# Design and Implementation of Fault Tolerant Router for Network On Chip using FPGA

**Krishna J Parmar, Prof. Hardik H. Makwana**
[1]PG Student,[2]Assistant Professor
Dept. of Electronics and Communication
Hasmukh goswami college of engineering,GTU

**Abstract— *T*he growing complexity of Multiprocessor Systems on Chips (MPSoCs) is requiring communication resources that can only be provided by a highly-scalable communication infrastructure. The effectiveness of this approach largely depends on the availability of a design methodology.With technology scaling, as the geometries of the transistors reach the physical limits of operation, another important design challenge of SoCs will be to provide dynamic (run-time) support against permanent and intermittent faults that can occur in the system. The interconnect will be susceptible to various noise sources such as cross-talk, coupling noise, process variations, etc.. Designing systems under such uncertain conditions becomes a challenge.Therefore, the goal is to solve some of the most important and time intensive problems encountered during NOC design can be solved with Fault tolerant Router.. While preserving the throughput, the network load, and the data packet latency of NOC router.**

***Index terms*- Fault Tolerant, XY Routing, VLSI Architecture, and FPGA**

## I. INTRODUCTION

With the mature and further development of deep submicron Very-Large-Scale-Integration (VLSI) circuit technology, chip design industry is facing a new serious problems: with the development of chip function and performance requirements, chip scale is becoming more and more large, working frequency is becoming higher. This results in longer development cycle, and design quality is more difficult to control, design cost is higher. These make system-on-chip (SoC) which is based on the bus structure as the main features are difficult to meet the requirements increasingly, mainly in the challenges of the system extensibility, difficulties in single clock synchronization, the design efficiency and the scissors

The emergence of NoC solved a series of questions which on-chip interconnect posed in the multi-core design of SoC effectively, and the architecture of NoC is aimed to improve SoC interconnect problems. Its main design goal is to support the structure of an interconnect communications between the kernels on-chip [4], [5]. This interconnected structure is composed of routing switching network and resources, adopted the design idea of global asynchronous locally synchronous (GALS) [6]. In short, NoC solves three big problems brought by SoC bus structure architecturally: the scalability resulted by the limited address space [7], [8]; the communication efficiency caused by sharing communication [9], [10]; the energy consumption and area problems due to the global synchronization [11], [12]. So it attracts more and more attentions of integrated circuit designers, and becomes a hot technology of computer industry and chip design community to study.

## II. ROUTER ARCHITECTURE

The basic building block of any NoC is its router which is responsible for guiding the data packets to the next router or an IP as per a routing strategy specified inside it . The assumption made in this design is that all the resources are homogeneous and deliver packets of same length. A basic router for mesh topology has five input and output ports as displayed in *Fig. 4.2* The internal structure of router consists of three main components, a FIFO Buffer at the input from each of the four directional ports i.e. north, south, east and west and one from the computational resource, an Arbiter and a Crossbar.

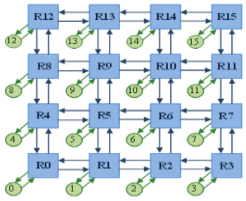


Fig. 1. 4×4Mesh Architecture

A better understanding of the functionality of a router can be obtained by looking at the path traversed by a data packet inside the router. Assume a data packet comes into the router through one of the input ports of a FIFO Buffer. Then the FIFO

Buffer sends the destination address part of the packet to the Arbiter. The arbiter performs the arbitration process and on request grant, sends the arbitration result to the Crossbar block. It also sends a grant signal to the FIFO. The FIFO then pops the data packet leading to its injection into the

input port of crossbar. The packet then traverses through the crossbar from its input port to corresponding output port based on the arbitration result. Finally, the packet leaves the router.

