Dependability Assessment of a Fault-tolerant Stack Processor

Olaf Malassé¹, Khalid Bouragba²*, Hicham Belhadaoui², Jean-François Aubry³, Mohamed Ouzzif²

¹ Arts et Métiers Paristech, Metz, France
   olaf.malasse@metz.ensam.fr
² Laboratoire RITM ESTC, CED ENSEM, Université Hassan II Casablanca, Maroc
   {bouragba, ouzzif, belhadaoui}@est-uh2c.ac.ma
³ Centre de Recherche en Automatique de Nancy, France
   jean-francois.aubry@ensem.inpl-nancy.fr

Abstract. Nowadays, electronic systems are increasingly becoming attractive for many applications. Such systems aim to be more and more dependable, a fact that requires the assessment of properties such as the availability of safety function, the probability of spurious trip etc, and the identification of weak points. In our case study, the presence of programmable electronics devices imposes the existence of hardware/software interactions. In this work we apply the informational flow approach [12] to evaluate some dependability parameters of stack processor architecture [15] in order to make adjustments during the co-design step. The information flow model is derived from a VHDL-RTL modeling of the processor instruction set. Validation of this model is not part of this paper. This work assesses important reliability parameters of an embedded application in a stack processor architecture using two dynamic models. The first one (stack processor emulator [14]) allows the study of dynamic performance and the impact of a fault-tolerant technique. The second one (information flow approach [12]) evaluates the failure probability for each assembler instruction and for some program loops. The main objective is to estimate the failure probability of the whole application, by assuming completeness of VHDL-RTL model and sufficient verification during design phases of the processor. Fault tolerance is an essential requirement for critical programming systems, due

* Corresponding Author. Email: bouragba2008@gmail.com
to potential catastrophic consequences of failures. The challenge in this paper is to take into account the hardware-software interactions in the same model; which of the two generates the problem, is it the software, or the hardware or both of them? Hence, we suggest a methodic undertaking to develop a reliable architecture in the first stages of the co-design phase. The hierarchical modelling of information flow approach makes it possible to evaluate the efficiency of protecting program loops. These loops ensure the fault tolerance policy by recovering imminent failures and allow the application to run successfully thanks to permanent software recovery mechanism. In the last part of this work, we suggest a reliable solution based on the separation of concerns via the aspect oriented programming. This gives the possibility to decrease time execution.

**Keywords:** embedded systems, device modeling, reliability issues, VHDL-RTL modeling, fault-tolerant, IEC61508, stack processor, separation of concern, aspect oriented programming, NFR.

### 1 Introduction

- The complex systems require a dependable architecture and, hence, should take into account hardware/software interactions since the very beginning of the co-design process [18, 20]. This is even more important when looking strongly at dependable systems. One of the problems appearing in programmable systems - especially in embedded applications- is the evaluation of dependability attributes. The validation of the design of these complex systems is very difficult [8]. These systems operating in industrial environment are subject to different radiation phenomena whose effects are often called “single event upset” (SEU). Generally, these systems make use of software techniques to address and tolerate these soft errors. Fault-tolerant strategies can be integrated and implemented in order to enhance the architecture, to recover imminent failures or to allow the application to run successfully. By applying the information flow modeling on several benchmark applications, we evaluate the probabilities of different functional modes.

- Most studies, however, have focused on fault coverage and error latency of hardware fault-tolerant mechanisms in digital systems as dependability measures [22]. Recently, it has been reported that the environmental transient faults could be masked only by software without hardware error masking mechanisms [1]. Thus, a substantial number of faults do not affect the program results for several reasons: faults whose errors are neutralized by the next instructions, faults affecting the execution of instructions that do not contribute to the benchmark results, and faults whose errors are tolerated by the semantic of the running benchmark. This effect should be considered properly because even a small change of system, fault coverage value can affect the system’s dependability [12]. Using the stack processor emulator and benchmarks, an emulation model for fault injection was developed to estimate the dependability of the com-