



ELSEVIER

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: [www.elsevier.com/locate/mejo](http://www.elsevier.com/locate/mejo)

# A two-level approximate model driven framework for characterizing Multi-Cell Upsets impacts on processors



Jiajia Jiao<sup>a</sup>, Diana Marculescu<sup>b</sup>, Da-Cheng Juan<sup>c</sup>, Yuzhuo Fu<sup>d</sup>

<sup>a</sup> Shanghai Maritime University, China

<sup>b</sup> Carnegie Mellon University, United States

<sup>c</sup> Google, United States

<sup>d</sup> Shanghai Jiao Tong University, United States

## ARTICLE INFO

### Article history:

Received 30 March 2015

Received in revised form

6 October 2015

Accepted 25 November 2015

Available online 17 December 2015

### Keywords:

Soft error analysis

Multi-Cell Upsets

Probabilistic Graphical Model

Boundary model

## ABSTRACT

Soft error analysis is very significant for a good tradeoff between processor design cost (e.g. area and power) and reliability. In this paper, we propose an approximate model driven framework for efficient soft error analysis in processors. The proposed framework includes: 1) an approximate Probabilistic Graphical Model (PGM) for the Single Bit Upset (SBU) estimation, uses average-and-max policy to handle the mapped PGM structure, node parameter and inference fast; 2) an approximate boundary model for the more complex Multi-Cell Upsets (MCU) case, adopts relax-and-strict way to reuse the approximate PGM model and characterize MCU patterns completely. The comprehensive results confirm that, compared with the state-of-the-art, the proposed two-level methodology based on approximate models achieves fast estimation up to more  $15.37 \times$  speedup while only 8.14% accuracy loss on average. Furthermore, the complex MCU impacts are also estimated by the proposed method at the same order of magnitude as the runtime of the simple SBU case.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

Soft errors, also called transient faults or single-event upsets, are typically caused by external radiation or internal electrical noise. With increased technology scaling, soft errors have become a challenge for reliable design since (1) soft error rate increases exponentially [1]; and (2) Multi-Cell Upsets (MCU) ratio has increased to 40% since 65 nm [2,3]. These make soft error estimation more difficult.

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 MCU-induced soft error estimation often uses: (1) Fault Injection (FI) to guarantee accuracy [4,5]; (2) Fault free based methods to improve estimation speed [6,7]. The former is too time consuming while the latter only focuses on some specified components, like L2Cache. How to achieve efficient estimation of MCU with guaranteed accuracy is still an open problem.

In this paper, we propose a two-level approximate model for much more efficient MCU estimation in a processor. The key point is exploiting the small inexactness of approximation for a large of estimation speed benefits. Especially such a framework can be applicable for the inexactness tolerance of some applications like stream processing [11], as well as the online reconfigurable architecture [28]. The main contributions include:

- Constructing an approximately mapped PGM model for SBU estimation by modifying the Bayesian structure and node parameter in a subtle way.
- Decomposing complex MCU issue into SBUs via a histogram analysis based boundary model for the approximate estimation.
- Proposing a unified two-level framework for the approximate MCU estimation in processors and evaluate its effectiveness based on comprehensive simulation results, up to more  $15.37 \times$  speedup while only 8.17% accuracy loss.

The remainder of this paper is organized as follows. Section 2 provides the related work. Section 3 describes the detailed problem formulation and Section 4 details the proposed approximate framework. Section 5 shows the implementation while Section 6

E-mail addresses: [jiaojiajia@shmtu.edu.cn](mailto:jiaojiajia@shmtu.edu.cn) (J. Jiao), [dianam@cmu.edu.cn](mailto:dianam@cmu.edu.cn) (D. Marculescu), [ruan.dave@gmail.com](mailto:ruan.dave@gmail.com) (D.-C. Juan), [fuyuzhuo@ic.sjtu.edu.cn](mailto:fuyuzhuo@ic.sjtu.edu.cn) (Y. Fu).

gives the experimental results. Finally, Section 7 concludes the paper.

## 2. Related work

Like the Single Bit Upset (SBU) estimation, the metric *FIT* (Failure in Time, the number of errors during  $10^9$  h) is calculated by two steps: 1) estimating Architectural Vulnerability Factor (*AVF*), which represents the probability that a soft error induced upset results in a user-visible error in the final output at architecture-level [8]; 2) summing 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), where  $FIT_{raw}$  is the inherent *FIT* due to the joint effects of physical environment, device and circuit designs.

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

The *AVF* estimation is very critical to calculate the final *FIT*. And thereby our focus is also the accurate and fast *AVF* estimation. The existing works to address the *AVF* estimation of MCUs focus on two points: accuracy and speed.