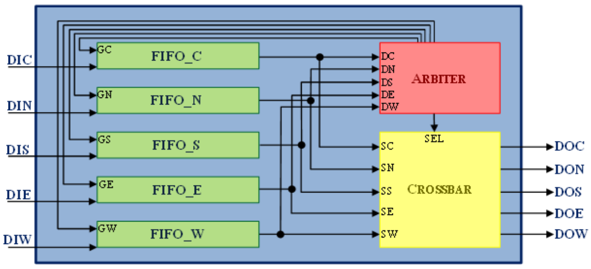


Fig 2. Router Block Diagram

### III Simulation Setup

This section describes how simulation is performed and what are the conditions that is considered in our evaluation.

We evaluated our algorithm on 2D mesh network. The network size during sim-ulation is fixed to 12 12 tiles. The input port of each switch has a FIFO size of 4 its and we used uniform traffic model to simulate the performance of the network. Under the uniform traffic model a PE sends a 4- bit packet to other PE with equal probability. Simulation time is 6000 cycles and 1000 cycles as warm-up time. The warm-up time means that the statistic results are considered after 1000 simulation cycles in order to have more realistic evaluation. In non-faulty network each node can generates 85 packets in 6000 simulation cycles without any packet loss.

We have simulated seven faulty conditions for the 12 12 network by injecting 1, 3, 5, 7, 10, 15 ,20 faults in the entire network. Simulation of each faulty condition has been repeated 100 times, in each of them the faults are located randomly in some of the network switches. To have a realistic distribution among the faults which corrupt only one switch port (single port deactivation) , and the faults which corrupt the entire switch operation (switch deactivation) the statistics provided by [11] is used. Accordingly, 60% of the faults lead to port deactivation and the rest 40% have switch deactivation effect. The result are averaged over 100 simulations for each faulty condition.

### Evaluation Metrics

The performance metric used in evaluation are drop ratio, average latency and throughput. which are defined as follows:Drop ratio: Drop ratio shows the rate of unsuccessful packet deliveries, which is defined by the following formula [11]:

Drop Ratio = (#injected Packet #received Packets)/(#injected Packets)*100%

In a network with live-lock and deadlock free routing algorithm, injected packets are delivered to the destination when there exist enough resources in the network. Upon packet injection by the sender processing elements, available network resources (i.e., buffer space for packet storage in the connected switch) are considered. When the resources are not enough for the packet injection, PE cannot inject the packet. Ac-cording to the application the packet may be dropped or wait until some resources get released. In our simulations, we evaluate the network metrics under certain loads. When due to the non-enough resource packet injection must be postponed, it is considered as a drop. Therefore, drop ratio parameter implicitly indicates to the cases where resources are not enough to achieve certain load in the network.

Packet Latency: Packet latency is defined for each packet as the time between packet injection by the sender and packet delivery at the destination, and it is com-monly measured in clock cycle (simulation cycle). The latency has direct relation with the network load (load$_{net}$). Latency of a packet is less when the network load is lower. The minimum time between packet injection and delivery can be counted by the switch delays and the number of switches in the routing path. However, due to network traffic, the packet has to wait in some switches to get required resources. In this thesis, packet latency is calculated by the following formula:

Latency= $T_{receivd}$ $T_{send}$

In which $T_{send}$ is the time that the header it of the packet leaves the source PE, and $T_{receive}$ is the time that the tail it of the packet reaches the destination PE.

Throughput: Throughput refers to how many packets can be transferred and received by destination switch in network in a given amount of time. It is used to measure the performance of NoCs. Normalized throughput is obtained by the following formula:

Throughput = (#received packet)/ Simulation cycle network size

Saturation Throughput: Saturation throughput is the maximum throughput that the network can produce, and the throughput cannot be more than saturation throughput [12].

Experiments

Experiment 1: Connectivity

The purpose of the connectivity test is to be assure that all processing elements, which are connected to non-faulty and non-deactivated switches, can communicate with each other. For this purpose, we have simulated 7 faulty conditions and each faulty condition has been repeated 100 times in different location. To check pair to pair connection, each processing element sends a packet to all other PEs, (e.g., in a 12 12 non-faulty network, each PE sends packets to the other 143 PEs). We have monitored drop ratio in each condition. The results show 0% drop ratio. It means that, all PEs, connected to the non-faulty and non-deactivated switches, can communicate with each other. In other words, there exists a routing path among each pair of PEs in the network using the routing algorithm presented in this thesis.

