METHODOLOGY AND ALGORITHMS FOR HIGH-LEVEL MODELLING OF COSMIC RADIATION IMPACTS ON ELECTRICAL SYSTEMS

by

HASSAN ANWAR

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, "02-DECEMBER-2017"

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Claude Thiebeault, Thesis Supervisor
Département de génie électrique à l'École de technologie supérieure

Mrs. Yvon Savaria, Co-supervisor
Département de génie électrique à l'École Polytechnique de Montréal

M. First Name Last Name, President of the Board of Examiners
Department and institution

M. First Name Last Name, External Independent Examiner
Department and institution

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "20-DECEMBER-2017"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

METHODOLOGY AND ALGORITHMS FOR HIGH-LEVEL MODELLING OF COSMIC RADIATION IMPACTS ON ELECTRICAL SYSTEMS

HASSAN ANWAR

ABSTRACT

The effects of cosmic radiation (CR) on aircraft's embedded electronics are part of research from last few years. The low electrical conductivity of composite materials combined with the required increasing voltage levels of the aircraft lead to reinforcement of electromagnetic (EM) protection. Aircraft implies more electrical systems while composite material does not bring the same level of EM shielding against conventional EM environment. Aircraft flying at altitude/latitude (55,000 feet), for long flight times (more than 15 hours) and cross polar routes (North or South latitudes) are prone to CR. Without an atmosphere to protect from ionizing or particle radiation, current CMOS based electronics are subject to hard and soft errors, generalized performance reduction, accelerated wear, and, ultimately, unrecoverable system failure. Consequently, equipment protection against CR is becoming as critical as protection against any external environment. Today, solutions to protect electrical systems from CR are developed in an incremental way from previous observation, experience and knowledge. Unfortunately, these solutions are costly, time, and energy consuming e.g., dedicated heavy conductive electrical path way and redundant electrical functions. Consequently, to progress more rapidly towards the safe and energyefficient aircraft, it is now necessary to anticipate the integration/installation constraints of the electrical system in the early phase of the aircraft design to relax weight and drag penalty of the CR plenty. To this end, electrical system providers need a unique computer environment for performing CR prototyping supporting the decision making for the selection of the most suitable light-weight CR protective solutions, while maintaining safety at its highest level. In this project, we will study the novel algorithms and methodology for high levels modeling of cosmic radiation impacts on the aircraft flying at the altitude/latitude of 55,000 ft. This project is the extension of the AVIO403 project which studies the impact of cosmic radiation onboard avionics systems and also the part of a big project named EPICEA (Electromagnetic Platform for lightweight Integration/Installation of electrical systems in Composite Electrical Aircraft). In this project, starting from the review process of the AVIO403 project. We will perform the bibliographic review of the CR effects on the electrical systems. The results and data collected during the AVIO403 project by using already available software e.g., MATLAB. We will develop a dynamic highlevel fault simulator that consists of the analysis of consequences of cosmic radiation effects on electrical systems. This project aims at defining a novel approach for high levels modeling of cosmic radiation impacts on electrical systems. In particular, our research intends to provide solutions able to mitigate multiple problems, including, but not limited to a) SRAM based CR emulation strategy for complex systems, b) Signature generation on the FPGA-based emulation platform and for the radiation-based experiment c) A wellthoughtful preparation of cosmic radiationbased experiments at TRIUMF in Vancouver. d) Modelling the faulty behaviour for the more complex system e.g., sequential circuits, generation and analysis of the signatures. We also need to adapt the results to aircraft conditions since the data recorded on the test-bench aircraft and in-

flight experiments are/will be on a metallic structure. We will use behavioral simulation tools to evaluate the consequences of CR in different conditions at electrical components and systems level. We also add prediction features in our software to predict the behavior of cosmic rays. We will develop the methodology to enable computer model that helps control interactions of cosmic rays with the electronic components. We will investigate the sensitivity of electrical systems to CR concerning their criticality level. We need to focus on the electronic components malfunctions and damages. Few companies e.g., ISONEO and Bombardier Aerospace will involve. These companies provide some useful guidelines e.g., ISONEO will provide the requirements for their CR computer models at system level that would apply to the EPICEA platform. We will present our results and perform the experiments with those gathered during the EPICEA project.

