Microprocessors and Microsystems 38 (2014) 1037–1045

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Variability-tolerant routing algorithms for Networks-on-Chip

Eman Kamel Gawish [a], M. Watheq El-Kharashi [b,*], M.F. Abu-Elyazeed [a]

[a] Department of Electronics and Electrical Communications Engineering, Cairo University, Giza 12613, Egypt
[b] Department of Computer and Systems Engineering, Ain Shams University, Cairo 11517, Egypt

ARTICLE INFO

Article history:
Available online 1 September 2014

Keywords:
Networks-on-Chip (NoC)
Process variations
Routing algorithms

ABSTRACT

This paper proposes variability-tolerant routing algorithms for mesh-based Networks-on-Chip (NoC). Different NoC routing algorithms are modified, from variability perspective, to route flits through links with lower failure probability. The algorithms considered in this study are XY, West-First, Negative-First, and Odd–Even routing algorithms. To evaluate our variability-tolerant routing algorithms, a cycle-accurate simulator, NoCTweak, is used to measure how tolerant the resultant NoCs are against process variations. Results reflect the efficiency of our routing algorithms to overcome the process variation problems in modern fabrication technologies. For example, variability-tolerant West-First routing algorithm achieves up to 56% reduction in NoC overall failure rate.

1. Introduction

Networks-on-Chip (NoC) have appeared as good alternatives to global interconnects because of their optimized electrical properties, such as better performance in terms of power, delay, bandwidth, and scalability, compared to buses and global interconnects. Efficient NoC designs address the issues of performance, silicon area consumption, power/energy efficiency, reliability, and variability. These issues are the fundamental design drivers for an efficient NoC design [1].

The inability to precisely control the manufacturing process might result in unpredictable behavior of both device and wire, which in turn causes performance and power variations as well as an error-prone behavior. This becomes particularly important for modern fabrication technologies with feature sizes smaller than 65 nm. The reasons for higher variation effects at smaller feature sizes can be summarized as follows:

1. The process-resulting variations become comparable to the full length or width of the device.
2. The feature size approaches the fundamental dimensions, such as the size of atoms and the wave-length of the light, which are used for patterning lithography masks.

Process variations mainly result from front-end and back-end fabrication processes. The front-end fabrication processes are those involved in the fabrication of devices, whereas back-end processes are those involved in the fabrication of interconnects. Both the front-end and the back-end fabrication processes can have either random or systematic variability effects. Systematic variation effects have spatial correlation and usually arise from lithography, Chemical Mechanical Polishing (CMP), or etching fabrication steps. These effects cause systematic variations in gate length, threshold voltage, or Line Width Roughness (LWR). Random variability effects do not have any spatial correlation and are random in nature, like Random Dopant Fluctuation (RDF), Oxide Thickness Fluctuation (OTF), or Line Edge Roughness (LER) [2]. As technology scales down, identical NoC links encompass current and delay variations due to CMOS fabrication process variations causing, error at the link receiver, which we consider a link failure [3–5].

The paths the flits are routed through on an NoC are determined by the used routing algorithm. Different paths go through different NoC links having different link delay variations, which results in different link failure probabilities. In this paper, we consider the average link failure probability that flits go through for a certain traffic pattern as an indicator of how certain NoC routing algorithms are prone to process-induced delay variations.

Routing algorithms can be classified into two types: deterministic and adaptive. In deterministic routing, a path is completely determined by its source and destination addresses. On the other hand, a routing technique is called adaptive if, given a source and a destination addresses; the path taken by a particular flit depends on dynamic network conditions (e.g., congested links due to traffic variability and minimum length to destination) [6]. In this paper, we work with one of the main deterministic routing algorithms (ordered XY routing) and three adaptive routing algorithms

* Corresponding author.
E-mail address: watheq.elkharashi@eng.asu.edu.eg (M.W. El-Kharashi).

http://dx.doi.org/10.1016/j.micpro.2014.08.002
0141-9331/

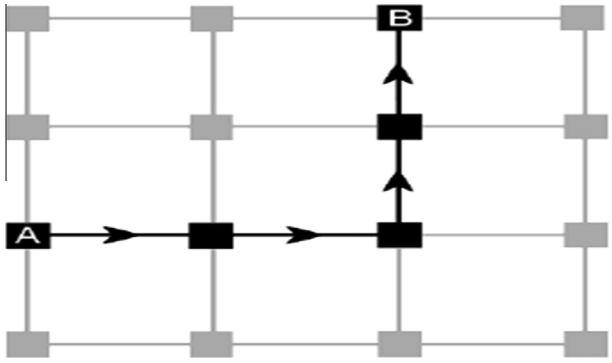

Fig. 1. XY routing from router A to router B [7].