Experiment 2: $t_{min}$

In our evaluation each processing element (PE) is sending packet to any other PE with the same time interval. The minimum time interval required between two consecutive packet injections is called $t_{min}$ [11]. If PE injects the first packet at time $t = t_1$, and the second packet at time $t = t_2$, then $t_2$ $t_1$ $t_{min}$ otherwise the packet is dropped in PE. Minimum time interval varies based on the network size, buffer size, traffic model, packet size and other network switch parameters. In order to find $t_{min}$, we have reduced the time interval and monitor the drop ratio in a non-faulty network.

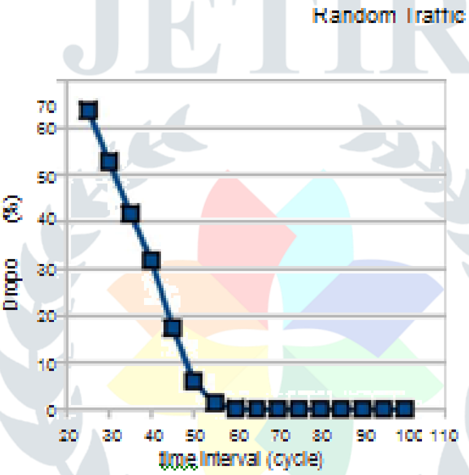


Figure 3: $t_{min}$ for uniform traffic model.

As it is shown in figure 3, for uniform traffic, when the time interval is less than 60 cycles, the packets are dropped in the non-faulty network. So the $t_{min}$ for uniform traffic model is 60 cycles.By having the minimum time interval, the load per PE is defined as:

$$load_{PE} = t_{min}/t_{avg}$$

Where $t_{avg}$ is the average time between two consecutive packet injection and is greater than $t_{min}$ $Load_{net}$ has a direct relation with $load_{PE}$ and the number of the PEs that are able to inject packet. The network load is defined as:

$$Load_{ne}t = \text{Active PEs/Network Size}$$

In a non-faulty network $load_{net}$ is equal with $load_{P E}$ since all PEs contribute in the network traffic. But in the faulty network $load_{net}$ is less than $load_{P E}$.

Experiment 3: Saturation Throughput

In order to achieve the saturation throughput we increase the $load_{P E}$ by decreasing the $t_{avg}$. As it is shown in figure 4, increasing the packet injection rate doesn't lead to further increase of network throughput. The red and yellow part of the figure show the situations where the $t_{avg}$ is less than $t_{min}$ and we have packet loss in these situations.

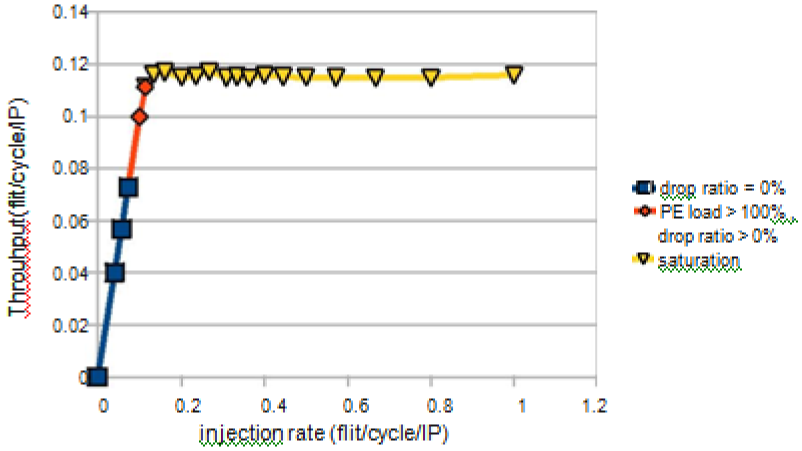


Fig. 4 Saturation throughput in non-faulty network.