Keywords: FPGA, SEU, Modeling

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

DUT	Design Under Test
FPGA	Field-Programmable Gate Array HDL Hardware Description Language
SBU	Single-Bit Upset
SEE	Single-Event Upset
SEL	Single-Event Latch-up
SER	Soft Error Rate
SET	Single-Event Transient
SEU	Single-Event Upset
SRAM	Static Random Access Memory
TMR	Triple Modular Redundancy

CHAPTER 1

INTRODUCTION

1.1 Context and Motivation

This Thesis is a part of a joint EU-Canada research project - called "EPICEA" - Electromagnetic Platform for lightweight Integration/Installation of electrical systems in Composite Electrical Aircraft will approach numerous avionic engineering design issues in the advancement of future aircraft, aiming at a significant reduction of energy consumption through more electrical aircraft and systems integration. This thesis strives to understand the electromagnetic (EM) issues on composite electric aircraft (CEA). This includes the analysis and characterization of EM coupling, interconnects, and cosmic radiations (CR) on electrical systems together with new concepts of antennas design to maintain performance in a composite environment without modifying aircraft aerodynamics.

Our contribution to this project (EPICEA) — "the study of CR effects on aircraft electrical systems based on the reconfigurable fabric, e.g., Field Programmable Field array (FPGA)." This research work focuses on design and implementation of the FPGA-based platform to help to investigate the effects of CR on an embedded electronic system of the aircraft and to model and analyze the faulty behavior of the sequential digital circuits. This thesis aims to make the high-level fault model that will use to investigate the effects of radiation on the aircraft flying at the altitude of 50,000 feet or above. This thesis helps to find; at high altitude when aircraft gets more exposure to the radiation the reliability of the FPGA based embedded systems is a primary concern; the phenomena causing faults in the FPGA based embedded systems should be studied in a way to know early in the embedded electronic design of the aircraft if mitigation strategies are required to deal this higher radiation level.

In order to accomplish our tasks, we need to understand the space radiation environment which has two preliminary sources — galactic cosmic radiation and solar energetic particles Tobiska *et al.* (2015). The galactic cosmic radiation from outside the solar system consists mostly of

energetic protons and heavy ions, e.g., iron. Solar energetic particles are commonly associated with the solar flare events and primarily dominated by the proton. The space radiation is an unavoidable space weather phenomena. This research thesis is solely concern with the aircraft flying at the altitude of 50,000 feet or above, where the vulnerability of the circuit is due to the neutrons Xilinx (accessed 2017a). The cosmic rays are originating in outer space and travel at nearly the speed of light and strike the earth from all directions. These cosmic radiations are ranging from lightest to heaviest elements in the periodic table. When these high-energy cosmic rays interact with the earth's magnetosphere, neutrons are generated, often referred to as an air shower Lesea *et al.* (2005). An intense neutron environment exists at higher altitudes in the atmosphere, 10 km to 40 km. Long-haul aircraft are flying at the altitudes of 50,000 feet nearly 15 km at the latitude of 60° under the influence of greatest neutron flux approximately 500 times that a ground-based observer in Newyork City Lesea *et al.* (2005). When these neutrons interact with the semiconductor, e.g., silicon they can produce secondary particles if these secondary particles are charged and they can generate the trails of an electron-hole pair of a few microns in length and if this trail happens near the PN-junction in a transistor, a voltage spike can generate. This voltage spike is enough to change the state of a memory cell or flip-flop, results in a drastic consequence and cause Single Event Upsets (SEUs) also called soft-error in SRAM-based FPGAs. The impact and implications of the soft-errors due to the interaction of neutrons with the avionics embedded system are now recognized as an area of active research. Especially, the incident happened with the Qantas Flight Airbus A330-303 flying from Singapore to Perth went under the two terrifying dives due to the malfunction of the on-flight computer. After, the investigation it revealed that high-energy particles from the outer space — were the responsible for the malfunction of the computer. And, the potential triggering event was the single-event effect (SEE) interacting with one of the integrated circuits (ICs) within the CPU module.

Therefore, fault management strategies are essential to apply on the aircraft's embedded systems. Before need to know to apply the mitigation techniques early in the embedded electronic design, if we know the high-level fault model of the FPGA-based circuits at high-level of ab-

straction that facilitate without going into the detailed simulation get the faulty behavior of the component at high-level.

