Dual-NI Architectures for Fault Tolerant NoC

Ville Rantala[1,2], Teijo Lehtonen[1,2], Pasi Liljeberg[2], Juha Plosila[2,3]
[1]Turku Centre for Computer Science (TUCS), Joukahaisenkatu 3-5 B, FIN-20520 Turku, Finland
[2]University of Turku, Department of Information Technology
[3]Academy of Finland, Research Council for Natural Sciences and Engineering
Email: ville.rantala@utu.fi, teijo.lehtonen@utu.fi, pasi.liljeberg@utu.fi, juha.plosila@utu.fi

***Abstract*—Fault tolerance is an important aspect in designing Network-on-Chip (NoC) architectures for future multiprocessor systems. The paper presents dual network interface (dual-NI) NoC architectures for fault tolerant Network-on-Chip implementations. Network topologies and routing algorithms for dual-NI NoC architectures are presented and analyzed. The dual-NI architectures are based on connecting two network interfaces on each core (processing element). Additional network interfaces increase alternative routes to reach the cores and enter the network from cores in case of faulty links and routers. The aim is to improve fault tolerance on the topology level which means the delivery of packets to all the cores even when there are faulty links or routers in the network. The analysis shows that the fault tolerance of the NoC can be improved with dual network interface structures by decreasing the average hop counts and keeping the cores connectable even in the case of faults. The overhead of such structures is reasonable.**

I. Introduction

Shrinking technologies and simultaneously increasing complexity of the implemented systems increase the possibility of different faults on integrated circuits. There is a need for mechanisms to keep systems working even in the case of a few permanent faults in the system. The permanent faults may be a result of for instance by device aging or errors during the manufacturing process. The topology level fault tolerance mechanisms are designed to extend device lifetime and more importantly to increase the manufacturing yield.

The Network-on-Chip (NoC) paradigm [1], [2] provides several ways to develop mechanisms to increase system performance and reliability. A characteristic feature of NoC architectures is the existence of multiple alternative communication paths between any two cores (processing elements). Efficient utilization of these alternative paths may lead to an increased performance and reliability also in the case of permanent faults in the system.

The paper presents dual network interface (dual-NI) architectures to improve topology level fault tolerance in complex integrated systems. The topology level fault tolerance refers to network topology design which focuses developing topologies that support the system operation in the case of faults. Recent proposals focus on methods keeping the system working although some parts of the system are shut down [8]. A possible situation is that a router, which is attached to some core or a link connecting the router to the core is faulty. Especially in the case of a faulty router in topology level fault tolerant design it is typical to route packets around the router and not let the faultiness of the router affect the use of other cores in the system. However, in most of the cases like this, the core attached to the faulty router or link is no more usable.

The idea of dual-NI NoC architectures is to compose a system where faultiness of individual routers or links does not entirely disable the usage of the cores. A dual-NI structure has been previously applied to a tree topology [4] but here it is also analyzed for mesh topology networks. The idea of multiple network interfaces per one core has been shown to improve network fault tolerance [5].

The analyzed dual-NI NoC architectures are based on the structure where every core has two network interfaces connected to two different routers. In the case of a faulty router or link, the packets can be directed to and from the core through the other network interface instead of the one connected to the faulty parts of the network. Another advantage of multiple network interfaces is the possibility to shorten the paths between cores. The routing algorithm can be designed so that the selection of network interfaces at the sender and the receiver is primarily done to minimize the routing path between them.

This paper is organized as follows. The dual network interface architectures are discussed in Section II. Section III discusses routing algorithms for dual-NI architectures. The presented dual-NI architectures are analyzed and the simulation environment is presented in Section IV and finally the conclusions are drawn in Section V.

II. Dual Network Interface Architectures

A dual network interface NoC architecture is an architecture where each core has two network interfaces and is connected to the network via two different routers. Several known NoC topologies can be adapted to dual-NI architectures. For instance the conversion of mesh, tree and torus topologies to dual-NI topologies is quite straightforward. The addressing of the resources and the routing algorithm may require modifications to enable reliability and performance of dual-NI architectures.

