Microprocessors and Microsystems 36 (2012) 531–542

Contents lists available at SciVerse ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

An efficient routing methodology to tolerate static and dynamic faults in 2-D mesh networks-on-chip

Farshad Safaei *, Majed ValadBeigi

Faculty of ECE, Shahid Beheshti University GC, Evin 1983963113, Tehran, Iran

ARTICLE INFO

Article history:
Available online 12 June 2012

Keywords:
Networks-on-chip
Fault-tolerance
Mesh topology
Dynamic fault
Static fault
Fault regions

ABSTRACT

The move towards nanoscale Integrated Circuits (ICs) increases performance and capacity, but poses process variation and reliability challenges which may cause several faults on routers in Networks-on-Chips (NoCs). While utilizing healthy routers in an NoC is desirable, faulty regions with different shapes are formed gathering faulty routers. Fault regions can be used to lead the fault-tolerant routing algorithms to perform data transmission between healthy routers. In this paper a distributed fault-tolerant routing methodology for mesh networks is proposed which supports static and dynamic fault model. The static fault model supports minimal routing path which tolerates both convex and concave fault regions, while keeping the area and power overhead at a minimum level. Moreover, unlike most previous methods that support dynamic fault models, the presented method is able to tolerate any number of faults with any shapes of fault regions without disabling healthy nodes. The performance of the method is extensively evaluated, and the results show that our proposed method is valid for mesh topology, which has graceful performance degradation and allows the network to remain fully operational facing with the failures.

1. Introduction

The advances in the semiconductor technology and the shrinking feature size in the deep submicron era, have led to an impressive increase in number of transistors available on a single chip. This huge number of transistors enables the integration of complex system-on-chip (SoC) designs containing a large number of processing cores together with large amounts of embedded memory and high-bandwidth I/O on a single chip. On the other hand, the applications are becoming more and more complex. For example, a modern cell phone not only should support audio processing but also needs to process images, videos, networks and internet protocols and support operating systems [1].

The large amount of computational resources together with complicated communication patterns due to complex applications requires higher degrees of support from communication resources [1]. The lack of scalability in bus-based systems and large area overhead of point-to-point dedicated links on one hand, and the scalability and performance of switch-based networks and packet-based communication in the internet and traditional parallel computing, on the other hand, have motivated researchers to propose packet-switched Network-on-Chip (NoC) architectures to overcome complex on-chip communication problems.

Although the concepts of NoC are inspired from the networks in parallel processors, they have some special properties which differ from the traditional networks. Compared to traditional networks, power consumption and fault-tolerance are becoming increasingly important constraints in NoC design [2]. With the shrinking feature size, the reliability becomes more difficult to achieve due to smaller voltage supplies and higher amount of computing elements per area unit, which result in smaller noise immunity and increased risk of cross-talk and other errors [2–4]. Moreover, critical leakage currents, high field effects, and process variation will lead to more transient and permanent failures of signals, logic values, devices, and interconnects [5].

In addition to the faults occurring during the system life-time, aggressive technology scaling has resulted in manufacturing and testing challenges [2]. Hence, a considerable portion of fabricated chips including some faults, forms the beginning of their life-time. Relaxing the requirement of 100% correctness in operation for devices and interconnects when designing chips may also dramatically reduces the costs of manufacturing, verification, and test [6]. This suggests that chips have to be designed with some built-in fault-tolerant techniques.

As communication infrastructure is an important issue of today's SoCs and Chip-Multiprocessors (CMPs), fault-tolerance has to be taken into account along with the design of this part of chips. Although having the same concept as traditional interconnection networks, the resources used in traditional networks in order to achieve fault-tolerance are not easily available in VLSI chips. Rout-

* Corresponding author. Tel.: +98 21 22904166.
E-mail addresses: f_safaei@sbu.ac.ir (F. Safaei), m.valadbeigi@mail.sbu.ac.ir (M. ValadBeigi).

0141-9331/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2012.05.014

ing algorithm, one of the most important aspects of NoCs, has been exploited by many researchers to guarantee a reliable operation of the network [7–9], by bypassing the faulty nodes and links. Deterministic algorithms do not behave well in the presence of random failures [10,11]. On the other hand, implementing adaptive dynamic routing for on-chip networks is prohibitive because of the need for very large buffers, lookup tables, and complex shortest-path algorithms [12].

To address the above mentioned problems, the goal of this paper is to design, implement, and evaluate a new fault-tolerant routing methodology which can be incorporated with suitable fault-tolerant routing algorithms to prevent static and dynamic fault patterns in NoCs with 2-D mesh topology.