1.2 Problem Statement

A soft error will occur when a radiation event causes enough charge disturbance to flip the state of a logic gate, memory element, or flip-flop. The soft errors are also referred as radiation-induced faults, e.g., neutrons from the cosmic rays. The vulnerability of FPGA based embedded systems to soft errors increased as an undesired side effect of advancement of semiconductor technologies. The fault caused by the SEUs have implications on the behavior of the system. However, the soft errors do not break the silicon but can corrupt the underlying functionality and produce severe consequences if they are successfully propagated to the primary outputs or stored by the memory cell. In this work, we will evaluate the faulty behaviors of the sequential circuit, given the faulty sequential circuit behavior, and model it to the high-level of abstraction. We want to make the high-level fault model to see the severity of the faulty behavior induced by the SEUs.

1.3 Research Objectives

In this thesis, I propose a methodology for modeling the faulty sequential circuits to model the output of the soft error problem at a behavioral level. We will use the high-level modeling techniques, e.g., markovian-system analysis to model the faulty behavior of the sequential circuits. The faulty behavior model is based on the analytical expression for modeling the signature (circuit fault response) from the experimental data generating by performing fault emulation on the FPGA. In this work, a novel approach will introduce to analyze the fault origination and propagation in the case of faulty sequential circuits, and will develop new models that can accurately see the severity of the faulty behavior from the signatures of the faulty circuits. The objectives of the present Ph.D. thesis are explained below:

- The main objective of this thesis is to develop a faulty behavior model for FPGA-based sequential circuits described at a high-level of abstraction. The main purpose of developing a behavioral fault model of a circuit is to generate a library of faulty components reuseable at high-level of abstraction

- The developed models could be used to replace any component of the entire circuit with faulty versions of the components described at a high-level of abstraction. The purpose of doing so to ensure that the effect of faulty behavior of each component on a system could be analyzed at a high-level of abstraction and the mitigation technique could be used to improve the robustness of critical parts of the design.

- The model will have the capability to compute the signature, also provide the worst-case input test vector, which has the highest probability to generate the faulty output, for any given sequential circuit. The model for sequential logic will have the ability to measure the maximum and average signature probabilities to construct a novel probabilistic model.

- Develop an abstraction of hardware signature that can be integrated into the high-level model, e.g., Simulink models.

- The goal of this research is to develop an approach for modeling the faulty behavior of a digital sequential circuit in the presence of the fault injection. The concept behind the fault injection process is to accelerate the occurrences of the signatures in the system to evaluate its functional behavior under the influence of expected faults on the FPGA-based systems.

The main challenges we foresee are:

- Emulate faults in the models at high-level of abstraction. A suitable fault injection will require.

- Soft-error analysis for the sequential circuits.

- Make a model at higher-level of abstraction from the data extracted at a lower level.

- Fault behavioral modeling.
- Fault behavioral model of different sequential circuits, e.g., counter, and FIR filter.
- Develop a library of the faulty behavioral components of the circuit segments: forming a Simulink model.

1.4 Contribution

This research thesis proposes a fault behavior model develop with the modeling techniques, e.g., Markovian-analysis in a novel way. The markovian system analysis will use to synthesize the faulty output of a circuit at high-level of abstraction. The existing faulty behavior model of the sequential circuits are not accurate as mostly models were based on simulation and on fault occurrence assumption.

- In this work, we will construct a high-level fault model that model the signature of the sequential circuits (ITC'99 benchmark). The high-level behavioral model based on the signatures from the sequential circuits.
- This work will focus on the soft-error susceptibility for sequential circuits which are different from the combinational circuits. The error in the sequential circuit can be propagated back to the inputs, or circuits outputs can be affected for several consecutive clock cycles making the design more vulnerable.
- Our model will have the capability can efficiently model the worst case signature and the input vector correlated with this signature, through probabilistic modeling.
- The objective of the model is to find the severity of a faulty behavior of the sequential circuit.
- We will provide the signature for a sequential circuit.
- Study the system susceptibility under neutron-induced single event effect.

- We will study the circuit-specific mitigation bounds for fault-tolerant computing.

CHAPTER 2

RELATED WORK AND PRIOR WORK

This chapter is dedicated to the revision of some of the fundamental concepts and current research in different areas related to this project: radiation effects on SRAM-FPGAs, soft-error, hard-error. Fault-injection, SEUs, Signatures, and benchmarks for radation testing. All of these topics are equally relevant for the purpose of this research that, ideally, places itself as an attractive research project.

