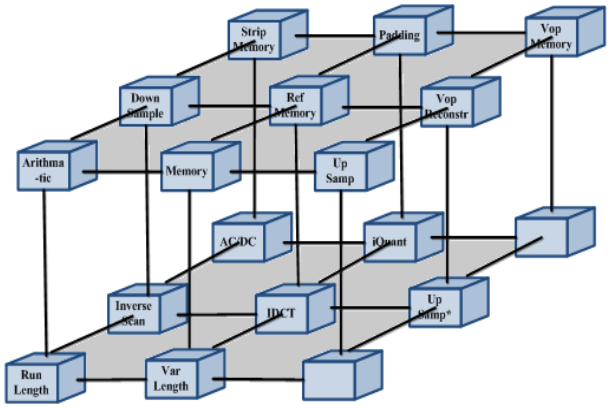

Fig. 2. Mapping of VOPD Core graph on 3DMesh

B. Routing Algorithm

The routing algorithm determines the path that each packet follows between a source-destination pair. Routing algorithms noticeably affect the cost and performance of NoC parameters i.e. area, power consumption and average message latency. Due to determined sources and destinations in application-specific NoC minimum-distance routing algorithms are mostly considered in this area which is computed off-line and admissible paths stored into the routing tables. For avoiding any possible deadlocks, we used the concept of application specific channel dependency graph (ASCDG) introduced in [26].

The proposed methodology is topology and application agnostic, the state-of-art mapping algorithm proposed in [27] is used to map Video Object Plan Decoder (VOPD) cores onto mesh topology. We have used minimum-distance routing algorithm in 3Dmesh i.e. XYZ. In XYZ routing algorithm, a packet is routed first along the X axis, then the Y axis and finally the Z axis.

IV. The Proposed Fault-Tolerant Design

In the mesh-based architecture which is the simplest and most dominant topology for today's regular 3D network on chips, each core is connected to a single router.

The natural and simplest extension to the baseline NoC router to facilitate a 3D layout is simply adding two additional physical ports to each router; one for Up and one for Down, along with the associated buffers, arbiters (VC arbiters and Switch Arbiters), and crossbar extension. We can extend a traditional NoC fabric to the third dimension by simply adding such routers at each layer.

If a failure occurs in a router in this topology, the failed router cannot be used any more for routing packets and the directly connected core obviously loses its communication with the network, so the expected requirements of the mapped application are not satisfied and the whole system breaks down.

In order to recover the inaccessibility of the core, we provide two steps: 1) Applying the spare router and new link interface. 2) Bypassing ports in each router to increase reliability and performance.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1. *Selecting Spare Router and Using Link Interface*

By assuming a faulty router in a mesh-based NoC, it is obvious that the core directly connected to the faulty router is inaccessible. Hence, each core is connected to two routers. They are main and spare routers which have been proposed in [20]. A router can inform the neighbors about its faulty status by setting a fault-status flag. This flag is checked by the neighboring routers and cores before starting any communication with the router.

Adding one port to all none-edge routers is not desirable, so to minimize hardware overhead, we used a component called Link Interface instead of router port. This module is suggested in order to reduce the overhead. After entrance of header flit into this module, destination address is decoded. As an example, if the address shows that connected core is the destination, header and its following flits will be sent to backup network interface of the core, otherwise they are routed to another output towards next router.

The Link Interface has been implemented with three processes which run concurrently; therefore it is able to transmit three dataflow as shown in Fig.3. This module has been also designed without using clock pulse that leads us to achieve better response time and power consumption.

Furthermore, we modify the algorithm in [28] not only for obtaining the best result in average response time, extra communication cost and system reliability, but also for having more routing path opportunities. This algorithm is explained in details in Fig.4.

