Microelectronics Reliability 55 (2015) 251–263

Contents lists available at ScienceDirect

Microelectronics Reliability

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

Exploiting component dependency for accurate and efficient soft error analysis via Probabilistic Graphical Models ☆

Jiajia Jiao [a,*], Da-Cheng Juan [b], Diana Marculescu [b], Yuzhuo Fu [a]

[a] School of Microelectronics, Shanghai Jiao Tong University, China
[b] Department of Electrical & Computer Engineering, Carnegie Mellon University, United States

ARTICLE INFO

Article history:
Received 29 April 2014
Received in revised form 29 July 2014
Accepted 15 September 2014
Available online 19 October 2014

Keywords:
Soft error
ACE
AVF
Logical masking effects
Probabilistic Graphical Models

ABSTRACT

As the technology node continues to scale, soft errors have become a major issue for reliable processor designs. In this paper, we propose a framework that accurately and efficiently estimates the Architectural Vulnerability Factor (AVF) of critical storage structures of a processor. The proposed approach exploits the masking effects between array structure (e.g., register files and Caches) and logic units (e.g., Int-ALU) via the unified Probabilistic Graphical Models (PGM) methodology, and can provide *guaranteed* AVFs by two accuracy–efficiency tradeoff solutions. The experimental results have confirmed that, compared to current state-of-the-art approaches, the proposed framework achieves accurate and efficient estimation via two instanced solutions: (1) first-order masking effects up to 45.96% and on average 8.48% accuracy improvement with 52.01× speedup; (2) high-order masking effects average 87.28% accuracy improvement with 43.87× speedup. The two different accuracy–efficiency tradeoff of proposed MEA-PGM can be applied into different estimation scenarios (e.g., short time to market of general mobile devices and high reliable requirements in aerospace platforms) in flexibility.

1. Introduction

Soft errors, also known as transient faults or single-event upset, are caused by external radiation or electrical noise. Due to lower supply voltages, higher integration density, and other factors, the soft error rate dramatically increases as the technology node scales down [1]. To effectively tradeoff the design cost (e.g., area) with higher reliability, accurate and efficient estimation of soft error impacts is required at an early design stage. 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.

Accurate and efficient estimation of soft error impact is required at an early design stage to effectively tradeoff the design cost (e.g., area) with reliability. Existing work on soft error estimation often uses: (1) Fault Injection (FI) to guarantee accuracy [2–4,25]; (2) Fault free based analytical models to improve speed [5–8]. The former is too time consuming while the latter only provide the over-pessimistic value. How to achieve accurate and efficient estimation is still an open problem.

In this paper, we propose a new framework; Masking-Effect-Aware analysis cooperated with Probabilistic-Graphical-Models (MEA-PGM) methodology to estimate the soft error impacts on a processor. To the best of our knowledge, the proposed MEA-PGM brings the following contributions:

- *Masking-effect exploration via PGM.* We explore and exploit the masking effects generated from component dependencies via PGM. The masking effects used to significantly reduce the "false-positive" soft-error rate, are characterized by the effective PGM methodology.
- *Flexible accuracy–efficiency tradeoff.* Based on the masking effects discovered via PGM concept, MEA-PGE provides guaranteed AVF estimations efficiently via two instanced PGM implementations respectively: (1) 8.48% accuracy improvement and 52.01× speedup for the short time-to-market of general processor design; (2) 87.28% accuracy improvement and 43.87× speedup for high reliable requirements of aerospace applications.
- *Comprehensive evaluation.* To conduct a comprehensive evaluation, we compare the proposed MEA-PGM with three commonly-used models by both industry and academia [6,8,9].

☆ This work was supported by Innovation Program of Shanghai Municipal Education Commission, Numbered 14ZZ018.

* Corresponding author at: No. 800 Dongchuan Road, Minhang District, Shanghai, 200240, China. Tel.: +86 13671828752.

E-mail addresses: jiaojiajia@ic.sjtu.edu.cn (J. Jiao), dacheng@cmu.edu (D.-C. Juan), dianam@cmu.edu (D. Marculescu), fuyuzhuo@ic.sjtu.edu.cn (Y. Fu).

http://dx.doi.org/10.1016/j.microrel.2014.09.011
0026-2714/

The experimental results confirm the proposed MEA-PGM is cost-effective: 8.48–87.28% higher accuracy over optimal ACE methods and 43.87–52.01× speedup over FI method.

The remainder of this paper is organized as follows. Section 2 provides the necessary related work review. Section 3 gives the detailed problem formulation and Section 4 details the proposed MEA-PGM. Section 5 shows the implementation flow, while Section 6 provides the experimental results. Section 7 concludes the paper and points to future work. Section 8 serves as the appendix that provides the Supplementary materials.

2. Related work

The section depicts the related work of soft error estimation: (1) AVF + SoFA methodology overview in Section 2.1; (2) accurate Fault injection versus. Fast Fault Free Analysis for AVF estimation in Section 2.2; (3) existing works on ACE methods; (4) works review of masking effects estimation.