Experiment 4: Performance Comparison

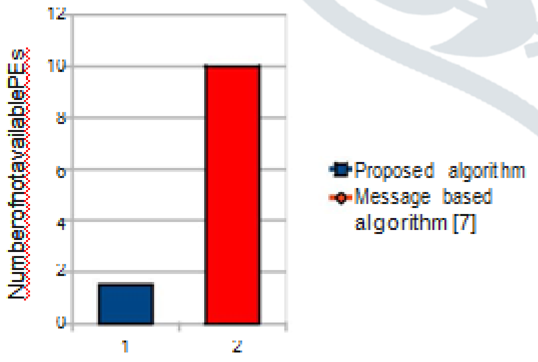
To demonstrate the effectiveness of our fault tolerant routing algorithm, we com-pare it to a state-of-the-art fault tolerant routing algorithm [7] which use the same network architecture and switching technique.

Variable Load

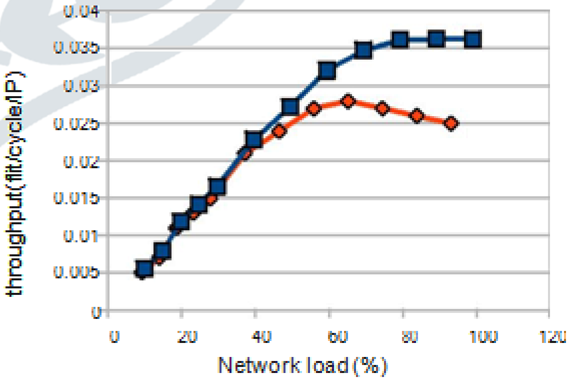
In the first experiment, both routing algorithms are computed in presence of 5 in-jected faults in the network. In our simulation the traffic model is uniform. Load$_{P E}$ is the same in two networks but load$_{net}$ depends on the available PEs. Figure 4.3

(a) shows that in reference algorithm ten switches are deactivated, but in proposed algorithm only two switches are deactivated.

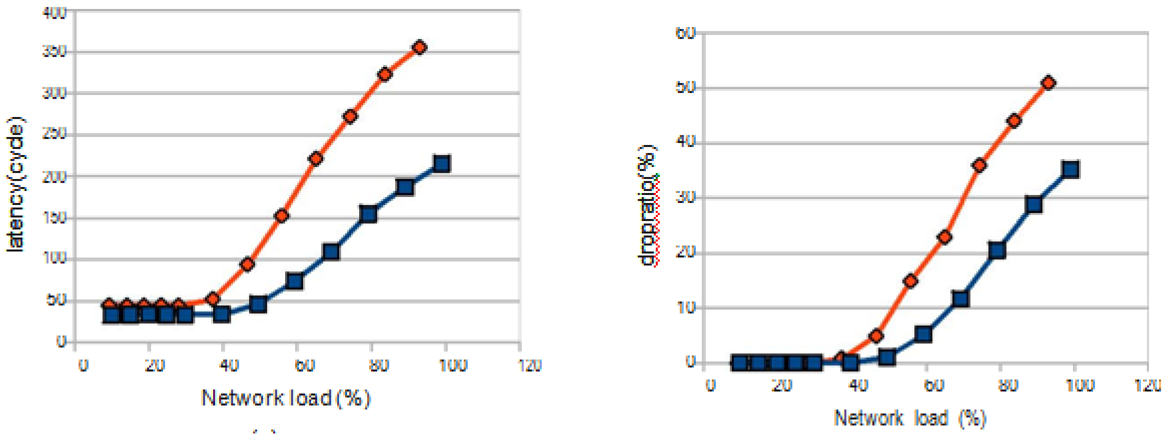
Figure 4.3 (b) shows that when the network load is under 40 percents the through-put of the two algorithms are the same. Drop ratio is zero percent and all generated packets reach their destinations. But when the network load is increased the pro-posed algorithm has higher throughput due to the less drop ratio. The reason is that in the reference algorithm more switches are deactivated so the available rout-ing paths (network resources) are less than the proposed algorithm. With the same reason, as it can be seen from the figure 4.3 (c) the proposed algorithm has less average latency than the reference algorithm. Figure 4.3 (d) shows that both algo-rithms have no drop ratio in low loads. And the reason is that the algorithms are fault tolerant and all generated packets reach their destinations, but in higher loads, since reference algorithm has less network resources, more packets are dropped.