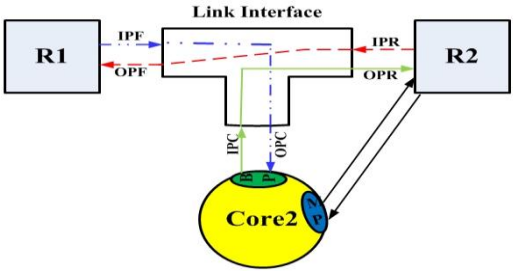

Fig. 3. The module Link Interface

```
Initialize (G (E, V));
For (All Routers)
        Router_Unused[i] =1;
Do
{
        Selected_Core =Find_Max_Comm (G (E, V));
       Available_Places =Find_All_Available_Places (Selected_Core);
       K=1;
       For (All Neighbor Routers)
            If (Router) unused[i] =1)
                {
                   Attach (Router[i], Selected_Core (Backup_Port));
                   Response_Time[K] = Calculate_Response_Time(Architecture);
                   Detach (Router[i], Selected_Core(Backup_Port));
                   K=K+1;
                }
 Selected_Router =Find_Best_Neighbor_router(Response_Time[]);
 If (Selected_Router_Unused_Port=True)          //Edge Routers
  {
    Attach (Router[i], Selected_Core (Backup_Port));
     Router_Unused (Selected_Router)=0;
  }
 Else     //Non-Edge Router
          Attach (Link_Interface, Selected _Core (Backup_Port));
  Update_Available_Places;
} While (Find Spare for All Cores);
```

Fig. 4. The pseudo code of selecting the best spare router

The proposed architecture that has been suggested in this step is shown in Fig.5. Each core is connected to its router via main local port and to the links using Link Interface via backup local port.

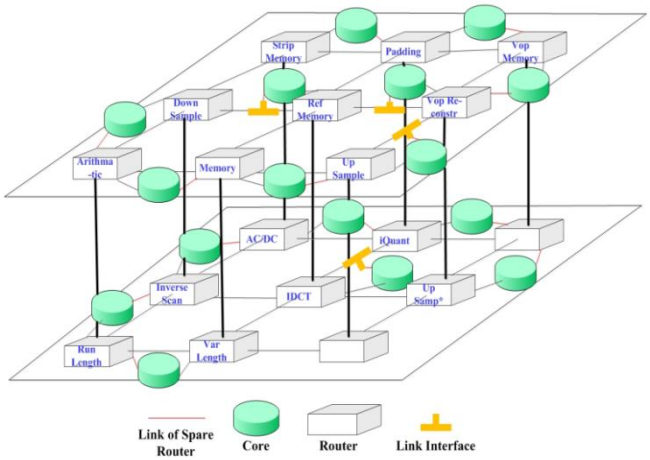

Fig. 5. The module Link Interface

2. *Bypassing Ports in a Router based on average response time of the system*

To enable a competent fault-tolerant NoC design, we develop proposed architecture, with bypassing between ports in the failed router, which remarkably decreases delay and communication cost.

There is constraint selecting the ports: each port is limited only one other port in the failed router. At the beginning, when a router fails, we indicate the connections do not reach to destination. Then select the two ports based on the lowest average response time. This algorithm is discussed in details in Fig.6.

Table I. indicates the best ports which are preferred for bypassing in each router N: North Port, E: East Port, W: West Port, S: South Port, U: UP Port and D: Down Port.

```
For (All Routers)
    Router [i] = Assume Fault;
   For (All Connections)
{
   If (Connection does not receive to destination)
            Connection [K] = 1;
}
    Do \\Connection==1;
   {
      Connection [j] ==0;
      Response Time[j] = Calculate Response_ Time (Connection[j])
           If (By-pass (two Ports) ==1) \\ before used
               Compare (Response Time (Connection (Before used), Connection[j]))
                        If (Response Time[j] is less)
                                Connection (Before used) =1;
           Else
                   By-pass (two Ports)=1;
     } While (All Connection Received);
```

Fig. 6. The pseudo code of bypassing ports

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
Selecting the best bypassing ports in each router

0 E-N L-W	**1** E-L	**2** N-E	**3** D-L
4 -	**5** E-N L-D	**6** U-L	**7** L-U S-E
8 N-L	**9** L-U	**10** U-L	**11** S-L
12 W-L	**13** W-D S-L	**14** -	**15** E-L

V. Experimental Results

Fault tolerance evaluates how reliability architecture can rout messages in despite different types of faults. Video Object Plan Decoder (VOPD) as a case study has been mapped on a 3×3×2 mesh topology using the best mapping technique proposed in [25]. To evaluate our design, we perform four most important fault-tolerant metrics: reliability, yield, response time and hardware cost.