2.1 Radiation Environment

A Single Event Effect (SEE) results from a single energetic particle. When the particle strikes a sensitive node in a semi-conductor device, the ionization by the particle might produce a current pulse inside the device, which might cause soft or hard errors in the configurtaion memory of the device. Results in data corruption, transient disturbance, high current conditions (non-destructive and destructive effects). SEE can if not handled well cause unwanted functional interrupts or in worst case catastrophic failures. Commonly, SEEs include single event upset (SEU), single event latch-up (SEL), single event burn-out (SEB), and single event transient (SET) etc as mentioned in Table **??**. SEEs may happen to electronic devices in these environments which is prune to the radiations. For example,

- Space (caused by space radiation)
- Air-plane (caused by atmospheric neutron)
- Close to nuclear reactor (caused by reaction neutron)
- Everywhere (IF caused by natural decay radiation in the materials of devices)

2.1.1 Faults Caused by Cosmic Rays in Digital Circuits

2.1.2 Single Event Effects Mechanism

2.1.3 Neutrons Effects on FPGA

FPGAs are complex reconfigurable devices that comprise a wide family of different resources. The basic structure of modern FPGAs includes interconnect resources, clock-management resources, configurable logic blocks (CLBs), input/output blocks (IOBs), and embedded blocks such as digital signal processors (DSPs), general-purpose processors, high-speed IOBs, and memories. CLBs are used to perform simple combinational and sequential logic. These blocks are typically formed of look-up tables (LUTs), multiplexers, flip-flops, and carry logic. Programmable interconnect resources, such as routing switches, allow interconnecting CLBs, IOBs and embedded blocks to implement multiple systems (Buell et al., 2007). The logic and routing resources in an FPGA are controlled by the bits of a configuration memory, which may be based on either antifuse, flash, or SRAM technology. The design flow of FPGA-based systems as shown in Figure 2.1 adapted from Hauck & DeHon (2010) involves the creation of a bitstream to load into the device.

2.1.4 Shielding Effect

The process starts with the system design written in a hardware description language (HDL), e.g., VHDL or Verilog. Next, the design is optimized and mapped into the FPGA's available resources through logical synthesis, technology mapping, placement, and routing. Finally, the generated bitstream downloaded into the device, and the device starts functioning according to the designer design. Like any other semiconductor device, FPGAs are sensitive to radiation effects. Mostly, these effects depend on the technology used to store the configuration data. Regarding the impact of SEEs on reliability and functionality, FPGAs based on SRAM technology are a particular class of devices. The foremost concern for SRAM-based FPGAs is SEUs within the configuration memory. In such devices, this memory may represent more than

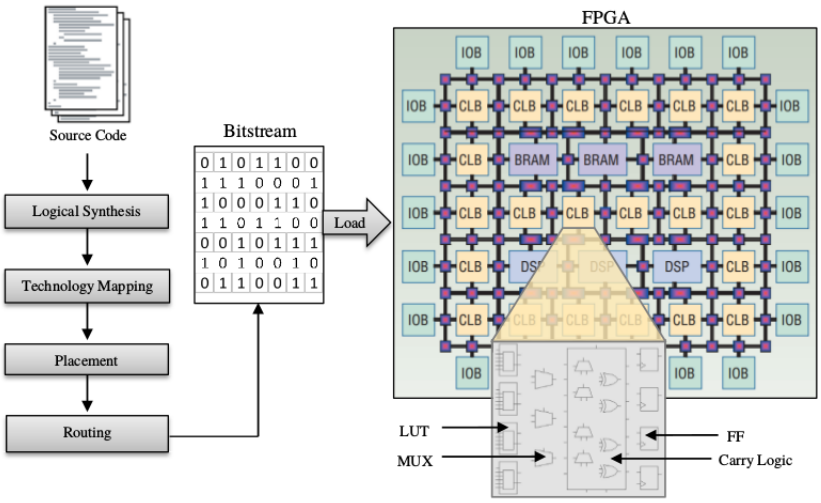


Figure 2.1 FPGA Structure and Design Flow

80 percent of the total memory bits, increasing the probability of configuration faults. Upset configuration bits may change the logic and routing of the implemented system, as shown in Figure 2.2, leading to functional failures in an unpredictable way. In contradiction, the primary concern for anti-fuse and flash-based FPGAs is SETs and SEUs within user flip-flops and block memories. However, the configuration memory blocks of anti-fuse and flash-based FPGAs offer a relative immunity to SEEs, but these devices have lower logic capacity and cannot be reprogrammed an unlimited number of times, making SRAM-based FPGAs more suitable for complex systems requiring frequent reconfiguration and adaptation Quinn & Wirthlin (2015); Violante *et al.* (2004).