(West-First, Negative-First, and Odd–Even minimal routing). These routing algorithms are briefed as follows:

- *XY routing*: XY routing is a dimension-order routing, which routes flits first in *x*, or horizontal direction, to the correct column and then in y, or vertical direction, to the receiver, as shown in Fig. 1. Addresses of the routers are their *XY*-coordinates [7].
- *West-First routing*: West-First routing algorithm prevents all turns to the west, so the flits going to the west must be first transmitted as far to the west as necessary. Routing flits to the west is not possible later [8]. Allowed turns in West-First routing are shown in Fig. 2(a).
- *Negative-First routing*: Negative-First routing algorithm allows all turns except turns from the positive direction to the negative direction. Flit routing to the negative directions must be done before anything else [9]. Allowed turns in Negative-First routing are shown in Fig. 2(b).
- *Odd–Even routing*: Odd–Even routing is a deadlock-free turn model, which prohibits turns from the east to the north and from the east to the south at tiles located in even columns and turns from the north to the west and the south to the west at tiles located in odd columns [10].

In this paper, we add the link failure probability to the adaptive conditions to be considered at routing (in addition to flit length, buffer size, and nearer dimension output port). We use an open source cycle accurate NoC simulator, NoCTweak [11], to simulate different traffic patterns with modified NoC routing algorithms.

The main contributions of this work are as follows:

1. Modeling, on the system level, the NoC link failure resulting from random and systematic process variations at certain technology node and mesh size.
2. Modifying XY, West-First, Negative-First, and Odd–Even routing algorithms in NoCTweak to consider link failure probability when routing, to obtain variability-tolerant routing algorithms.
3. Proposing the NoC failure rate as a measure of tolerance against process variations under certain NoC technology node, buffer size, injection rate, mesh size, traffic pattern, and routing algorithm.

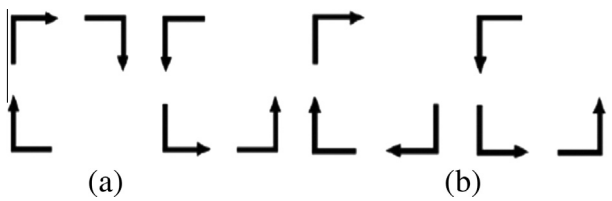

Fig. 2. Allowed turns in: (a) West-First routing and (b) Negative-First routing [8].

The rest of this paper is organized as follows. Related work is discussed in Section 2. Section 3 presents our NoC failure model. Section 4 shows the proposed variability-tolerant routing. Section 5 describes NoCTweak and analyzes the simulation results. We draw conclusions and give ideas for future work in Section 6.

2. Related work

There have been several work in the literature addressing process variations effects on NoC. Initially variability effects on NoC routers were addressed [1,12,13]. As technology scales down, interconnect delay dominates gate delay. Hence, research considered variability effects on NoC links [2–5]. Subsequently, research aimed for fault-tolerant routing algorithms [14–23].

Nicopoulos et al. presented the first comprehensive evaluation of NoC susceptibility to process variability effects and proposed an array of architectural improvements in the form of a new router design to increase resiliency to these effects [1]. By process variation exploration, Nicopoulos et al. identified the contribution of each major router stage to the overall critical path delay. The contribution to delay was used to guide the proposed modifications to improve process variation resilience without adversely affecting performance.

Sivaswamy and Bazargan tolerated variations by a variation-aware router that was optimized according to statistical critical delay path [12]. They also proposed a modification to the clock network to deliver programmable skews to different flip-flops, tolerating variations within the clock paths.

Konstantinos et al. proposed a circuit-level fault modeling tool to capture run-time process-induced random delay variations and their corresponding system-level faults. The tool points out to the router components that need resilient design [13].

Mehranzadeh and Hoodgar presented a fault-aware routing algorithm scheme called FAXY based on XY routing algorithm [6]. With FAXY routing, a flit first traverses along the *X* direction and then along the *Y* direction. When a flit traverses along the *X* direction and a link is masked due to a permanent fault, it traverses along the *Y* direction in order to increase the overall network throughput and prevent flit losses.

Wu et al. proposed an improved routing algorithm which tolerates a single link fault in 2D mesh NoC [14]. Their algorithm is deadlock-free, yet is subject to flit loss.

Ebrahimi et al. proposed deadlock-free fully adaptive routing algorithm using virtual channels along the *X* and *Y* directions [15]. Ebrahimi proposed fault-tolerant routing tolerating the fault probability resulting from random variations only. In this paper, we propose a set of variability-tolerant routing algorithms that consider link failure probability resulting from both systematic and random variations.

The turn model is originated from Glass and Ni work in [16]. They introduced three adaptive routing algorithms: West-First, North-Last, and Negative-First. These routing algorithms eliminate deadlocks without adding virtual channels by prohibiting some global turns. The turn model also results in routing algorithms that are ideal for fault tolerance, live-lock free, and highly adaptive [16–18]. Glass and Ni also proposed a turn-based fault-tolerant routing algorithm in [17] that is based on modification of the Negative-First routing algorithm. This algorithm can deal with any one-faulty-router topology. In this proposal, each routing function depends on the coordinates (*Y*, *X*) of the router, the packet destination, the input channels, and the size of the mesh.

The Position-Route method proposed in [19] is a deterministic routing algorithm. When the destination is to the west, packets are first sent in the west direction up to the column of the destination, and then, sent to either north or south. Otherwise, packets are