A. Reliability Analysis

The reliability of a NoC router $R_r(t)$ is the probability that a router performs its functionalities correctly from time 0 to time *t*. It is decided by the failure rate, λ(t), which is measured by the number of failures per time unit. After the router has passed the infant mortality period, we can express $R_r(t)$ using the following equation [29]:

$$R_r(t) = e^{-\lambda t} \quad (1)$$

To calculate the reliability of the system, simple serial and parallel rules are used; hence for each pair of source and destination nodes in core graph, the reliability of the path is production of reliability of the routers which are met through the unique path according to XYZ algorithm. Then the system reliability is calculated again, each time assuming each individual router fails and accordingly a new rerouting path is contemplating XYZ algorithm and the additional spare routers. The final analytical formulation for system reliability is as (2):

$$R_{NOC} = \prod_{(i,j,k)\epsilon CTG} [\prod_{t=1}^{s} R_t + \sum_{t=1}^{s} (1 - R_t).(R_{i,j,k)|Router_t\ Fails})] \quad (2)$$

In which, (i,j,k) represent each pair of source and destination nodes in core graph. "S" shows the number of routers in each routing path and $R_{(i,j,k)}$ is the reliability of the path from i to j and then to k. We assume the failure rate of a router is λ=0.00315(times/year) [30].

Furthermore, for showing effect of mapping and number of TSVs that is used in sending packets, we execute different mapping algorithms in VOPD application such as Onyx [25], using least TSVs (LTSV), and using most TSVs (MTSV). The reliability of 3Dmesh compared to our design (3DP) over 1 to 50 years is shown in Fig.7.

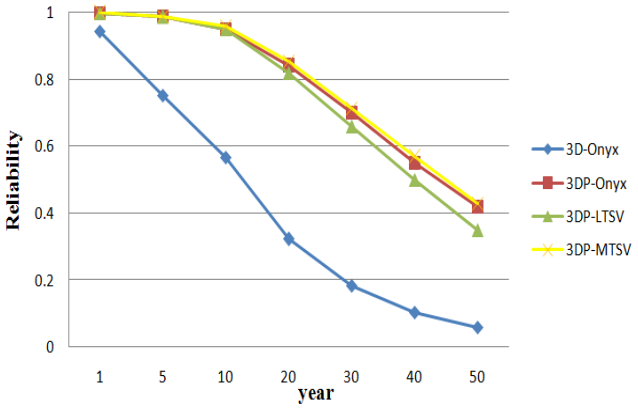

Fig. 7. Reliability of 3Dmesh and our design

Moreover, the reliability of the links is affected in reliability of system as can be seen in Eq.3. The failure rate of a horizontal link is λ_{LH}=0.0088(times/year) [30]. Because failure probability of vertical links is more than horizontal links [14], we assume the failure rate of a vertical link is λ_{LV}=N.λ_{LH} (N=1, 3, 6, 12).

$$R_{NOC}=\prod_{(i,j,k)\epsilon CTG} [\prod_{t=1}^{s} R_t.R_{Link\ |LinkTSV} + \sum_{t=1}^{s}(1-R_t).(R_{(i,j,k).}R_{Link\ |LinkTSV}\ |Router_t Fails] \quad (3)$$

The reliability of 3Dmesh compared to our design when add reliability of links is displayed in Fig.8, also the probability of faulty free connections when one to three routers fail are displayed in Table II.

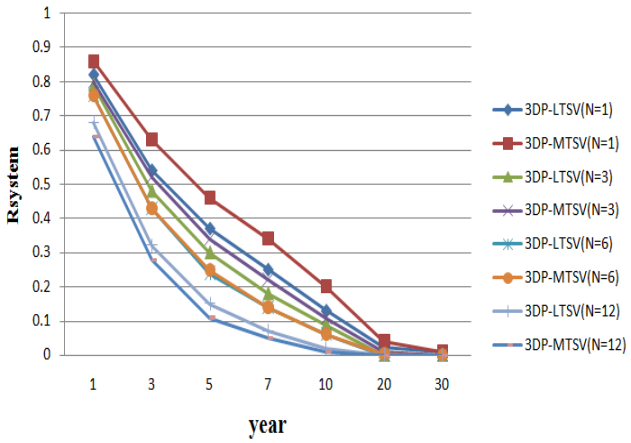

Fig. 8. Reliability of our proposed design in two different mappings

TABLE II
Probability Faulty-Free Connections

Number of Faulty Routers	2DNoC	3DNoC
1	100%	100%
2	87%	92%
3	68%	78%

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. *Yield Analysis*

The yield of NoC is modeled by the yield of block and the yield of connection. Recent research has shown that correlation factor (Ω) between components should also be introduced in the yield calculation [31]. According to [31], our system's yield is modeled by equation (4).

$$y_{NoC} = (y_{Block})^n . y_{conn} \quad (4)$$

We assume a channel has 64-bit data wires, 4-bit control and 8-bit parity wires. The parameters used in our yield evaluation work are listed in Table III. [30].