A basic mesh NoC topology, consisting of 9 routers and 9 processing elements is presented in Fig. 1. The core–network connections are realized with a single network interface and a single connection from an core to a router. This is the typical way to implement a mesh NoC. The dual-NI extension of the mesh NoC is presented in Fig. 2. The presented dual-NI mesh network consists of 16 cores and 23 routers. Each core has

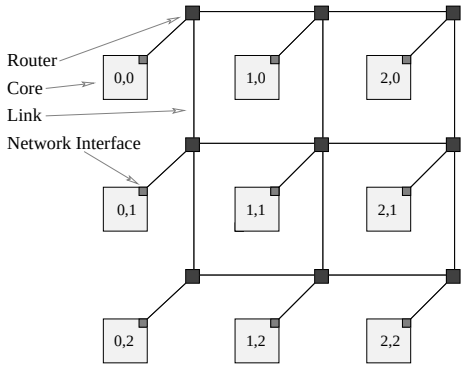

Fig. 1. A basic mesh topology consisting of 9 cores and 9 routers.

two network interfaces connected to different routers which makes it possible to reach every resource through at least two different paths.

The other analyzed Network-on-Chip topology is a tree which in its basic form is illustrated in Fig. 3. The basic tree topology has single core–network connections per core. The highest ancestor router on the top of the network is removed and replaced with a link connecting the two routers on the top of the tree. A dual-NI tree architecture is presented in Fig. 4. The links on each level of the tree are folded to enable the alternative routing paths between the multiple network interfaces of the cores.

The addressing of the network resources is illustrated inside the cores on the figures 1, 2, 3 and 4. On the mesh networks the addresses are based on the X and Y coordinates of the cores starting from the top left corner of the network. Processing elements on the tree networks are addressed with consecutive numbers.

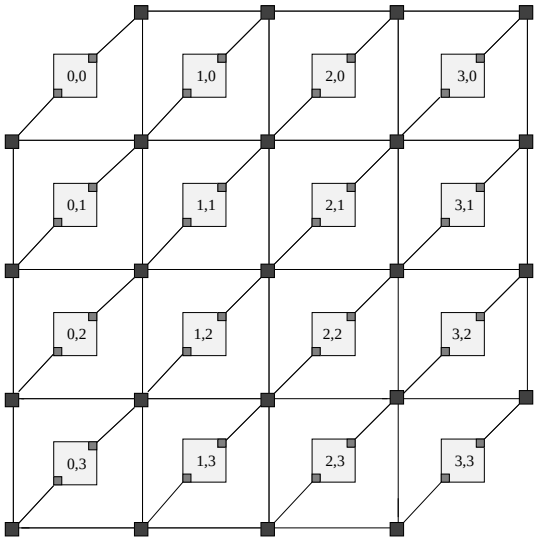

Fig. 2. A dual-NI mesh topology consisting of 16 cores and 23 routers.

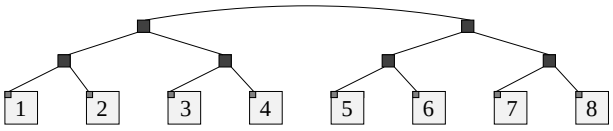

Fig. 3. A basic tree topology consisting of 8 cores and 6 routers.

III. Routing

The presented dual-NI NoC architectures utilize routing algorithms which are based on basic deterministic algorithms but improved with adaptive features. In fault tolerant NoC architectures it is essential to keep the routing straightforward to maximize its robustness and keep the area overhead and design complexity on a reasonable level. The routing algorithms, which are presented below in this section, work deterministically on faultless networks. When the primary routing direction is in an inoperable condition the adaptive routing algorithm chooses another path among the usable ones.

