Microprocessors and Microsystems 39 (2015) 1204–1214

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Cross-layer reliability evaluation, moving from the hardware architecture to the system level: A CLERECO EU project overview

A. Vallero [a], S. Tselonis [b], N. Foutris [b], M. Kaliorakis [b], M. Kooli [c], A. Savino [a], G. Politano [a], A. Bosio [c], G. Di Natale [c], D. Gizopoulos [b], S. Di Carlo [a,*]

[a] Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy
[b] Department of Informatics and Telecommunications, University of Athens, Athens, Greece
[c] LIRMM (Université Montpellier/CNRS UMR 5506), Montpellier, France

ARTICLE INFO

Article history:
Received 23 December 2014
Revised 11 May 2015
Accepted 4 June 2015
Available online 20 June 2015

Keywords:
Reliability evaluation
Fault injection
Statistical models

ABSTRACT

Advanced computing systems realized in forthcoming technologies hold the promise of a significant increase of computational capabilities. However, the same path that is leading technologies toward these remarkable achievements is also making electronic devices increasingly unreliable. Developing new methods to evaluate the reliability of these systems in an early design stage has the potential to save costs, produce optimized designs and have a positive impact on the product time-to-market.

CLERECO European FP7 research project addresses early reliability evaluation with a cross-layer approach across different computing disciplines, across computing system layers and across computing market segments. The fundamental objective of the project is to investigate in depth a methodology to assess system reliability early in the design cycle of the future systems of the emerging computing continuum. This paper presents a general overview of the CLERECO project focusing on the main tools and models that are being developed that could be of interest for the research community and engineering practice.

1. Introduction

Most things we rely on in our everyday life contain electronic-based information and have enough computing power to run embedded software applications, which connect to the Internet and remote advanced computing services to get access to virtually unlimited resources. This future computing continuum, composed of a wide set of heterogeneous platforms, promises to be a fertile environment to engineer advanced services with high added value.

Radiation effects, wear-out, aging and variability throughout the operational period of a system, extreme scaling processes that move towards 12 nm manufacturing process nodes and beyond, the high design complexity, and a fast time-to-market demand are expected to make system components extremely unreliable. As an example, the single bit error rate of a six-transistor SRAM that is in the order 1.5×10^{-6} for a 22 nm technology is expected to increase up to 5.5×10^{-5} in 16 nm technology and 2.6×10^{-4} in 12 nm technology [1].

From a reliability perspective, system designers have to meet precise reliability requirements. These requirements are highly domain dependent and are influenced by the criticality of the considered system or component (e.g., aerospace and medical applications require very low failure rates). Reliability is therefore increasingly driving several design decisions at the technology, hardware and software level.

Error management solutions at all design/implementation levels are feasible. Technology can be hardened [2–8], hardware architectures may include redundancy [9–19], and finally all software layers may implement error detection and recovery mechanisms [20–26]. On the one hand, this enables designers to apply cross-layer holistic design approaches to manage errors in their systems. On the other hand, this enlarges the design space making design optimization complex.

Nowadays, the dominant approach to design reliable systems consists in worst-case design. However, it is well known that several reliability-oriented design decisions lead to costs in terms of area, complexity, performance and energy budget [27]. Reliability engineers and system architects need to be provided with adequate tools to cope with this complexity and to take design decision able to enable reliability targets to be met with minimum cost.

* Corresponding author.Tel.:+39 011 090 7080; fax:+39 011 090 7099.
E-mail address: stefano.dicarlo@polito.it (S. Di Carlo).

http://dx.doi.org/10.1016/j.micpro.2015.06.003
0141-9331/

Moreover, these decision must be taken as early as possible in the design process when redesign and optimizations are still affordable. Products failing to reach the reliability objectives in the late stage of the design may lead to commercial failure with severe economical consequences.

Current reliability analysis approaches strongly rely on massive and time-consuming RTL fault injection campaigns, which are becoming a bottleneck due the increasing complexity of computing systems. Simulating a complete system composed of microprocessors and accelerators embedding several tens of processing cores and memory blocks, and executing complex applications is becoming prohibitive. Fault injection at the RTL level can require several months of CPU time. This strongly impacts the project TTM and poses a serious threat on the success of a product in case the target reliability levels are not reached and redesign of part of the system is required. Moreover RTL fault injection requires a full system already designed and can be applied only in the late stages of the design process. At these stages, design modifications to improve reliability are excessively costly.