TABLE III
Parameters for yield calculations

α	Clustering parameter	2	y_i	True yield of other components	99.5%
Ω_i	Defect coverage for all components	99%	y_{w-ctrl}	Yield of control or parity wires	99.99%
y_c	True core yield	97%	σ_{i-j}	Correlation among two components	0.5

We perform the Monte Carlo simulation for the NoC yield. Fig.9 shows the flow chart of our simulation. The experimental results are shown in Fig.10.

The yield of 3DNoC is very low [14], but the yield of system has improved 17% with applying our fault-tolerant architecture.

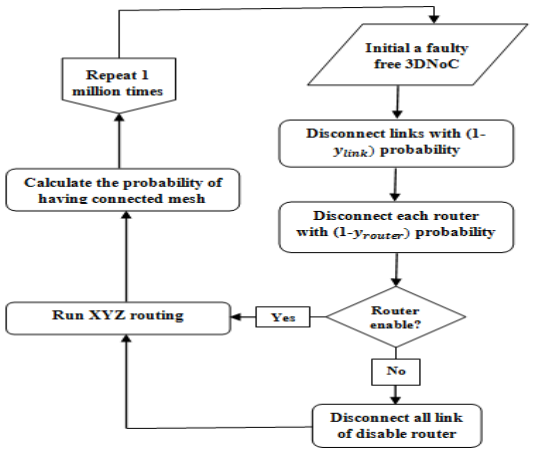

Fig. 9. Monte Carlo simulation

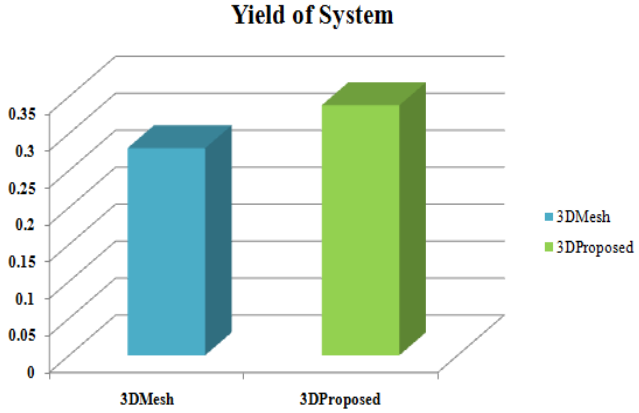

Fig. 10. The yield analysis

C. *Average Response time Analysis*

In order to compare the average response time of the reliable architecture and 3Dmesh, they have been designed in VHDL and synthesized using Xilinx ISE.

The proposed architecture significantly decrease the average response time on the faultless and 18 possible faulty routers compared with the 3Dmesh architecture as illustrated in Fig.11.

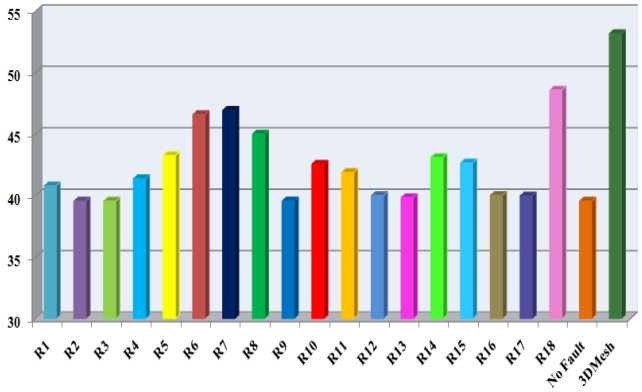

Fig. 11. Average response time of system in different situations

It should be pointed that it actually is a great achievement to develop a fault-tolerant NoC design which also has better performance. To explain in details, when all routers are correctly operating, new architecture improves the average response time by 52% comparing to mesh.

In Fig.12 we observe that the proposed approach allows decreasing the response time of system by 34% and tolerating permanent failure of each single router.