(a)



(b)

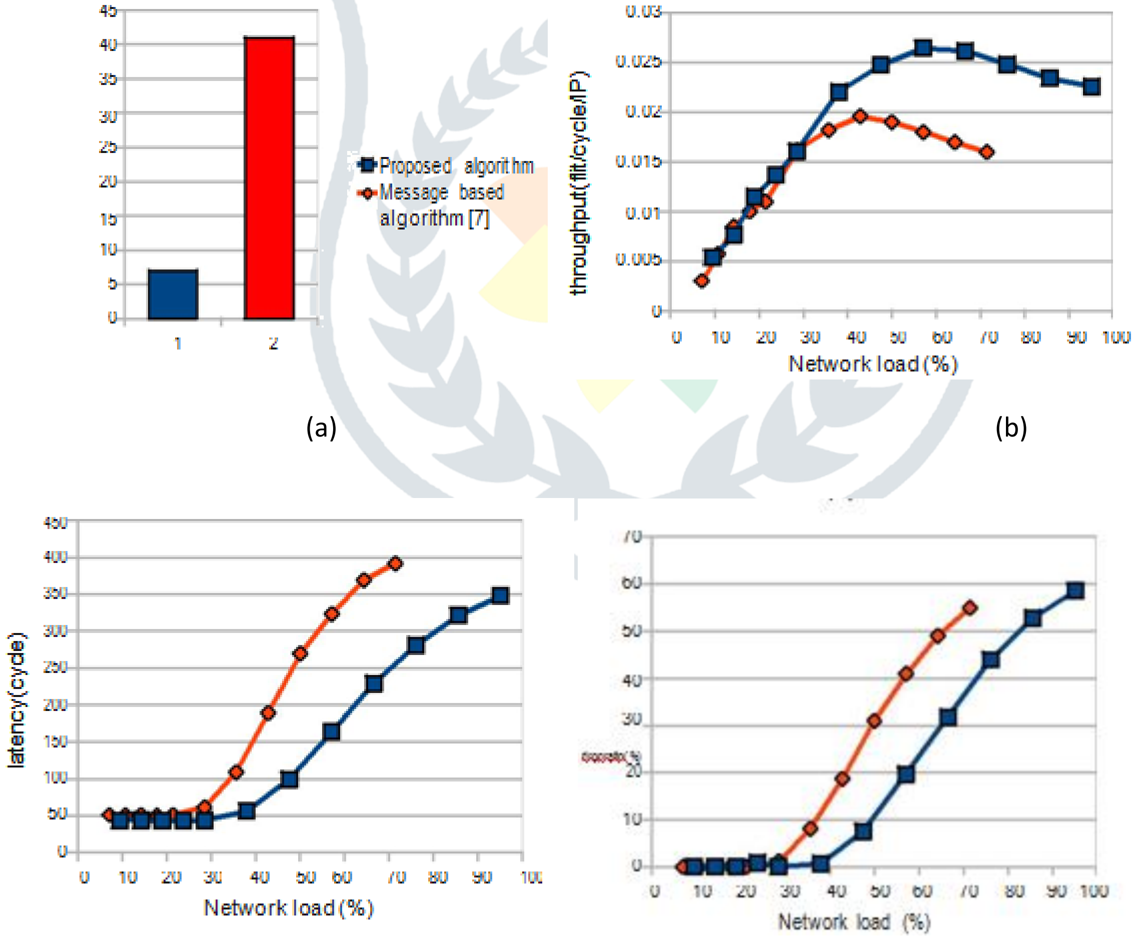


(c) (d)

Figure 5: Performance comparison of the algorithms under the uniform traf-c when the fault counts are 5, and the network size is 12 12 (a) Number of not available processing elements (b) Average through-put (c) Average packet latency (d) Average drop ratio.