The rest of the paper is organized as follows. In Section 2, a preliminary description of network architecture is introduced. This section also reviews some backgrounds about the concepts of fault models pertained to fault-tolerant routing used in the paper. In Section 3, we review previous work relevant to this paper. Our fault-tolerant approach and the metrics for performance evaluation in 2-D mesh are introduced in Section 4. The extension to the dynamic fault model and dynamic transition phase are described in Section 5. In Section 6, the experimental results with different fault regions and other related parameters are shown. Finally, in Section 7 we conclude by summarizing our main contribution.

2. Preliminaries

This section begins with a discussion of mesh networks and then describes the necessary background information that is used in the paper.

2.1. Mesh networks

The topology of a network defines how the nodes are interconnected and is generally modeled as a graph in which the vertices represent the nodes and the edges denote the communication lines. The NoC concept has been proposed to overcome the problems of traditional wire/bus-based interconnections [2]. The NoC uses interconnection networks to connect intellectual properties (IPs) and to cope with the increased size and complexity of IPs. Besides, different topologies of interconnection networks can provide different features in NoC. Among different topologies, the 2-D mesh is one of the most popular topologies, in which the routers of NoC are connected as a 2-D array and each IP is connected to an individual router. The mesh networks, the tree networks, the star networks and the simple loop networks, frequently appear in various applications of networks. Many interconnection networks are constructed by Cartesian product [13].

Definition 1 [13]. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. The Cartesian product of G_1 and G_2, denoted by $G_1 \times G_2$, is the graph with vertex set $V_1 \times V_2$ such that (u_1, v_1) is joined to $(u_2, v_2)k$ times if and only if either $u_1 = u_2$ and v_1 is joined to $v_2 k$ times in G_2 or $v_1 = v_2$ and u_1 is joined to $u_2 k$ times in G_1.

Definition 2 [13]. The topological structure of a 2-D mesh network $M(m_1, m_2)$ is defined as the Cartesian product of two paths $P_{m_1} \times P_{m_2}$, where P_{m_i} is the path graph with m_i vertices.

Due to its modularity and symmetry in terms of link length, the mesh topology is one of the most desirable topologies regarding to characteristics; the number of links per node does not change if additional nodes are added to the mesh. Therefore, the mesh topology offers a very good scalability. Additionally, because mesh network consists of fewer links per node compared to most other architectures, it has low cost.

2.2. Fault models

Fault-tolerance is defined as the ability of a system to continue operation despite presence of faults [14]. In this sense, fault-tolerance is closely related to concepts such as reliability, availability, and dependability, as it serves by providing these features.

Faults in a network take many forms, such as hardware faults, software bugs, or malicious sniffing or removal of packets. The first step in dealing with errors is to understand the nature of component failures and then to develop simple models that allow us to reason out the failures and the methods for handling them. Classification of faults by nature is either *random* or *systematic* faults. Random faults are usually hardware faults affecting the system components, which occur with a certain probability, while systematic faults such as software failures are faults which are not random, whether a component has it or not [14]. We assume that such permanent failures are detected and contained on a node or link boundary. Thus, faults are assumed to be fail-stop [15], meaning that we do not consider Byzantine (i.e., malicious) faults [14]. In the contexts of fault-tolerant routing, these are common assumptions [14–17].

Faults also can be classified by their duration as *transient* and *permanent* faults [14]. Transient faults persist in the system for only a short duration, while permanent faults remain in the system until it is repaired and may be either *dynamic* or *static*. In a dynamic fault model, once a new fault is found, actions are taken in order to appropriately handle the faulty component which allows the system to reconfigure at the hardware level, and preserves the original network topology. A static system provides a fault-tolerant routing algorithm that will bypass any failed nodes.

2.3. Fault patterns

To simplify the routing algorithm, adjacent faulty nodes are coalesced into *fault regions*, which may lead to different patterns of failed components. To analyze the performance of fault-tolerant routing algorithms, it is important to identify and quantify the fault regions, which may occur in the network. The shapes of such fault regions are often restricted by the fault model in use. Furthermore, the fault model may impose additional restrictions on the locations of the faults. For instance, faults may not be allowed on the edges of the mesh, and there may be a minimum distance between separate fault regions. A routing algorithm applying such a fault model is generally to tolerate all fault combinations conforming to the fault model, thus, the provided fault-tolerance is defined by the fault model. In the case that a fault combination is not conforming to the fault model in use, healthy nodes are marked as faulty (i.e., disabled) in order to create proper fault regions. Although healthy nodes are disabled, a fault combination is considered to be tolerated as long as all the nodes that are neither faulty nor disabled, are connected through valid paths. Fault regions extended by faulty components, may form *convex* (also referred to as the block fault model) or *concave* shaped fault patterns [14–18].

Definition 3 (*[14–18]*). A convex region is defined as a region $\mathcal{R}$ in which a line segment connecting any two points in $\mathcal{R}$lies entirely within $\mathcal{R}$. If we change the "line segment" in the standard convex region definition to "horizontal or vertical line segment", the resulted region is called rectilinear convex segments. Any region that is not convex is a concave region.