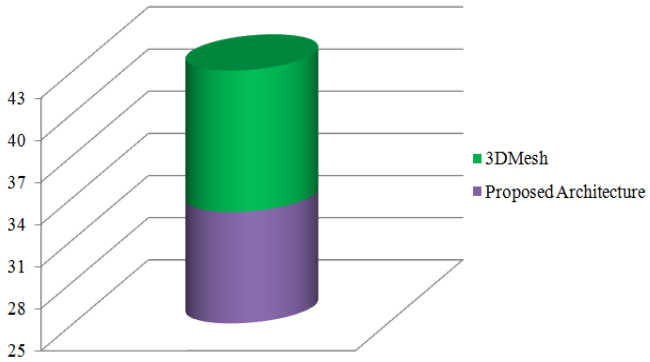

Fig. 12. Comparing proposed architecture with 3DMesh in terms of average response time

D. *Hardware Cost*

To recover from a permanent fault, hardware redundancy is mandatory and reduction of this overhead has always been an important issue in this area. In our design, we do not add any router port and instead a link interface was developed which helps to achieve less hardware overhead.

The proposed architecture designed and implemented in the Vertex E FPGA (v50ecs144-6). Our proposed fault-tolerant architecture added 15% redundancy camper with 3DMesh in the worst case as shown in Fig.13.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

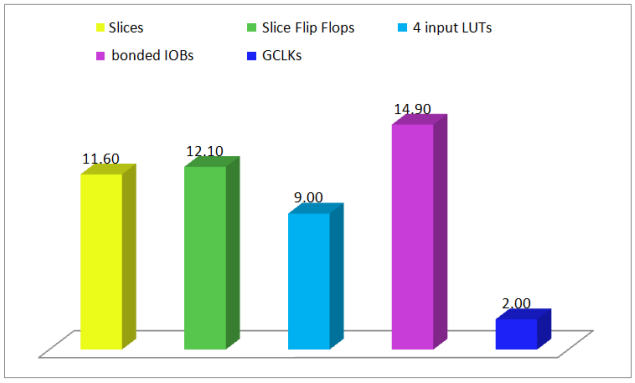

Fig. 13. Comparing hardware overhead of proposed design with 3DMesh (all utilities compare in percent)

VI. Conclusion

To achieve the targeted reliability, the errors should be recovered. Permanent error recovery results in huge area and energy overhead. In this paper, a fault- tolerant application-specific architecture has been proposed to improve latency and yield with up to 15% area overhead in 3DNoC.
This architecture is topology, application, mapping agnostic, but using a mapping with least hop counts is very important, because it leads to improving reliability, and yield. Furthermore, this fault-tolerant approach designed in VHDL and synthesized using Xilinx ISE. Simulation results show that the proposed method in 3DNOC reduces latency by 52% and achieves an up to 17% higher yield, compared to mesh. In 3DMesh, we must apply a mapping algorithm with using at least TSV and Extra Bandwidth.