To further compare the proposed algorithm with the reference algorithm, the number of faults are increased to 10 faults.

Average Result



(a) (b)

(c) (d)

Figure 6: Performance comparison of the algorithms under the uniform when the fault counts are 10, and the network size is 12 12 (a) Number of not available processing elements (b) Average throughput (c) Average packet latency (d) Average drop ratio.

Figure 6 (a) shows that in reference algorithm forty switches are deactivated, but in proposed algorithm only ten switches are deactivated. Figure 6 (b) indicates that, due to the number of the available PE the maximum possible load$_{net}$ improves from 70% to 97%, which results in higher throughput in our algorithm. Figure 6(c) shows that reference algorithm has higher latency. The main reason is that forty switches are deactivated in the reference algorithm, on the other hand only ten switches are deactivated in our algorithm. As a result there are less available routing paths in reference algorithm. As it is shown in figure 6 (d), the same as previous examine the higher drop ratio of the reference algorithm is due to less available routing path.

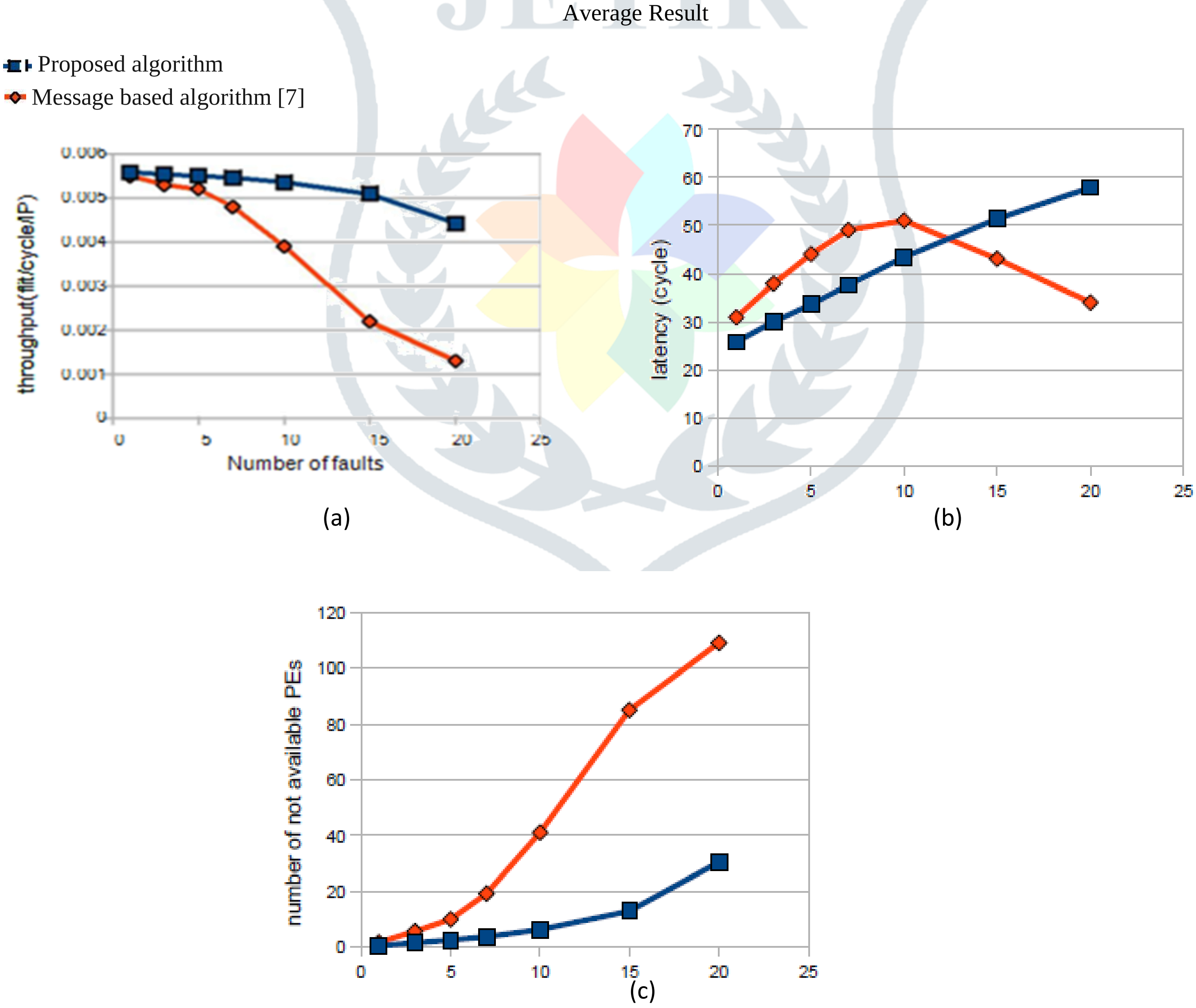
In the next section we compare the algorithms in terms of throughput and latency with a variable number of faults.

