A VLSI Architecture for Enhancing the Fault Tolerance of NoC using Quad-spare Mesh Topology and Dynamic Reconfiguration

Yu REN, Leibo LIU*, Shouyi YIN, Qinghua WU and Shaojun WEI
Institute of Microelectronics and the National Lab for Information Science and Technology
Tsinghua University, Beijing 100084, China
Email: liulb@tsinghua.edu.cn

Jie HAN
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB, Canada T6G 2V4

***Abstract*—Effective fault tolerant techniques are crucial for a Network-on-Chip (NoC) to achieve reliable communication. In this paper, a novel VLSI architecture employing redundant routers is proposed to enhance the fault tolerance of an NoC. The NoC mesh is divided into blocks of 2×2 routers with a spare router placed in the center. The proposed fault-tolerant architecture, referred to as a quad-spare mesh, can be dynamically reconfigured by changing control signals without altering the underlying topology. This dynamic reconfiguration and its corresponding routing algorithm are demonstrated in detail. Experimental results show that the proposed design achieves significant improvements on reliability compared with those reported in the literature.**

I. INTRODUCTION

As an increasing number of processing elements (PEs) has been integrated on a single chip, a variety of interconnection schemes have been proposed, including crossbars, rings, buses, and Network-on-Chip (NoC) [1]. The packet-based NoC is considered a promising solution to the interconnection challenges of future SoC designs [2]. It is scalable and has been widely utilized to decouple communication from computation, thus improving performance.

The reliability of an NoC is critical to guarantee the reliability of communication. Many solutions have been proposed to improve the reliability of a system; these include fault tolerant routing algorithms [3,4] and various topologies for implementing the communication infrastructure [5,6]. These methods, however, cannot make use of the good PEs when there are faulty routers in the network.

With the increasing circuit density and more hardware resources on chip available, redundancy techniques have been widely used for fault tolerance. Redundancy can be achieved at different levels, including microarchitecture level [7,8], core level [9] and router level [10]. Router-level redundancy is considered efficient as the mesh size increases. In a router-level redundant design proposed in [10], each column has a spare router located on the top row. This method works well, however the reconfiguration is complicated and cannot be performed dynamically.

In this paper, a novel router-level redundant scheme, referred to as a quad-spare mesh, is proposed to solve the problem of isolated PEs in the presence of faulty routers. In the quad-spare mesh, an NoC is divided into 2×2 blocks. One spare router is then placed in the center of each block. Network reconfiguration is performed dynamically by control signals without altering the underlying topology. Hence, this process is transparent to the operating system and application software. The design is evaluated with fault tolerance metrics such as system reliability and mean time to failure (MTTF). Simulation results show that this scheme has achieved significant improvements by these metrics compared with those reported in [10]. For a 10×10 NoC, the MTTF of the proposed scheme is 40.0% higher than that of the design in [10]. In an NoC with a single error, the maximum throughput only decreases by 5.19%, compared with the error-free case.

The rest of this paper is organized as follows: Section II describes the proposed fault tolerant NoC architecture. In Section III, evaluation and experimental results are presented. Finally, Section IV concludes this paper.

II. DESIGN OF THE QUAD-SPARE MESH

In this section, the proposed quad-spare mesh topology is first presented. The topology reconfiguration and routing algorithm based on a bypass mechanism is then introduced. This topology reconfiguration can work dynamically, i.e., the faulty routers can be replaced by spare ones while the NoC is working.

A. Quad-spare Mesh Topology

The topology of a quad-spare mesh is shown in Fig. 1. The original 2D mesh is partitioned into blocks. Each block has an

978-1-4673-5762-3/13/$31.00 ©2013 IEEE

array of 2×2 routers, with one spare router placed in the center. The spare router can replace any faulty router in the block. Hence, this design is referred to as a quad-spare mesh.

