

# Accelerated assessment of fine-grain AVF in NoC using a Multi-Cell Upsets considered fault injection



Jiajia Jiao<sup>a,\*</sup>, Yuzhuo Fu<sup>a</sup>, Shijie Wen<sup>b</sup>

<sup>a</sup> School of Micro Electronics, Shanghai Jiao Tong University, China  
<sup>b</sup> Cisco Research Center, USA

## ARTICLE INFO

### Article history:

Received 10 February 2014

Received in revised form 11 June 2014

Accepted 16 June 2014

Available online 14 July 2014

### Keywords:

AVF assessment

NoC

Soft error

MCU

Acceleration

Fault injection

## ABSTRACT

With the increasing threat of soft errors induced bits upset, Network on Chip (NoC) as the communication infrastructure in many-core systems has been proven a reliability bottleneck in a fault tolerant parallel system. The often-used metric Architecture Vulnerability Factor (AVF), measures the architecture-level soft error impacts to compromise the design cost of fault tolerant schemes and reliability well. As a complementary of existing estimation methods about standard IP like processor and Cache, this work aims at an accelerated fault injection methodology for the fine-grain AVF assessment in NoC via two components: (1) modeling the complex fault patterns of both Multi-Cell Upsets (MCU) and Single Bit Upset (SBU) in the standard Fault Injection (FI) method; (2) accelerating the estimation via classifying and exploiting the fine-grain metrics according to different error impacts. The comprehensive simulation results using the diverse configures (e.g., varying fault model, benchmark, traffic load, network size and fault list size) also demonstrate that the proposed approach (i) shrinks the estimation inaccuracy due to MCU patterns 18.89% underestimation in no protection case and 88.92% overestimation under ECC (Error Correction Coding) protection on average; (ii) achieves about 5× speedup without estimation accuracy loss via phased pre-analysis based on fine-grain classification; (iii) verifies ECC a cost-effective mechanism to protect NoC router: soft errors reduced by about 50% over the no protection case, with only less than 2% area overhead.

© 2014 Elsevier Ltd. All rights reserved.

## 1. Introduction

Soft errors, also known as transient faults or single-event upset, are caused by external radiation or electrical noise. With the scaling technology, lower supply voltages and higher integration density, the soft error has becoming a challenge for the reliable design and mainly embodies: (1) the soft error rate increases exponentially as the technology node scales down [1]; (2) the more and more multi-bit upset cannot be ignored any longer, as Fig. 1 shows the statistics of the MCU ratio increasing dramatically to 40% after 45 nm technology node. The two challenges thereby bring more difficulties for the works on soft error mitigation, estimation and so on.

To effectively tradeoff the design cost (e.g., area) with reliability, the accurate and efficient estimation methods are required to characterize the soft error impacts at an early design stage [2]. Such estimates, conventionally, are used to identify the components

with a high vulnerability to soft errors, and thereby system designers can effectively deploy mitigation strategies to minimize the impacts brought by soft errors without introducing much design overhead. Furthermore, efficient estimation approaches can also reduce the time to market and support the online fault-tolerant design very well. Therefore, the accurate and efficient estimation of soft error impacts is significant for a reliable design.

In the meantime, the scaling technology also results in Network on Chip (NoC) to be the novel communication infrastructure instead of traditional bus or crossbar for its low latency, good scalability and high bandwidth. However, the soft error is threatening the NoC reliability. Especially, the report of Intel 80-cores shows that a 5 GHz router takes up 17% area in each tile except the wires [12]. Namely, the soft error hit rate in NoCs is a high possibility, and could not be neglected from the area consumption perspective. On the other hand, Bernhard Fechner et al considered the dependencies of function modules in the parallel systems like many-core, FPGA as well as GPU, etc. and proposed one theoretical method to analyze the bottleneck of the whole fault tolerant parallel system in [13]. Taking FPGA for example, the results of proposed model exemplarily showed the switchboxes was the main reliability

\* Corresponding author. Tel.: +86 21 34204546 1124.

E-mail addresses: [jiaojiajia@ic.sjtu.edu.cn](mailto:jiaojiajia@ic.sjtu.edu.cn) (J. Jiao), [fuyuzhuo@ic.sjtu.edu.cn](mailto:fuyuzhuo@ic.sjtu.edu.cn) (Y. Fu), [shwen@cisco.com](mailto:shwen@cisco.com) (S. Wen).

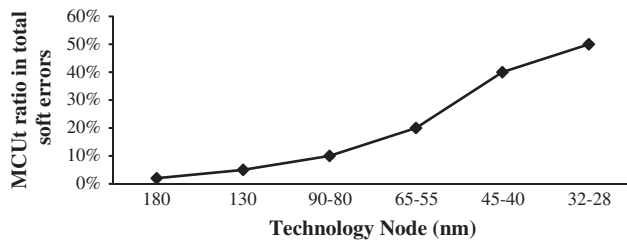


Fig. 1. MCU total ratio variation with scaling IC technology from Cisco Corp.

bottleneck. That's to say, the reliability of communication components is critical for a whole parallel system. Therefore, it is necessary to pay more attention to the soft error challenges in NoCs.