2.2 Design Verification by Fault Injection

As we discussed before, SRAM-based FPGAs are particularly sensitive to SEUs. The configuration memory is the most sensitive part, by changing the configuration memory, may affect the overall functionality of the system. The work have done so far deal the SEU effects on FPGAs, combines the simulation, radiation, and emulation testing Quinn & Wirthlin (2015); Violante *et al.* (2004); Hobeika *et al.* (2014); Robache *et al.* (2013); Quinn *et al.* (2015); Souari *et al.* (2015). These papers described how they make the faulty behavior of the system to build

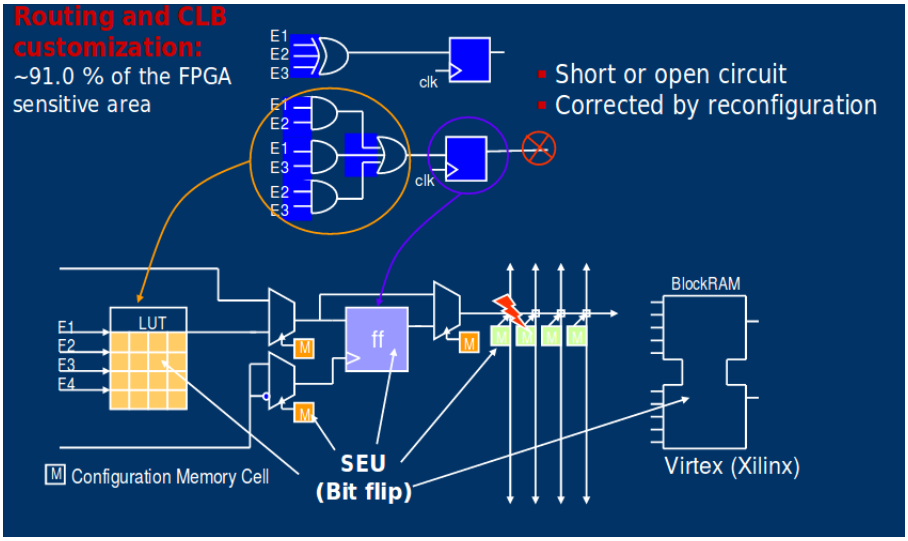


Figure 2.2 Upset FPGA configuration bits may change the logic and routing.

an accurate representation of the system. The work presented in Quinn *et al.* (2015) described the benchmark that can be used for the reliability and radiation effects study on FPGAs and microprocessors. FPGAs offer high densities and run-time programmability facility make inconvenient to use in the aerospace domain. But, FPGAs are sensitive to high-energy ions. We need to study the sensitivity of SRAM-based FPGAs to heavy ions that show the suitability and analysis of effects of radiation on FPGAs when employed in space, e.g., usage of FPGAs in aircraft. The work presented in Hobeika *et al.* (2014) investigate the sensitivity of SRAM-based FPGAs devices not only for the simulation-based approach but also used emulation and radiation testing for evaluating the effects of SEUs. The work presented in Souari *et al.* (2015) described the fault injection emulation in Xilinx FPGA based on the identification of critical configuration bits. Based on SRAM-based FPGAs, two aspects can be considered:

- SEUs may alter the contents of a register in the data path, or the content of the state register.
- SEUS may alter the content of the configuration memory.

2.2.1 Simulation

The work presented in Robache *et al.* (2013) discuss the fault simulation, fault emulation, and radiation testing. Starting from the simulation, I can interrogate how the authors used the concept of signatures to capture and reproduce the faulty behavior due to SEUs very early in the design process. Radiation testing is an expensive approach and requires a state-of-the-art facility. The alternative to the radiation testing is the fault-injection approach. The work presented in Hobeika *et al.* (2014) described the concept of faulty behavior signature. The work demonstrates how faulty behavior signatures allow building high-level models, e.g., high-level faulty model, i.e., Simulink, that reflects the faulty behavior of a combinational circuit represented at gate-level (injected with one fault arbitrarily selected from a fault list). The main contribution of this work is to capture the effects of radiations on a circuit modeled at a low abstraction level and then abstract it to a higher level. This challenge can be accomplished by introducing the concept of faulty behavior signature. The fault injection tool that is used named - LIFTING. The purpose of this tool is to study the effects of different types of faults on a circuit at gate-level. The tool used the stuck-at 0 and 1 are injected in each node of the design. The LIFTING is a simulation-based gate-level fault injection tool that used the circuit netlist file, e.g. *.v file, input test vectors, and fault parameters as inputs produced two output files. The one is the golden report and the second consist of fault injection report. These two outputs are used to generate the signatures. The signature represents the compressed faulty behavior of the circuit. The signatures consist of arrays of errors and their probabilities of occurrence. The signatures are either arithmetic or logic. The third step is to make a high-level model that corresponds the low-level circuit. The work presented in this paper helps to make a faulty block with Simulink that reads a signature and generates errors according to the distribution.