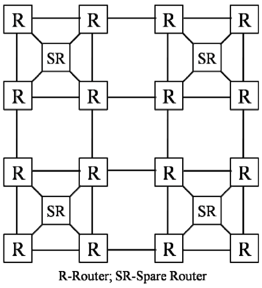

Figure 1. Quad-spare mesh topology

The communication path between a PE and a router/spare router in one block is shown in Fig. 2. Communication channels are divided into input and output channels. For clarity, the input and output channels of PEs in one block are shown in Fig. 3 (a) and (b). The input port of each PE is connected to the original router and spare router by a 2-to-1 multiplexer (MUX). As to the output port, each PE is also connected to the original router and spare router. Because the spare router has only one input port, a 4-to-1 MUX is used to select the input from the PE whose router is replaced by the spare router.

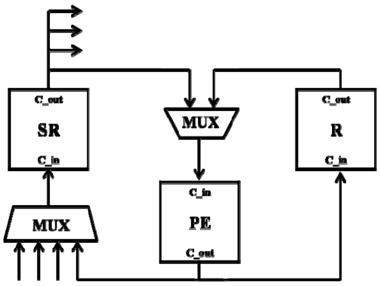

Figure 2. PE connections in a block

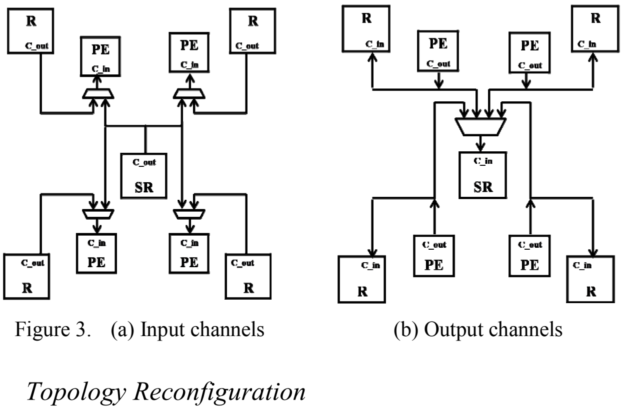

Figure 3. (a) Input channels (b) Output channels

B. Topology Reconfiguration

The reconfiguration works dynamically. Faults can be captured by add-on mechanisms such as the Built-In-Self-Test (BIST). When an error is detected, a message is sent to the topology-reconfiguration controller, so that adjacent routers will know which of their neighbors is faulty and then packets will be sent to spare routers. The topology is reconfigured just by changing the control signals. From the perspectives of the operating system and application software, the topology of the NoC and the routing algorithm remain unchanged. Moreover, the system does not have to stop working to reconfigure the network. As a result, the proposed method is transparent to the upper layer, which is a great advantage.

Fig. 4 shows an example of a 4×4 NoC reconfiguration. For clarity, PEs are not shown in this figure. In this example, Routers 5 and 14 are assumed to be faulty, so they are not shown in Fig. 4. The faulty routers are replaced by the spare ones in their blocks respectively, i.e., Router 5 is replaced by SR0 and Router 14 is replaced by SR3.

A bypass mechanism is used for paths between a spare router and routers in other blocks (e.g., SR0 to R6 and R9 in Fig. 4). Packets from Routers 6 and 9 are both sent to the spare router, while for packets sent from SR0 to R6 or R9, a switch is used. The switch works together with the spare router. The selection signal for this switch is generated by the routing module of the spare router. When an error occurs and a spare router is used, the switch selects the routing path, just as a router. The connecting paths between routers and PEs can be reconfigured by changing the selection signals of the MUXs in Figs. 2 and 3 by the controller.

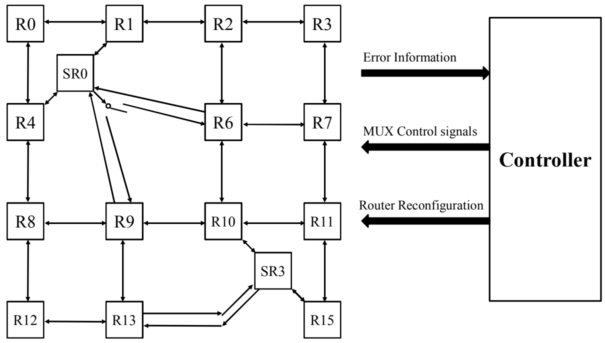

Figure 4. An NoC reconfiguration example

C. Routing Algorithm

When an error is detected in the router, the error information will be sent to the controller, and then the routing tables of adjacent routers will be reconfigured by the controller. In addition, the spare router will also be reconfigured, based on the direction of the faulty router. If the spare router is routing to the direction of the faulty router, a switch is used, for which the selection signal is generated by the spare router accordingly. In this way, packets can be transmitted between original routers and spare routers following the routing tables.