### 2.1. Accuracy

Like SBU estimation [4,5], MCU uses FI [3,12] to calculate *AVF* in Eq. (2), where  $N_{err}$  denotes the number of simulations with an observed fault and  $N_{total}$  represents the total number of simulations. A large number of simulations will be required for an accurate *AVF* value, which makes inefficient (usually up to days) estimation though Maniatakos et al. uses selective policy for about up to  $18 \times$  speedup [12].

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

### 2.2. Speed

The fault free analysis uses one simulation to extract the necessary information and analyze the soft error impacts fast. The typical representatives are the ACE (Architecturally Correct Execution) analysis [8,9], the Markova model based reliability analysis and probabilistic theory based estimation methods [13]. The ACE analysis is more popular due to its good scalability. It 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. The *AVF* of a structure with a bit width of  $N$  can be expressed as [8,9]:

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

Therefore, ACE is faster and more suitable for early design stage exploration. Its accuracy can be improved by exploiting the error masking effects [10,14,15], and their estimation speed can be accelerated via machine learning based prediction [16] or a first-order mechanical model [17]. However, existing ACE works [10,14–17] are only applicable for SBU case. To achieve efficient MCU estimation, the new estimation framework is required to extend the MCU case. Furthermore, the inexactness tolerance of stream applications and the small probability of worst cases occurring for random soft errors, are both encouraging a novel method to exploits the possible “small accuracy loss for large speed improvement”.

In this paper, we propose an approximate model driven framework to achieve MCU estimation fast and accurately. This approach not only decomposes the MCU into SBU problem well,

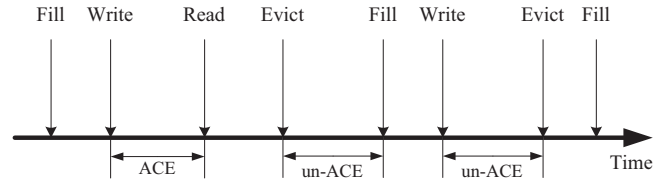


Fig. 1. ACE computation policy for tracing the storage structure.

but also explores an approximate PGM model for efficient SBU estimation.

## 3. Problem formulation

This section describes the target object, fault model, basic PGM model and detailed problem formulation respectively.

### 3.1. Target object

In this paper, we just focus on the soft error analysis of storage structures. The reasons are twofold: (1) storage structures dominate in overall soft error impact. Control-logic/data-path, has inherently less impact on overall reliability [18]. (2) Control-logic/data-path estimation at architecture-level has been addressed well [8], while the remaining challenges for control-logic/data-path are at circuit-level [19], beyond the scope of this paper.

Here, we take the register files as a case study: (1) since the register files are frequently-accessed and extremely vulnerable to soft errors [20,21]. (2) Furthermore, not like L2 Caches, the soft errors in register files will not be corrected by the expensive Error Correction Coding (ECC) [22,23]. (3) Register file is the unique entry to the logical unit ALU. The accurate *AVF* of other storage structures like L1 data Cache also depends on the masking rate of register file. And the proposed method can also be generalized to all array structures easily (e.g., L1 data Cache).

### 3.2. MCU fault model

The soft error distribution includes the error types (SBU and MCUs), the MCU physical map patterns, and their associated distribution probabilities. In this paper, we use the fault model in Fig. 2 as our basic MCU considered model.

The maximum number of bits upset per event is nine and the entire set of soft error types is {SBU, MCU\_2 ..... MCU\_9}. The probability of SBU occurrence is 52% and the remaining MCU patterns make up the remaining 48%.

Based on the result of Pareto analysis, MCU\_2 bits and MCU\_3 bits dominate in MCU patterns, up to 90%. The physical MCU information for MCU\_2 and MCU\_3 is available in Fig. 2(b). However, the error patterns of MCU\_4 to MCU\_9 are unknown. Therefore, we assume their error patterns can be determined by randomly extending the fault region of MCU\_3 as shown in Fig. 3. The central case (Fig. 3(a)) and the bound case from (Fig. 3(b)) are extended to MCU\_5 (Fig. 3(c)) and MCU\_6 (Fig. 3(d)) respectively. The above assumption is reasonable for considering the random feature of soft errors and the locality of a single MCU occurrence. Moreover, the occurrence probability of MCU\_4 to MCU\_9 is relatively low, less than 5% of overall bit upset. Therefore, the assumed physical distribution is not likely to introduce high accuracy loss.

### 3.3. Basic PGM model for SBU estimation

In our previous work [10], the PGM is introduced for the SBU estimation. Its main idea is to map the soft error issue into a PGM

Download English Version:

<https://daneshyari.com/en/article/546905>

Download Persian Version:

<https://daneshyari.com/article/546905>

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