The dual-NI mesh NoC architecture utilizes a progressive dimension order routing algorithm [3], [7] which is based on well-known and widely applied XY routing algorithm. Each router has a simple routing table including the addresses of the cores attached to it. Basically a packet is routed along progressive path towards its destination by comparing the destination address in the packet and the addresses of the cores attached to the current router. If the address of the packet's destination has been found from the router's routing table, the packet is directed to the destination core. Otherwise the algorithm decides to which direction the packet has to be forwarded to get it closer to its destination core.

The packets are always directed to the progressive directions, which means that each hop in the network moves the packet towards its destination. The utilization of a progressive routing prevents generation of livelocks and minimizes network resource loading. In the case of faulty links in progressive directions the packets are directed to another progressive direction if there is one. If all the progressive directions are in unusable state the packets are dropped.

The routing algorithm on the presented dual-NI tree architecture is a turn-back routing which is previously used in fault tolerant tree architectures [4]. The routing is based on routing tables and moving packets up and down in the tree network. Each router has a routing table consisting addresses of all its child cores or cores which can be reached by routing packets downwards from the router. The router is called an ancestor of these child cores. If the destination is not included in the current routing table the packet is routed upwards in the network until it arrives to an ancestor of its destination. From there the packet is directed downwards towards its destination. At each point the router checks if the preferred links are faulty. In the case of faulty links a packet is directed back to the ancestor, or if it is not possible, the packet is dropped.

The dual-NI tree architecture utilizes a packet lifetime which defines how long a single packet can move in the network. The lifetime of the packets is determined on the

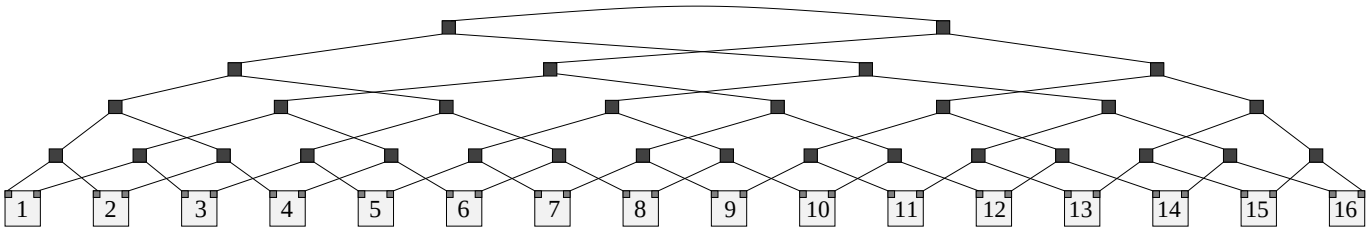

Fig. 4. A dual-NI tree topology consisting of 16 cores and 30 routers.

grounds of the network size so that the packets can be delivered if it is possible, but they are dropped if they cannot reach their destinations. The lifetime is realized with a counter value in the packet which is set when the packet is sent and decremented on each hop in the network.

IV. Analysis

Mesh (Fig. 2) and tree (Fig. 4) topologies with dual-NI structures were modelled and simulated. For comparison purposes there were also models of basic tree and mesh NoCs with one network interface per core (Fig. 3 and Fig. 1 both extended to contain 16 cores). Network models utilize progressive routing algorithms (see Section III) which observe the usability of links and can adaptively switch between progressive route alternatives when there are faulty links connected to a router.

Randomized traffic patterns were injected to the system and number of delivered packets as well as core connectability and average hop counts were monitored. The packet delivery analysis tracks the number of sent packets and monitors how many of them has been delivered to the destination. Core connectability is defined so that a core is connectable when it is able to send and receive packets through at least one functional link. Connectability of a core still does not necessarily guarantee the connectability between every source-destination pair in the network. The average hop count analysis monitors the lengths of the routing paths in the network. A hop is an event where a packet is processed in a router and moved to the next router. The hop count of a packet is the number of hops from the sender to the receiver.