2.1. AVF + SoFA methodology

AVF + SoFA (Sum of Failure All) methodology measures the soft error impacts. The metric *FIT* (Failure in Time, the number of errors during 10^9 h) is calculated by two steps: (1) estimate Architectural Vulnerability Factor (AVF), which represents the probability that a single-bit upset results in a user-visible error in the final output at architecture-level [5]; (2) sum up *FIT* of all components while the *i*th component FIT_i can be calculated by the product of AVF and FIT_{raw} in Eq. (1) [33], where FIT_{raw} is the inherent FIT due to the joint effects of physical environment, device and circuit designs.

$$FIT_i = AVF_i \times FIT_{raw\_i} \quad (1)$$

AVF estimation is very critical to calculate the final *FIT*. And thereby our focus is accurate and fast AVF estimation.

2.2. Fault injection versus. Fault free ACE analysis

The existing works to address the AVF estimation focus on two aspects: accuracy and speed.

Accurate fault injection. Soft error estimation uses FI [2–4,25] to calculate AVF in Eq. (2),

$$AVF(FI) = \frac{N_{err}}{N_{total}} \quad (2)$$

where N_{err} denotes the number of simulations with an observed fault and N_{total} represents the total number of simulations. The accuracy of AVF estimated by FI is directly related to N_{total}. In other words, a large number of simulations will be required for an accurate AVF, which is very inefficient (usually up to days) though Maniatakos et al. uses selective policy for about up to18× speedup [25].

Fast fault free analysis. The outstanding representative of fault free analysis is the very popular and simple ACE (Architectural Correct Execution) analysis. The ACE method provides an alternative to estimate AVF by using only one (or at most two) simulation runs. The idea is to exploit the structure of a typical processor by analyzing the cases, where some single bit faults will not produce an error in a program's output. This method measures in cycles each ACE piece (a critical time period which will affect the architectural state or application output, during which the final output can be affected by an event upset in). Instead, an un-ACE piece is not harmful in Fig. 1. Based on this concept, the AVF of a structure with a bit width of *N* can be expressed as:

$$AVF(ACE) = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{ACE\ cycles\ for\ bit\ i}{Total\ cycles}\right) \quad (3)$$

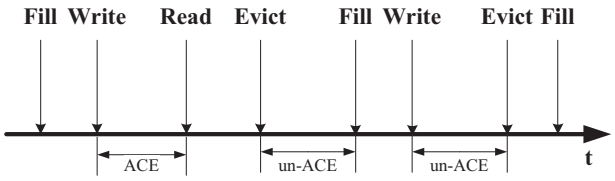

Fig. 1. ACE piece for storage structure based on access types.

Compared to FI, ACE is faster and more suitable for early design stage exploration. In the context of the complex multi-core/many-core architectures, simulation time and number of components that needs to be considered increase for each design generation, and therefore, ACE is a preferred method in both industry and academia [7–9] than FI. However, these ACE methods need to be conservative, and leads to a pessimistic estimation. The AVF estimated by ACE is 2–3× higher than the AVF estimated by FI [2]. Note that overestimating AVF directly leads to excessive area or performance overhead due to overprotection; for example, it has been shown that merely 7.74% overestimation may cause up to 40.39% area overhead [22]. In this paper, we focus on the accuracy improvement of ACE while still keeping the high speed of ACE estimation.

2.3. Related work on ACE analysis

Prior work on ACE addresses two aspects: accuracy and speed.

Wang et al. [2] and George et al. [4] both pointed that AVFs computed using ACE analysis method overestimate the result of soft error rate by 2×–3× in many fault injection experiments. To overcome the conservatism of the method, fine grain ACE via adding more simulation details is used in [7,8] to reduce the gap. For example, the branch instruction does not need one destination register, and thus the field of destination address in ROB does not include ACE bits. In a different work [9] FI is mixed with ACE to compute the AVF of the Cache structure more accurately.

Other related work focuses on ACE estimation speed. The authors use machine learning approaches [10,11] or a mechanics model [12] to simplify ACE analysis and achieve faster AVF prediction. All these methods assume AVF of ACE analysis itself is accurate and aim at inexpensive AVF estimation.

Accurate ACE analysis not only minimizes the design overhead of mitigation schemes, but also provides reliable training data for machine learning based prediction. Therefore, more accurate ACE analysis is our goal in this paper.

2.4. Related work on masking effects estimation

Generally speaking, masking effects fall in three categories: (i) electrical masking; (ii) latching-window masking, and (iii) logical masking. So far, several effective methods incorporating electrical masking and latching-window masking have been proposed like SEAT-LA [18] and CEP [19]. However, logical masking effects where an error occurs in a logic unit or a storage element but has no effect on the architectural state or application output, is usually captured in the FI method [20].

For ACE analysis, only a limited set of methods [8,9] have considered the logic masking effects in AVF estimation. Fu et al. [8] used the extreme cases of AND/OR instruction (e.g., one operand is zero) and NOP instructions. Haghdoost et al. [9] combined the ACE analysis with the average masking rate computed by FI to determine a more accurate AVF for Cache structures. Besides the two system-level masking effects characterization, some works on instruction-level masking effects also are estimated in [27] via an analytical model and he metric of PVF (Program Vulnerability Factor) is defined in [28] to evaluate the soft error impacts from