2.2.2 Emulation

The emulation of SEUs in an FPGA is done by flipping the bit in the configuration memory. The emulation can be done by using the IP provided by the Xilinx named - LogiCORE. The work adopted the emulation work also proposed in the Hobeika *et al.* (2013). The work de-

scribed a completed automated methodology to emulate SEUs on an FPGA efficiently. The authors used the reconfigurable flight control system based on a reference adaptive control model. The difference between the work presented in Hobeika *et al.* (2014) and Hobeika *et al.* (2013) is that; in Hobeika *et al.* (2013) the authors used the flight control system that is based on a linear plant model. Whereas, in Hobeika *et al.* (2014)the emulation is performed on the circuits (adder and multiplier). The work presented in Hobeika *et al.* (2014) used the SEU controller. The emulation is the four step process.

- Identification of an emulation zone.
- Fault list generation.
- SEU emulation.
- Result Analysis.

The identification of an emulation zone used the concept of the essential bits which can be extracted by the Xilinx BitGen command. For example, 253227 bits are identified as the total essential bits in Hobeika *et al.* (2013), among them, 57464 belongs to the interested essential bits. The step is used to minimize the time because an FPGA device contains millions of configurable bits, emulating a bit flip for every cell would be time-consuming. BitGen gives only the essential bits; that considered critical bits. The second step generates the fault list. This action creates a list of the corresponding bit addresses (exact bit position to be emulated). For example, authors observed 7000 emulation requests in Hobeika *et al.* (2013). The third step used auto-correct mode in which one bit is flipped at a time and the detect-only mode (bit flips accumulation possible) where bits are flipped without correction. In the final step, an in-house script is used to characterize and quantify the design sensitivity to SEUs. This script is used to compare the results with the faulty one and fault-free. Authors observed 638 total number of failure in Hobeika *et al.* (2013). Similarly, authors observed 80384 essential bits among them 2454 are considered as the interested ones for the adder circuit, for multiplier interested essential bits are 1314 among 92337 total essential bits. The emulation can be performed on the

Virtex 5. The 16-bit adder and 8-bit-by-8-bit multiplier are used as a testing circuit. The signatures are recorded in the accumulation mode. And, the estimation of the critical bits performed in the auto-correct mode. The Emulation setup presented in Souari *et al.* (2015) adopted the approach for the estimation by fault injection based on the sensitivity. Authors proposed the method in which fault are injected based on the specific bits configurations defined according to their contents and the type of FPGA resources. This new approach outperformed the traditional random fault injection with speed up factors to two orders of magnitude. This fault injection method based on the prioritizing specific subsets of configuration bits. These configuration bits are classified with the statistical analysis according to their values (0 or 1, and 2). The SEU controller a macro developed by Xilinx assuring fault injection, detection, and correction is used a fault injection engine in their experiments. The fault injection is prioritized using the following three steps:

- Classification of the configuration bits into subsets. a. Bits set to 1/0 of LUT. b. Bits set to 1/0 configuring other than LUT. c. Bits set to 1/0 configuring other resources not identified as potentially critical by bitgen.

- Estimating the number of critical bits of the set by randomly injecting faults in the bits of each set. This method helps to find the most critical zones of the FPGA.

- Prioritized the fault injection in the identified (step-2) most critical zones. These classification steps are done with the help of EBC and EBD files provided by the bitgen. The experimental results presented in [5] evaluated the SEU sensitiveness as well as bitgen efficiency. The results are evaluated between random fault injection with different prioritized bit subsets. The first observation authors concluded - the bitgen did not accurately identify all the critical bits meaning the bitgen limitations. Second authors did the prioritizing the most sensitive subset. It would involve exhaustive fault injection. The authors used fault injection to get an estimated number of critical bits as well as the related estimation error. They used the term critical bit error estimate (CBEE). The authors claimed the CBEE observed for the random approach is higher than the observed under the bits subsets. The

ratio of observed critical bits (ROCB) observed for the random injection is far less than the different bits subsets.