III. EVALUATION AND EXPERIMENTAL RESULTS

A. Reliability Analysis

A system is considered to be reliable if all the PEs function properly and exchange data correctly. In a traditional $N \times N$ mesh, if one router fails to work, the data being transmitted may get lost. An NoC with spare routers, however, has the advantage of replacing the faulty router with a spare one. With more faulty routers in a system, an NoC with spares can either replace all the faulty routers and continues to work, or cannot repair the network and fails to work. Whether it will work or not depends on the pattern of faulty routers. The reliability of an NoC system is defined as the probability of the system to function correctly, provided that each router works independently and fails with an error probability. Given the mesh size and the error rate of each router, 10,000 patterns are randomly generated. The reliability of the NoC is then measured by the statistical outcomes from the network simulation.

To make a comparison, another router-level redundant scheme in [10] is also simulated. Since it has a spare router in each column, it is referred to as a column-spare NoC in this paper. The following figure shows the reliability of both quad-spare and column-spare NoCs in different sizes with various router failure rates. It can be clearly seen that the reliability of the quad-spare NoC is higher than the column-spare NoC.

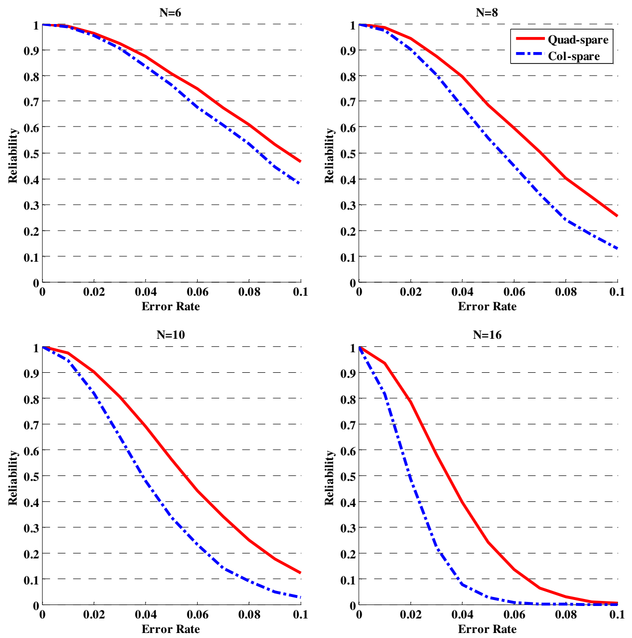

Figure 5. Reliability of various spare NoCs with different router failure rates (the proposed design vs. the design in [10])

B. *Time Degradation Reliability Analysis*

The performance of hardware may degrade with time. Next, time degradation is taken into consideration. The reliability of an NoC router $R_r(t)$ is defined as the probability that a router works correctly from time 0 to *t*. It is determined by the failure rate $\lambda(t)$, which is a measure in the number of failures per time unit. After the router is initialized, $R_r(t)$ is given by the exponential failure law as [11]:

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

The analytical models for the traditional mesh, column-spare mesh and quad-spare mesh are given by (2) (3) and (4). In an $N \times N$ network, a traditional mesh without redundancy requires all of its routers working properly, so its reliability is given by (2). The column-spare mesh is spilt into *N* columns. Therefore every column must function correctly to ensure the system's functionality. The reliability of such a system is then the product of each column's reliability, R_c. Each column has *N+1* routers. A column works correctly if either all the routers within the column are operational, which gives a reliability of $R_r^{\ N+1}$, or *N* of the *N+1* routers are operational. In the latter case, one of the *N+1* routers is faulty, resulting in a reliability of $C_N^{N+1}(1-R_r)(R_r)^N$. Therefore, the reliability of the column-spare mesh is given by (3). As to the quad-spare mesh, a system has $(N\ /\ 2)^2$ blocks and every block has to work correctly to ensure the system's reliability. A block works correctly if at least four of the five routers in the block are operational. This leads to the reliability of the quad-spare mesh given by (4).