A. Simulation Environment

The dual-NI NoC architectures were analyzed using *FANSI* Network-on-Chip simulator [6]. The simulation environment makes it possible to model traffic distribution and system behavior with different routing algorithms and in the presence of faulty routers and links. The simulator gives models for cores, routers, links and network interfaces which are used to form a model of a NoC. The connections between routers and cores and routing algortihm on the routers are fully customizable. There is also a model for a packet which is utilized to model traffic on the network.

The NoC architectures were tested by sending 1000 packets with random destination addresses from each core. The simulations were repeated with different number of faulty links in randomized locations in the network.

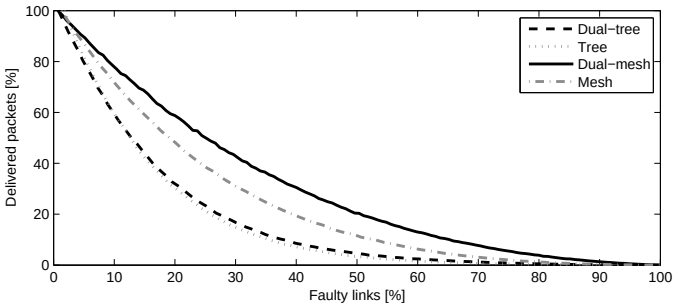

Fig. 5. Delivered packets as a function of faulty links in the network.

B. Results

The simulation results, including delivered packets, connectable cores and average hop counts are presented in the figures 5, 6 and 7 respectively.

Fig. 5 shows that the dual-NI mesh architecture is the most efficient of these four candidates to guarantee the packet delivery. The largest difference in packet delivery is approximately when one third of the links in the network are faulty. At that point the difference between the dual-NI mesh and the basic mesh is roughly 12 percentage units and between the dual-NI mesh and the dual-NI tree it is approximately 25 percentage units. The difference between the dual-NI and the basic tree architectures at that point is just two percentage units. Hence, the presented mesh architecture improves the topology level fault tolerance significantly but the tree architecture does not reach any notable reliability increase with respect to its complexity. The bottleneck of the tree network performance and fault tolerance may still be the small number of alternative routes in the network. Improvement to the tree architecture could be reached by using some fat tree topology with multiple links between the routers.

As can be seen from the Fig. 6 the connectability of cores is improved significantly with the dual-NI architectures. The connectability is only dependent on the number of network interfaces on a single core and is independent of the network topology.

The average hop counts in Fig. 7 describe the length of possible routing paths in a faulty network. The decreasing hop count as a function of faults in Fig. 7 is because the packets with the longest routing paths are most probably dropped and the utilization of these paths is not possible. The figure

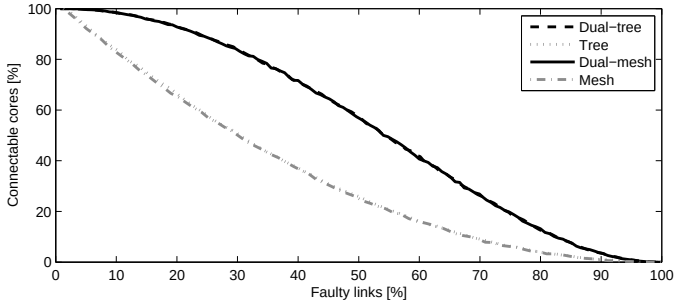

Fig. 6. Connectable cores as a function of faulty links in the network.