2.2.3 Radiation testing

The hardware setup consists of two Artix-7 board. Board A used as a reference and board- B is subjected to radiations. The board-A is not bombarded, and it hosted the counters, reference design error detection and signature computation, memories to store signatures and communication controller. A total 20 runs performed on the adder and 14 on the multipliers. Arithmetic errors for both approached DSP and LUT are observed (151 vs. 291 for DSP). This is due to DSP strategy; SEUs can add registers in the data path, leading to the sequential type of errors. The authors in this work compare the results from the fault simulation, fault emulation, and radiation testing. The purpose is to express as signatures, intended to reproduce the faulty behavior. They showed that simulation and emulation based signatures could contain the same error values as obtained with radiation but their probability of occurrence could significantly different. The arithmetic signature for TRIUMF to emulation is 85.3

2.3 Fault Models

2.3.1 Stuck-at Fault Model

2.3.2 Functional Fault Model

The suitable selection of the benchmark for the radiation testing of microprocessor and FPGAs is a recently topic of ongoing research. The benchmarks are used to evaluate the performance under different architectures, technology, and compiler. There is no such standard benchmark employed to study microprocessor and FPGAs under the effects of radiations; make it difficult to assess the changes in fabrication technology, architecture, and circuitry. The work presented in Quinn *et al.* (2015)described the software and hardware benchmark under the neutron test

data. The unavailability of the such a benchmark for testing because radiation hardness assurance techniques are applied only to circuit layouts or manufacturing process. There is no standard test circuits available, researcher, used flip-flop or D-latches to compare their results. In recent years, radiation effects community shown interest to develop a standard set of circuits that include complex and realistic algorithms and can be adapted to different FPGAs. Currently, without standard benchmark researcher used the following approach for testing:

- Homemade Design.
- Circuits from Opencore.
- Proprietary designs.

The problem with this approach as no two organizations used the same set of codes or circuits, difficult to make the comparison. There is a need for collaboration to make a suitable set of benchmark for reliability application and study the effects of radiation under the same conditions. The criteria used to set a standard benchmark including:

Repeatability of benchmark tests. A representative of deployed computing workload. Availability of fixed input vectors. Cross-platform implementation. The ability to repeat test itself is an important part of the standardized testing. By repeating the algorithms, the input test vector, the compilation, the synthesis setting help researchers to have the enough information. It is necessary to provide a wide variety of realistic algorithms so that the system can be tested as likely to the realistic application. Defining the input test vector is an essential step because many hardware errors can be observed under the specific set of the test vector. It is an open question which input test vector should be adopted, under the specific set of criteria. Finally, the implementation of the algorithms in portable languages help to use the same set of codes on the different platform. For example, assembly language for the microprocessors limit the ability to compare and port codes on the different platform. But the hardware benchmark developed in VHDL can ease the problem; the same circuit can be ported to any FPGA.

FPGA Radiation Benchmark

The FPGA benchmark mentioned in this paper is ITC'99 which is well defined ATPG benchmark. This benchmark meets all the requirements, e.g., realistic algorithms, input vectors, scalability, and portability. The circuits are implemented in the HDL so that it can be ported to different FPGAs. The first 15 circuits from the ITC'99 are adopted for the benchmark as shown in Table I.

Software Radiation Benchmark

The software radiation benchmark is harder to design than the FPGA radiation benchmark. The development of the standard set of algorithm that can be ported on different architectures would be a challenging task e.g., porting an algorithm to 16-bit microcontroller to GPU. The authors are interested in the software benchmark where the computational load can be divided into the parallel processes or run on a single core. The commonly used software benchmark comprises of fast fourier transform, matrix multiplication and quick-sort algorithm as they are commonly used in many applications and useful for the evaluating the reliability of parallel processors. The software benchmark comprises the following code.

- AES-128;
- Cache test;
- FFT;
- Hotspot;
- HPCCG;
- Matrix Multiply;
- Quicksort

Radiation Testing The radiation testing is completed at Los Almos Neutron Science Center (LANSCE). The results are provided for the microcontroller, ARM cores, GPUs, and FPGAs.