The FP7 Collaboration Project CLERECO addresses early system reliability evaluation with a cross-layer approach [28–30]. The fundamental objective of the project is to investigate methodologies to accurately perform system reliability analysis focusing on the early stages of the design cycle for the future systems of the emerging computing continuum [31].

This paper presents an overview of the CLERECO project at the end of the first year of its research activity. It focuses on the tools that are being developed that could be of interest for the research community and engineering practice. Given the limited space, the paper does not provide detailed descriptions of all developed models and tools. The emphasis of the paper is instead to present CLERECO's perspective on the way system reliability analysis can be performed with a cross-layer approach considering the main layers that constitute a modern digital system.

The paper is organized as follows: Section 2 introduces the cross-layer approach to evaluate the system reliability. Section 3 overviews CLERECO's general methodology to perform system reliability analysis, and Sections 4 and 5 describe tools to evaluate the hardware and software contribution to the system reliability. Eventually, Section 6 presents final considerations and future perspectives for the system reliability estimation.

2. A cross-layer approach for system reliability evaluation

Performing cross-layer system reliability analysis, requires a deep understanding of the layers where faults appear in the system, how faults generate errors, and how errors propagate across layers, eventually impacting the final mission of the system.

Fig. 1 provides a graphical representation of how faults may be generated and propagated in a system. Following the Computing Community Consortium Visioning Study on Cross-Layer Reliability [32], a system can be seen as a stack go three main layers:

1. the technology layer that accounts for the raw technology used to build its hardware blocks,
2. the hardware layer that accounts for the hardware blocks and their architectures built on top of the technology, and
3. the software layer that includes the system and user applications executed on the hardware platform.

The technology used to build hardware components is the main root of hardware faults due to physical fabrication defects, aging or degradation (e.g., NBTI), environmental stress (e.g., radiations), and fabrication variability, etc. Within CLERECO we focus on how these faults propagate through the other layers composing the system. After a raw fault manifests in a hardware block, it can be propagated through the different hardware structures composing the system. Several masking effects can mitigate the impact of these faults. We define as *vulnerability factor* the conditional probability of a component to produce an erroneous result given the occurrence of a raw error in one of the lower layers of the design hierarchy. Several vulnerability factors do exist in a system.

Faults can be mitigated at the technology level by timing effects that prevent erroneous values to be sampled by memory elements (Time Vulnerability Factor – TVF) [33–35], or by logic masking effects (Cell Vulnerability Factor – CVF). Faults that manage to cross the technology layer and enter the hardware architecture layer can still be masked both at the micro-architecture level (μ Architecture Vulnerability Factor – μAVF) or at the architecture level (Architecture Vulnerability Factor – AVF) [36–38]. Finally those faults that are not masked at the hardware layer enter the software layer of the system by corrupting either data or instructions of software applications. These errors can damage the correct software execution by producing erroneous results if the computation is completed, or by preventing the execution of the application by causing exceptions, interrupts, abnormal terminations or applications hang-up. Nevertheless, the software stack can also play an important role in masking errors, introducing a further error masking effect (Software Vulnerability Factor – SVF), which may further improve the system reliability [39–46].

Performing system reliability analysis means calculating the different vulnerability factors associated with the components of a system, and then understanding how all masking effects work together and how they influence the behavior of the system. Fig. 2 provides a high-level view of the CLERECO cross-layer reliability evaluation flow. The key concept exploited in CLERECO is to analyze the three system layers separately computing different vulnerability factors for the different blocks. Vulnerability factors are then statistically combined in order to infer reliability measures at the system level. Analyzing the layers in isolation has the main advantage to reduce the complexity of the analysis focusing on the peculiar masking effects each layer can provide. As reported in Fig. 2, each layer defines an interface with the upper layer, which in turn sets how faults can be propagated from one layer to the next one. For each layer, in CLERECO, we devise to identify a set of tools and models able to perform this characterization.

Among the three layers composing the system, the technology layer is probably the most well studied layer. Studying faults that may affect new technologies such as FinFET [47], starts from the definition of predictive models [48] for the technology and requires the development of models for the basic cells (e.g., memory cells, boolean gates, etc.) that need to be analyzed. Resorting to these models, electrical simulations (e.g., Spice, TCAD) can be used to compute failure probability and to derive the TVF that will be required for the analysis of the next layers of the stack.

In this paper we focus on the vulnerability factors introduced by microprocessors and software routines and on the statistical models used to combine those vulnerability factors.

3. System level reliability modeling

Early system reliability analysis requires the identification of a proper high level statistical model enabling to represent the system and its vulnerability factors and to perform statistical reasoning.