$$R_{sys\_mesh}(t) = (R_r)^{N \times N} \tag{2}$$

$$R_{sys\_col\_spare}(t) = (R_c)^N = \left[R_r^{\ N+1} + C_N^{N+1}(1-R_r)(R_r)^N\right]^N \tag{3}$$

$$R_{sys\_quad\_spare}(t) = (R_{block})^{(N/2)^2} = \left[R_r^{\ 5} + C_4^5(1-R_r)(R_r)^4\right]^{(N/2)^2} \tag{4}$$

In the reliability analysis of (2) (3) and (4), the failure rate of a router is assumed to be $\lambda = 0.00315$ (times/year) [12]. The reliability of various meshes with different sizes over 1 to 10 years is shown in Fig. 6. As can be seen, the spare mesh architecture outperforms the traditional one and the quad-spare mesh has a higher reliability than the column-spare mesh. As the mesh size increases, the reliability decreases. The quad-spare mesh has the most slowly decreasing reliability – it still maintains a reliability of nearly 0.8 in the 10th year.

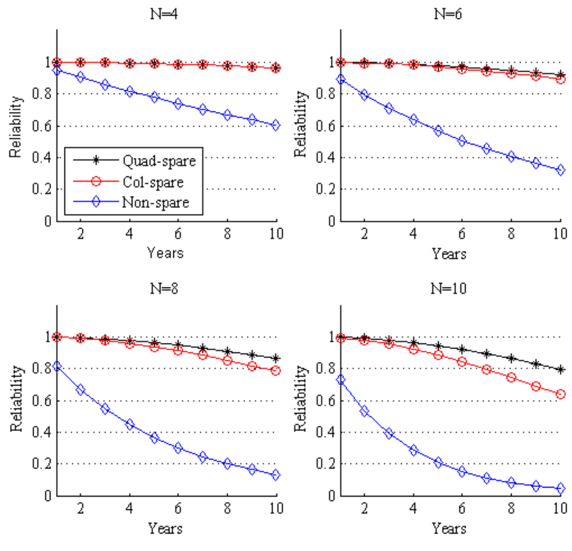

Figure 6. Reliability in time of the traditional non-spare NoC, the column-spare NoC and the proposed quad-spare NoC for different sizes.

C. *Mean Time to Failure Analysis*

The mean time to failure (MTTF) is the average time before a system fails, which is usually given as a function of the reliability in time,

$$MTTF = \int_0^\infty R(t)dt \tag{5}$$

Taking (2) (3) and (4) into (5), the MTTF for a traditional, column-spare and quad-spare mesh can be derived. The MTTF for different mesh sizes is shown in Fig. 7. The gain of MTTF is also depicted to highlight the effect of spare NoC with the NoC size. For a 10×10 NoC, the MTTF of the column-spare mesh is 1.236×10^5 hours, while the proposed scheme is 1.731×10^5 hours, which is 40.0% higher than the column-spare mesh.

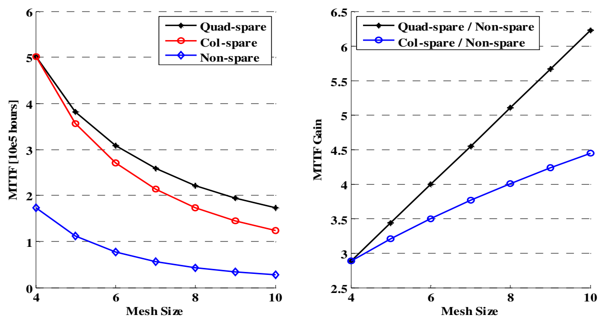

Figure 7. MTTF and MTTF gain of the traditional non-spare NoC, the column-spare NoC and the proposed quad-spare NoC for different sizes

D. *Throughput*

A flit-level simulator in C++ is used for measuring the throughput. In the simulation, 100 different fault patterns are randomly generated. All PEs are assumed to be healthy, with only one randomly chosen PE injecting packets. The throughput is the average number of flits received per clock cycle.