Reference

[1] Semiconductor Industry Association, "International technology roadmap for semiconductors," 2006.
[2] Fayez Gebali, Haytham Elmiligi, Mohamed Watheq El-Kharashi, "Networks-on-chips : theory and practice," 2009 by Taylor & Francis Group, LLC
[3] A. Hemani et al. Network on chip, " An architecture for billion transistor," In 18th IEEE NorChip Conference, November 2000.
[4] W. J. Dally, B. Towles, "On-chip interconnection networks," In Proceedings of the Design Automation Conference, p.p 684–689, June 2001.
[5] L. Benini, G. De Micheli. "Networks on chip: A new SoC paradigm," IEEE Computer, 49(2/3):70–71, January 2002.
[6] A.sheibanyard, F.Perrot, A.jantsch, " 3D Integration for NoC-based SoC Architecture", 2011,X.278 p. 100 illus.
[7] C. Constantinescu, "Trends and challenges in VLSI circuit reliability," IEEE Micro, 23(4):14-19,july-aug 2003.
[8] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost considerations in network on chip, " Integr. VLSI J. 38 (2004) (1): 19-42.
[9] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost considerations in network on chip," Integr. VLSI J. 38 (2004) (1): 19-42.
[10] S.Nassif, "Modeling and analysis of manufacturing variations", Proc. CICC, May 2001.
[11] S. Pan et al, "IVF, "Characterizing the vulnerability of microprocessor structures to intermittent faults," Proc.DATE 2010.
[12] C.Constantinescu, "Intermittent faults in VLSI circuits," Proc. IEEE Workshop on silicon Errors in logic (SELSE), 2007.
[13] Tudor Dumitraş, Radu Mărculescu, "On-Chip Stochastic Communication," date, vol. 1, pp.10790, Design, Automation and Test in Europe Conference and Exhibition (DATE'03), 2003.
[14] Igor Loi, Federico Angiolini, Shinobu Fujita, Luca Benini, "Characterization and Implementation of Fault-Tolerant Vertical Links for 3-DNetworks-on-Chip," IEEE Transactions On Computer-aided Design Of Integrated Circuits And Systams, VOL. 30, NO. 1, JANUARY 2011.
[15] V.F. Pavlidis and E.G. Friedman, "3-D Topologies for Networks-on-Chip," IEEE Trans. Very Large Scale Integration (VLSI '07), pp. 1081-1090, Oct. 2007.
[16] Brett Feero, Partha Pratim Pande, " Performance Evaluation for Three-Dimensional Networks-On-Chip", IEEE Computer Society Annual Symposium on VLSI(ISVLSI'07).
[17] Awet Yemane Weldezion, Matt Grange, Weerasekera, Hannu Tenhunen," Scalability of Network-on-Chip Communication Architecture for 3-D Meshes," 2009 IEEE
[18] T. Dumitras, S. Kerner , R. Marculescu, "Towards on-chip faulttolerant Communication," In Proc. Asia and South Pacific Design Automation Conference, 2003.
[19] Zewen Shi, Kaidi You, Yan Ying, Bei Huang, Xiaoyang Zeng, Zhiyi Yu, " A Scalable and Fault-Tolerant Routing Algorithm for NoCs,", International Symposium on Circuits, pp.165-168, 2010.
[20] F. Refan, H. Alemzadeh, S. Safari, P. Prinetto, Z. Navabi, "Reliability in Application Specific Mesh-Based NoC Architectures, " On-Line Testing Symposium, 2008. 14th IEEE International, pp.207 – 212 ,Jul. 2008.
[21] Fathollah Karimi Koupaei, Ahmad Khademzadeh, Majid Janidarmian, "Fault-Tolerant Application-Specific Network-on-Chip", IEEE 2011.
[22] S. Yan and B. Lin. Design of application-specific 3D networks-on-chip architectures. In *Proc. ICCD*, pages 142–149, 2008.
[23] S. Murali, C. Seiculescu, L. Benini, G. D. Micheli,"Synthesis of networks on chips for 3D systems on chips," In *Proc. ASPDAC*, p.p 242–247, 2009.
[24] S. Murali et al. "Designing application-specific networks on chips with floorplan information," In *Proc. ICCAD*, pages 355–362, 2006.
[25] M. Janidarmian, A. Khademzadeh, M. Tavanpour, "Onyx: A new heuristic bandwidth-constrained mapping of cores onto tile-based Network on Chip", IEICE Electron. Express, Vol. 6, No. 1, pp.1-7,January 2009.
[26] M. Palesi, R.Holsmark, S.Kumar, ―a methodology for design of application specific deadlock-free routing algorithms for NoC systems, Hardware/Software Codesign and System Synthesis, CODES+ISSS '06. Proceedings of the 4th International Conference, pp. 142-147, Oct. 2006.
[27] M. Janidarmian, A. Roshan Fekr, V. Samadi Bokharaei, "Application-Specific Networks-on-Chips Design", IAENG International Journal of Computer Science, 38:1, pp16-25, 2011.
[28] Majid Janidarmian, Melika Tinati, Ahmad Khademzadeh, Maryam Ghavibazou, Atena Roshan Fekr, "Special Issue on a Fault-tolerant Network on Chip Architecture" AIP Conf. Proc., Volume 1247, pp.191-204, 2010.
[29] L.T Wang, C.E Stroud, N.A Touba, "System-on-Chip Test Architectures," Nanometer Design for testability,Mogran Kauffmann,2008.
[30] W.J Dally and B. Towles, "Principles and Practice of Interconnection Networks," Morgan Kaufmann, 2003.
[31] Saeed Shamshiri, Kwang-Ting (Tim) Cheng, "Modeling Yield, Cost, and Quality of a Spare-enhanced Multi-core Chip,"IEEE TRANSACTIONS ON COMPUTERS2011.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)