



# Scenario preprocessing approach for the reconfiguration of fault-tolerant NoC-based MPSoCs



Jarbas Silveira <sup>a,\*</sup>, César Marcon <sup>b</sup>, Paulo Cortez <sup>a</sup>, Giovanni Barroso <sup>c</sup>, João M. Ferreira <sup>a</sup>, Rafael Mota <sup>a</sup>

<sup>a</sup> LESC-DETI, Federal University of Ceará, Fortaleza 90455-970, Brazil

<sup>b</sup> PPGCC, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre 90619-900, Brazil

<sup>c</sup> Department of Physics, Federal University of Ceará, Fortaleza 90455-970, Brazil

## ARTICLE INFO

### Keywords:

NoC  
Irregular topology  
Fault-tolerance  
Preprocessing  
Routing methods

## ABSTRACT

The latest technologies of integrated circuit manufacturing allow billions of transistors to be arranged on a single chip, enabling the chip to implement a complex parallel system, which requires a communications architecture that has high scalability and a high degree of parallelism, such as a Network-on-Chip (NoC). These technologies are very close to the physical limitations, which increases the faults in manufacturing and at runtime. Therefore, it is essential to provide a method for fault recovery that would enable the NoC to operate in the presence of faults and still ensure deadlock-free routing. The preprocessing of the most probable fault scenarios enables us to anticipate the calculation of deadlock-free routings, reducing the time that is necessary to interrupt the system during a fault occurrence. This work proposes a technique that employs the preprocessing of fault scenarios based on forecasting fault tendencies, which is performed with a fault threshold circuit operating in accordance with high-level software. We propose methods for dissimilarity analysis of scenarios based on cross-correlation measurements of link fault matrices. Experimental results employing RTL simulation with synthetic traffic prove the quality of the analytic metrics that are used to select the preprocessed scenarios. Furthermore, the experiments show the efficacy and efficiency of the proposed dissimilarity methods, quantifying the latency penalization when using the coverage scenarios approach.

© 2015 Elsevier B.V. All rights reserved.

## 1. Introduction

The evolution of Very-Large-Scale Integration (VLSI) semiconductor technology enables us to integrate hundreds or even thousands of cores into a single circuit. This massive integration allows us to implement the entire functionality of a system into a single chip, generating a System-on-Chip (SoC). The International Technology Roadmap for Semiconductors (ITRS) foresees thousands of Processing Elements (PEs) integrated into an SoC by 2020 [1]. The Network-on-Chip (NoC) technology will play a key role in the implementation of these highly integrated SoCs. The two-dimensional (2D) mesh is the most popular NoC topology; it offers a simple and regular structure, and the small wire length is suitable for the tile-based design [2]. Under the NoC communication paradigm, each PE that is placed in a tile is interconnected throughout with routers and links to other PEs, and the communication is normally performed by packet transmission [3].

Recent submicron technologies provide more process variability, increasing the quantity of defective components [4,5]. A defective router or link could ruin the 2D mesh communication structure, which would lead to an irregular topology (i.e., topologies derived from regular networks with induced faults or faulty links, also known as agnostic topologies) [6]. As a consequence, static and deterministic routing algorithms that are tailored to a regular NoC topology will not operate properly, thus rendering the chip useless [2]. Aiming to avoid this problem, the design of the routing algorithms must provide some degree of fault-tolerance while ensuring deadlock-freeness. There are two approaches to achieve deadlock-free routing algorithms for irregular topologies: (i) one approach is based on virtual channels (e.g., [7–12]), which requires a large extra area for multiplexing schemes and implies significant energy consumption by the buffers of the routers; and (ii) another approach is based on turn prohibition (e.g., [13–17]), which avoids deadlock by eliminating a subset of network turns. Our work employs this last approach using tables in each router that implement the routing algorithm. However, the area consumption and power dissipation make table-based techniques prohibitive for large networks. Aiming to overcome

\* Corresponding author.

E-mail addresses: [jarbas@lesc.ufc.br](mailto:jarbas@lesc.ufc.br) (J. Silveira), [cesar.marcon@pucrs.br](mailto:cesar.marcon@pucrs.br) (C. Marcon), [cortez@deti.ufc.br](mailto:cortez@deti.ufc.br) (P. Cortez).

these problems, several studies (e.g., [6,18,19]) have proposed to compress the routing table. Our work employs a technique that is similar to [6], which compresses the routing table according to the NoC regions.

Efficient implementations of routing strategies address the dynamicity of the faults with fast computation and reconfiguration of the fault-tolerant topology without requiring packet retransmission. This fast computation and reconfiguration could combine some static and/or dynamic tasks. For example, all of the possible fault scenarios could be computed before the system operation, which characterizes a static computation in opposition to the approach that computes new scenarios only during the system operation. Independent of the computation dynamicity, the reconfiguration could occur statically or dynamically. In the dynamic network reconfiguration, the entire system remains operating and the network traffic is not stopped, which implies that the router must know all of the communication possibilities to avoid deadlock conditions. On the other hand, in the static approach, the network enters into a reconfiguration phase, where it is halted and all packets are drained.