Fig. 8 shows the throughputs of 4×4 and 6×6 quad-spare NoCs, averaged over 100 different fault patterns. As revealed in the figures, an error-free network has the highest throughput. The throughput decreases with the number of faulty routers. However, in an $N \times N$ network with only a few faulty routers, our proposed design has the advantage of keeping the throughput almost as high as an error-free network. This is because the spare routers can replace some faulty routers, thus maintaining the functionality of the network and the throughput.

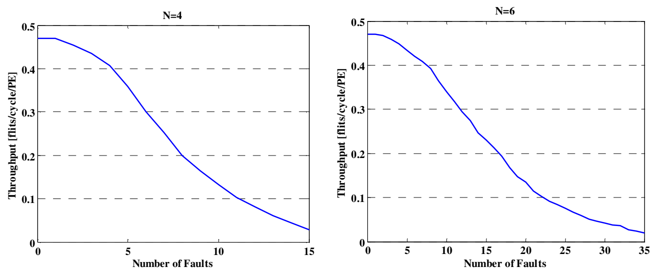

Figure 8. Average throughput for 16 PEs and 36 PEs.

IV. Conclusions

In this paper, a VLSI architecture employing router-level redundancy, referred to as a quad-spare mesh, is proposed for fault tolerant NoC designs. This design deals with the problem that a faulty router breaks the communication between healthy PEs. The proposed design significantly improves a system's reliability and its mean time to failure. Topology reconfiguration and routing algorithm can be performed dynamically. The NoC after reconfiguration is consistent to the original network, which implies that this design is transparent to the upper layers including operating systems and user applications.

In the proposed quad-spare mesh, the reliability of an NoC is improved by using spare routers, which are regularly connected to as many original routers as possible. Therefore, the hardware overhead is kept low with a high throughput. This idea on the use of redundancy is not restricted in the 2D-mesh topology; it could also be used in NoCs with other types of topologies and could further be extended to tolerate faulty links. The proposed fault tolerant architecture is therefore scalable and potentially useful in future NoC designs.

References

[1] Krewell, "Multicore showdown," Microprocessor Report, vol. 19, pp. 41-45, 2005.

[2] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in Proc. ACM/IEEE Design Automation Conf., Jun. 2001, pp. 18-22.

[3] M. Pirretti, G. M. Link, et al, "Fault tolerant algorithms for Network-on-Chip interconnect," in Proc. ISVLSI, 2004, pp. 46-51

[4] Z. Zhang, A. Greiner, S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2D-mesh Network-on-Chip," DAC 2008, pp.441-446

[5] H. Kariniemi, "Fault-tolerant XGFT Network-on-Chip for multi-processor System-on-Chip circuits", Proceedings of the International Conference on Field Programmable Logic and Applications, Finland, 2006, pp. 186-190

[6] P. P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a switch for Network-on-Chip applications," IEEE International Symposium on Circuits and Systems (ISCAS) 5: 217–220, 2003.

[7] P. Shivakumar, S. W. Keckler, C. R. Moore, and D. Burger, "Exploiting microarchitectural redundancy for defect tolerance," in Proc. IEEE Int. Computer Design Conf., Oct. 2003, pp. 481-488

[8] E. Schuchman and T. N. Vijaykumar, "Rescue: a microarchitecture for testability and defect tolerance," in Proc. IEEE/ACM Int. Symp. Computer Architecture, Jun. 2005, pp. 160-171.

[9] L. Zhang, Y. Han, Q. Xu, X. Li, H. Li, "On Topology Reconfiguration for Defect-Tolerant NoC-Based Homogeneous Manycore Systems," in IEEE Trans. on VLSI Systems, 17(9): 1173-1168, 2009.

[10] Y. C. Chang, C. T. Chiu, S. Y. Lin, C. K. Liu, "On the design and analysis of fault tolerant NoC architecture using spare routers", ASP-DAC 2011, pp. 431-436.

[11] L.T Wang, C.E Stroud, and N.A Touba, System-on-Chip Test Architectures: Nanometer Design for Testability, Morgan Kauffmann, 2008

[12] W.J Dally and B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2003