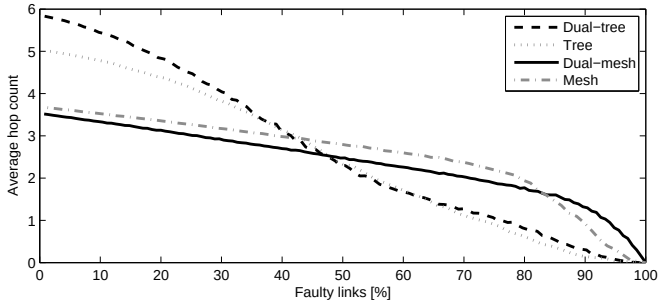

Fig. 7. Average hop counts as a function of faulty links in the network.

shows that in the mesh network the dual-NI structure does not increase the network latency but actually decreases it slightly because of the added shorter route alternatives. In the tree network the effect of added resources is more notable. As Fig. 7 shows, the average hop counts increase in faultless network but the difference decreases when the number of faults grows. The hop count increment on the faultless tree network is due to the added resources on the dual-NI tree network compared to the basic tree. With a very high percentage of faulty links the average hop counts collapse to zero because most of the packets are dropped in the beginning of their routing paths.

C. Design Complexity

The complexity of the different solutions is analyzed by comparing the numbers and sizes of different components on each architecture. The router is the component whose complexity varies in different NoC topologies. The complexity of routing algorithm affects the router complexity but even more significant is the amount of required buffer registers. The number of buffers is related to the number of input/output ports in the router, so the number of I/O ports is used to illustrate the router complexity. This parameter, combined with the numbers of cores and routers, illustrates quite well the complexity of the network. These three complexity parameters for different structures are presented on Table I.

The dual-NI tree network is slightly less complex than the dual-NI mesh network. However, if the performance of the tree network is improved by extending it to a fat-tree, the design complexity increases significantly.

TABLE I
TOPOLOGY COMPLEXITY PARAMETERS.

Topology	*Cores*	*Routers*	*Sum of router ports*
Basic mesh	16	16	64
Dual-NI mesh	16	23	103
Basic tree	16	14	42
Dual-NI tree	16	30	90

V. CONCLUSIONS AND FUTURE WORK

The presented analysis shows that the dual-NI Network-on-Chip architectures can be utilized to improve system fault tolerance on topology level. They also enable a performance increase in some cases by introducing more alternative routes and shortening distances between cores. An increased area and more complex design and logic are the main drawbacks.

Among the analyzed architectures the dual-NI NoC architecture with mesh topology has been found to be the most efficient to improve the topology fault tolerance with respect to the used metrics. The increase on the packet delivery is up to 12 percentage units depending on the amount of faults in the network.

Future work includes expansion of the dual NI architectures to other topologies as well as development of the routing algorithms for them. The architectures should also be analyzed with respect to congestion and performance which requires throughput and latency analysis. Implementation and architectural aspects of the dual-NI IP cores should also be analyzed.

ACKNOWLEDGEMENT

The authors would like to thank the Academy of Finland and the Nokia Foundation for financial support.

REFERENCES

[1] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, January 2002.

[2] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Design Automation Conference, 2001. Proceedings*, June 2001, pp. 684–689.

[3] M. Dehyadgari, M. Nickray, A. Afzali-kusha, and Z. Navabi, "Evaluation of pseudo adaptive XY routing using an object oriented model for NoC," in *Microelectronics, 2005. ICM 2005. The 17th International Conference on*, December 2005.

[4] K. Kariniemi and J. Nurmi, "Fault tolerant XGFT network on chip for multi processor system on chip circuits," in *Field Programmable Logic and Applications, 2005. International Conference on*, August 2005, pp. 203–210.

[5] T. Lehtonen, P. Liljeberg, and J. Plosila, "Fault tolerance analysis of NoC architectures," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, May 2007, pp. 361–364.

[6] T. Lehtonen, V. Rantala, P. Liljeberg, and J. Plosila, "Fansi: Fault tolerant Network-on-Chip simulator," Turku Centre for Computer Science (TUCS), Tech. Rep. 935, March 2009.

[7] V. Rantala, T. Lehtonen, P. Liljeberg, and J. Plosila, "Distributed traffic monitoring methods for adaptive Network-on-Chip," in *NORCHIP, 2008.*, November 2008, pp. 233–236.

[8] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2D-mesh Network-on-Chip," in *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, June 2008, pp. 441–446.