Variable Faults

In this experiment the uniform traffic has been applied, and the average latency, throughput and number of deactivated switches are compared in presence of various number of faults in 12 12 NoCs with a fixed load$_{P E}$ = 10%. Due to the low load there is no packet loss and the drop ratio is zero percent.

Figure 7 (a) shows that the network with the proposed algorithm is able to deliver more packets and the proposed algorithm has always higher throughput. The result shows that difference between the throughputs has direct relation to the number of the injected faults. As it is shown in figure 7 (b) when the number of faults are less than 10, the average latency of the proposed algorithm is less than the reference algorithm due to more possible routing paths. But by increasing the fault density, the result is the opposite, since the number of generated packets in the reference algorithm decreased and the network is less loaded.

Figure 7 (c) shows that the number of not available PEs in the proposed algorithm is higher than the reference algorithm especially with high fault densities. This is due to two factors. First, the PEs that are connected to the semi-faulty switch are often still reachable because only a port towards a neighboring switch is affected. Second, because of having smaller faulty region less switches are deactivated in proposed algorithm.



(a)

(b)

(c)

Figure7: Performance comparison of the algorithms under the uniform traffic when the processing element load is 10%, and the network size is 12 12 (a) Average throughput (b) Average packet latency (c) Number of not available processing elements.

## IV. ANALYSIS AND EXPERIMENTAL RESULTS

The synthesis reports of Fault Tolerant Route

## V. CONCLUSION

Finding out a routing algorithm which is able to deal with fully and partially de-fective switches and defective interconnects is the main objective of this thesis. The objective was achieved, and we found out the non-deterministic algorithm that fulfills our requirements. The proposed algorithm is an enhancement of the available region-based approach for NoCs [1]. The novelty of our approach is that link failures are modeled as semi-faulty switches and as a result the faulty region is smaller and less healthy switches are deactivated.Due to distinguishing faulty-links [11] we proposed a new idea (highway) that let the packets forwarded through the faulty-region which results in less traffic congestion. The performance shows the advantages of the proposed algorithm with state-of-the-art fault tolerant routing algorithms [7]. Since our algorithm has less deactivated switches it has always higher throughput and less latency. Router ihas been implemented on FPGA using HDL. To study the performance of this architecture, it is implementing in Xilinx.

## REFERENCES

[1] Boppana R.V., Chalasani S., "*Fault-tolerant wormhole routing algorithms for mesh networks*", IEEE Transactions on Computers, vol. 44, no. 7, pp. 848-864, 1995.

[2] Zhen Z., Greiner A., Taktak S., *"A recon gurable routing algorithm for a fault-tolerant 2D-mesh network-on-chip*", Design Automation Conference, pp. 441-446, 2008.