If each PE (connected to a router) preprocesses and stores scenarios, the system reconfiguration could be improved, which would reduce the time that the network is halted waiting for the routing algorithm computation that handles the topology changes. Unfortunately, preprocessing all of the possible fault scenarios is a very complex problem, which consumes time and memory. Aiming to minimize this problem, a system could statically elect and store only some scenarios to be used in an appropriate situation, such as a virtualization perspective or low power scenario. Additionally, in a given set of fault scenarios, there are some scenarios that can be discarded because they are covered by others, which enables us to minimize the memory area that is required to store the preprocessed scenarios.

This work employs Phoenix [20] as the fault-tolerant target system, which consists of a 2D mesh NoC and a layer for an operating system. Using Phoenix, this paper proposes an efficient approach to addressing dynamic faults on NoC links. This approach consists of preprocessing fault scenarios to quickly reconfigure the network with new deadlock-free routing, in the case of fault detection. Additionally, this work proposes a method for reducing a large set of scenarios based on cross-correlation to identify dissimilarities in sets of irregular topologies. This method consists of fault-monitors that are placed in each input port of the Phoenix NoC routers. Each fault-monitor evaluates whether the link is faulty or if there is a fault tendency. This information is transmitted to a PE that is locally connected to the router, and a communications driver placed inside the operating system preprocesses new scenarios based on this fault information.

This paper is organized as follows. Section 2 presents related work on routing mechanisms and reconfiguration processing. Section 3 describes the Phoenix architecture, which is the target architecture employed here. Section 4 describes the processing flow of fault-tolerant scenarios. In Section 5, we present the fundamentals of the fault scenario preprocessing. Section 6 presents some synthesis results of Phoenix for a specific MPSoC architecture. Section 7 describes the methodology and experimental results of the exploration of analytic metrics for runtime latency estimation, while Section 8 analyzes the methods and costs for the preprocessing approach. Finally, Section 9 presents the main conclusions of this work.

## 2. Related work and main contributions

Strategies applied to fault-tolerant routing for irregular NoCs can employ static or dynamic fault models. In the first strategy, all of

the tolerable faults must be known at the design time. Then, during the system operation, when a fault is discovered, the system is halted, the packets are dropped, and a new deadlock-free routing is employed. The second strategy does not need to halt the network, and the routing computation is performed at runtime. Moreover, the second strategy requires complex routing distributed mechanisms to avoid inconsistencies (e.g., in routing tables) during the network reconfiguration. The strategies that use static fault models (e.g., [21,22]) are less complex and less area consuming, but they can tolerate only a limited quantity of faults [23]. On the other hand, the routing mechanisms of strategies that use dynamic fault models (e.g., [23–25]) are more complex, but they allow many more changes to the network topology. This work focuses on this last approach.

A reconfigurable fault-tolerant system for irregular networks requires mechanisms for (i) fault detection and diagnosis; (ii) fault recognition reporting; (iii) deadlock-free routing computation; and (iv) **routing reconfiguration** (e.g., routing tables and auxiliary circuits). This section discusses studies that are related to the last mechanism, which is the main focus of this work, accounting for only network architectures *without virtual channels*. Additionally, the routing reconfiguration is composed of the **routing mechanism** and **reconfiguration processing**; both are discussed next.

The **routing mechanism** defines how the routing information is computed and stored. To support reconfiguration, the routing mechanism needs resources that could be changed during the runtime, which normally encompasses tables placed in each NoC router, storing the routing information. Routing tables support a large quantity of topologies and are easy to implement. On the other hand, they do not scale with an increase in the NoC size. Aiming to reach the scalability that is required for current and future highly populated NoCs, several studies employ techniques to compress or minimize the sizes of the routing tables, which is a complex task and could imply a loss of performance and/or an impossibility of reaching all of the target nodes. Examples of these studies are (i) Palesi et al. [18], which uses a table compression technique for application-specific routing; and (ii) Bolotin et al. [19], which uses a table minimization technique that applies a fixed function combined with minimal deviation tables.

Scalability is also achieved by dividing the network into regions and mapping those regions onto routing table entries. For example, Mejia et al. [6] proposed the Region Based Routing (RBR) approach, where each node contains a set of regions that are based on paths that cover all of the communications. RBR could be employed with a specific routing application, enabling the use of a small number of regions and assuring full network coverage. Fukushima, Fukushi and Yairi [26] propose another region-based approach that is based on a set of rectangular faulty regions and the corresponding deviation paths, which are employed to avoid the faulty regions. Their approach improves the work of Holsmark et al. [27], providing a complete and deadlock-free routing that reduces the faulty regions' sizes. Consequently, it reduces the number of nodes to be deactivated and the routing implementation complexity. Although the region-based approaches scale with the NoC size, it is necessary to have a large number of regions to maintain full NoC coverage with efficient paths, which prohibits a large routing table reduction. Aiming to overcome this problem, Rodrigo et al. [2] improved the previous LBDR work [28], proposing the universal Logic-Based Distributed Routing (uLBDR), which provides an efficient routing mechanism using a small set of configuration bits together with logic cells instead of large routing tables implemented in memories. The main limitation of their work is that to cover all of the deadlock situations, the approach must use virtual cut-through switching, which limits the packet size to the length of the buffers.

Download English Version:

<https://daneshyari.com/en/article/462632>

Download Persian Version:

<https://daneshyari.com/article/462632>

[Daneshyari.com](https://daneshyari.com)