Resilient NoCs [3,4,17,18] proposed novel soft error mitigation methods to guarantee its communication quality against the soft errors. However, despite of the demonstrated effectiveness of these solutions, applying them blindly across an entire design would result in prohibitive cost. Therefore, it is vital to use effective techniques to characterize the soft error impacts in NoCs at the early stage of design. Namely, we should estimate the complete masking factors such as logical, functional and application-aware to avoid overestimation.

However, the existing works on estimation still face some open problems: (1) pure SBU pattern assumption. The related works [5–9] to reliable NoCs all used Single Bit Upset (SBU) pattern to estimate. It is assumed that only one single bit hitting the system in random time and random location. Actually, with the scaling technology, the cell size as well as critical charge is getting smaller and the MCU cases dominate in soft error occurrences. Especially, our previous work injects some SBU faults independently (temporal Multi-Bit Upsets) [8] while the realistic and spatial MCU of the same soft error should be modeled; (2) low efficiency because of time consuming fault injection. Similar to the fault injection works on other components like core or memory [15,16], fault injection in NoCs keeps low estimation speed for a larger number of long simulations [9]. As a complementary work to these above works [5–9,15,16], the paper aims at MCU (Multi-Cell Upsets) considered Estimation framework based on Pre-analysis for Accelerated (MEPA) AVF assessment in NoCs based on FI from the following contributions:

- *Modeling the complete soft error patterns (MCU and SBU) using fault injection.* Remove the pure SBU assumption and model all the possible soft error patterns including both MCU and SBU. So that the evaluation based on fault injection is accurate enough.
- *Accelerating the estimation via phased pre-analysis.* Considering the general predictability of soft errors and special predictability of NoC architecture, we classify and exploit the fine-grain

metrics according to different error impacts. Such a way could be used to interrupt or cancel some predictable simulations. Thereby, the estimation time is saved with perfect accuracy.

- *Comprehensive evaluation of proposed method.* The comprehensive simulation results using the diverse configures (e.g., varying fault model, benchmark, traffic load, network size and fault list size) also demonstrate that the proposed approach is accurate (remove the estimation inaccuracy due to MCU patterns 18.89% underestimation in no protection case and 88.92% overestimation under ECC) and efficient (about 5× speedup).

The paper is organized as follows. Section 2 introduces the basic architecture. Section 3 presents problem formulation and the proposed method MEPA is detailed in Section 4. Section 5 provides the implementation while the simulation results and analysis are described in Section 6. Finally Section 7 concludes the paper.

## 2. Basic architecture

In this section, we introduce the basic NoC architecture (Section 2.1), and basic fault injection methodology (Section 2.2).

### 2.1. NoC architecture

Like the mainstream academic research and industry products (e.g., Intel's SCC, Tilera's Tile64/100) of many-core processor, we choose a 2D Mesh topology, XY routing algorithm, wormhole switching mechanism and Virtual Channel (VC) configuration as the basic NoC micro-architecture. In details, a five-port router in the Mesh center is shown in Fig. 2. Note that each dash-dotted box is duplicated five times in a complete router. Each input port in a router consists of VCs, a RC unit, a VCA, while an output port includes output registers and a controller. Each VC also has its own status registers (named VC.RC, VC.VCA and VC utilization) to store the results of RC, VCA allocation, and VC utilization state. OC (Output Controller) integrates the function of switch allocation (SA). RC and VCA are only designed for the head flit, and thus the flit type is a key point to make a decision. RC extracts the flit type information of a coming flit in VCs and determines the next-hop output port based on the corresponding destination address. Similarly, the VCA grants an available VC in the next hop from available CREDIT information. Unlike VCA, SA is used by all the flits to grant the switch. Then, the flit in an output register is transmitted to the next router through an inter-router link. Finally, repeat the above transmission process hop by hop until flits arrive at their destinations.

Similar to the NoC designs of TILE64 and SCC, the formats of a packet and a flit are given in Fig. 3(a)–(c) respectively. Each packet is composed of one head flit, one tail flit and several data flits. The head flit carries control information like a source ("src") and a

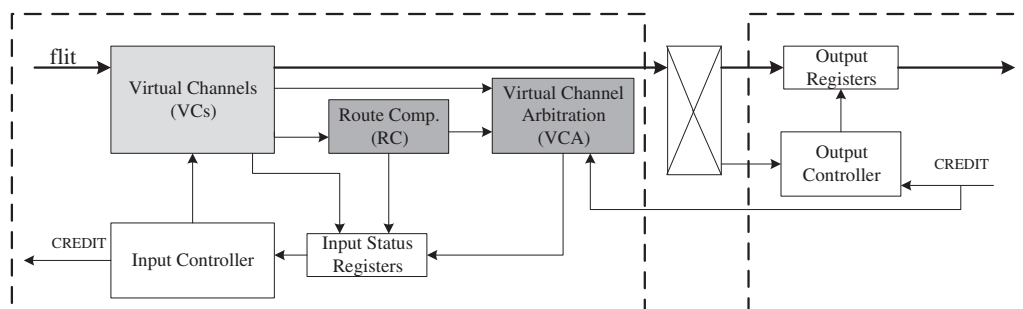


Fig. 2. Router architecture.

Download English Version:

<https://daneshyari.com/en/article/546817>

Download Persian Version:

<https://daneshyari.com/article/546817>

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