[3] Jovanovic S., Tanougast C., Weber S., Bobda C., "*A new deadlock-free fault-tolerant routing algorithm for NoC interconnections"*, International Conference on Field Programmable Logic and Applications, pp. 326-331, 2009.

[4] Fick D., DeOrio A., Chen G., Bertacco V., Sylvester D., Blaauw D., "*A highly resilient routing algorithm for fault-tolerant NoCs*", Europe Conference in Design, Automation and Test, pp. 21-26, 2009.

[5] Jie W., "*A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model*", IEEE Transactions on Computers, vol. 52, no. 9, pp. 1154-1169, 2003.

[6] Chen K., Chiu, G-M, "*Fault-tolerant routing algorithm for meshes without using virtual channels"*, Journal of Information Science and Engineering, vol. 14, no. 4, pp. 765-783, 1998.

[7] Holsmark S., Kumar S., "*Corrections to Chen and Chiu^as fault tolerant routing algorithm for mesh networks*", Journal of Information Science and Engineering, vol. 23, pp. 1649-1662, 2007.

[8] Fukushima Y., Fukushi M., Horiguchi S., "*Fault-tolerant routing algorithm for network on chip without virtual channels*", Defect and Fault Tolerance in VLSI Systems, pp. 313-321, 2009.

[9] Holsmark R., Palesi M., Kumar S., "*Deadlock free routing algorithms for ir-regular mesh topology NoC systems with rectangular regions"*, Journal of system architecture, vol. 54, Issue 3-4, pp. 427-440, 2008.

[10] Fukushima Y., Fukushi M., Yairi I. E., Hattori T., "*A hardware-oriented fault-tolerant routing algorithm for irregular 2D-mesh network-on-chip without virtual channels"*, Defect and Fault Tolerance in VLSI Systems, pp. 52-59, 2010.

[11] Dalirsani A., Holst S., Elm M., Wunderlich H., "*Structural Test for Graceful Degradation of NoC Switches"*, European Test Symposium, pp. 183-188, 2011.

[12] Neeb C., Wehn N., "*Issues on E cient Network-on-chip"*, Kaiserslautern Techn. Univ., 2008.

[13] Xu Y., Zhou J., Liu S., "*Research and analysis of routing algorithm for NoC"*, IEEE International Conference on Computer Research and Development (ICCRD), pp. 98-102 , 2011.

[14] Dally W.J, Charles L.S, "*Deadlock free message routing in multiprocessor in-terconnection networks"*, IEEE Transactions on Computers, pp. 547-553 , 1987.

[15] Rantala V., Lehtonen T., Plosila J., *"Network on chip routing algorithms*", TUCS Technical Report, 2006.

[16] Schonwald T., Zimmermann J., Bringmann O., Rosenstiel W., "*Fully adap-tive fault-tolerant routing algorithm for network-on-chip architectures*", IEEE Con-ference on Digital System Design Architectures Methods and Tools, pp. 527-534, 2007.

[17] Kohler A., Radetzki M., "*Fault-tolerant architecture and de ection routing for degradable NoC switches*", IEEE International Symposium on Networks-on-Chip, pp. 22-31, 2009.

[18] Momeni L., Rezazadeh A., Fathy M., "*A low latency routing algorithm for irregular mesh network-on-chip*", IEEE Modeling symposium on computer modelling and simulation, pp. 328-333, 2010.

[19] Kuo-Shun D., Ching-Tien H., Jyh-Jong T., *"Matrix transpose on meshes with wormhole and XY routing"*, IEEE Symposium on Parallel and Distributed Process-ing, pp. 656-663, 1994.

[20] Mehranzadeh A., Khademzadeh A., Mehran A., *"FADyAD- Fault and conges-tion aware routing algorithm based on DyAD algorithm*", International Symposium on Telecommunications (IST), pp. 274-279, 2010.

[21] Ge-Ming C., "*The odd-even turn model for adaptive routing*", IEEE Transac-tions on Parallel and Distributed Systems, vol. 11, no. 7, pp. 729-738, 2000.

WEB SITES

[1] http://www.hinduism.co.za/NoC.htm