The B13 from ITC99 is used under the hardware benchmark suite; Virtex- 5 is used as a hardware platform. They also provide the result for the mitigation. For mitigation, they used X-TMR and VERI-Place. The failure in time (FIT) are decreased under mitigation, but the overhead is increased (circuit area increased).

Hardware Benchmark Testing For the hardware radiation testing the authors used the B13 from the ITC'99 benchmark suite. The circuit is too small so it can be replicated 30 times, the implementation is done on the Virtex-5. Both unmitigated and mitigated version are tested. The results for FPGA radiation reports SDCs from the mitigated circuits normalized to the SDCs from the unmitigated circuits. Mitigated circuits are likely to fail at three times the rate of the unmitigated circuit, because of the increased size of the circuit from the mitigation process. The mitigated circuit cross-section is three times larger than an unmitigated circuit when SEUs accumulate. The authors conclude; the VERI-place mitigated circuits perform better than the X-TMR mitigated circuits.

Software Benchmark Testing Software benchmark radiation testing is done on the flash-based microcontroller, a ferroelectric-memory-based microcontroller, two ARMs, and GPUs. These components are tested with both mitigated and unmitigated codes. The results reported in the paper for two different microcontroller and two ARMs cores. For microprocessors: these microprocessors have very small SRAM the FITs are very small. In some cases, there is no error from the code during many days of testing. They also implemented the matrix multiplication, FFT, and Hotspot on NVIDIA K20 GPU and applied mitigation methods (ECC, ABFT, and DWC). The purpose is to see the effect of overhead by applying the mitigation technique; the overhead has been increased as compared it with the unhardened configuration. In short, the work presented in [3] evaluate a common set of hardware and software benchmarks to evaluate reliability and radiation effects on FPGA and microprocessors.

The impact of SEUs on SRAM FPGA devices has been studied in Bellato *et al.* (2004). Many techniques have been proposed to provide highly reliable FPGA devices, e.g. radiation-hardened FPGAs Rockett *et al.* (2007), in-order to lower the effect of radiation-induced SEUs.

However, radiation-hardened SRAM FPGAs typically have a low density, and they only may lower the probability of SEUs to occur but not completely avoid them. Therefore, non radiation-hardened FPGAs, like the Xilinx Kintex-7, are evaluated under a harsh radiation environment Wirthlin *et al.* (2014). Even on radiation-hardened FPGAs, the SEU rate in a low-earth orbit flight experiment can be up to 16 events per day Quinn *et al.* (2012). A wide variety of SEU fault mitigation techniques for SRAM-based FPGAs have been proposed during the past years. These techniques can be categorized into module redundancy techniques such as triple modular redundancy (TMR) Lyons & Vanderkulk (1962) and techniques that use scrubbing of the FPGA configuration memory Heiner *et al.* (2009). Also the combination of both techniques has been shown to be able to increase the reliability of FPGA modules significantly Ostler *et al.* (2009). FPGA-based TMR approaches replicate a given module which shall be protected either statically or dynamically Angermeier *et al.* (2011). The different granularities of voted replicas are evaluated in Bolchini *et al.* (2007). However, no upset rates and consequential no reliability figures are provided. Nevertheless, TMR techniques are known to often cause an excessive and unacceptable overhead in terms of power consumption and area. Since the intensity of a cosmic rays is not constant but may vary over several magnitudes depending on the solar activity, a worst-case radiation protection is far too expensive in most cases. A self-adaptive system is proposed in Glein *et al.* (2014), which monitors the current SEU rate and exploits the opportunity of partial reconfiguration of FPGAs to implement redundancy such as TMR on demand.

Memory scrubbing is a well-known correction technique for the configuration memory of SRAM-based FPGAs. It consists on re-writing the configuration memory after the FPGA is configured to restore its original content. It is often a transparent operation for the running application. This is possible because modern FPGAs offer a dynamic partial reconfiguration (DPR) feature. The circuit that enables the scrubbing is commonly named scrubber. Additionally, readback is the process of reading the configuration memory of the FPGA after it is configured. Both processes (readback and scrubbing) can be used to implement different scrubbing methodologies as shown in Herrera-Alzu & Lopez-Vallejo (2013). Scrubbing can

be implemented using an internal or external interface as shown in Berg *et al.* (2008). When external interface is used, the scrubbing logic is implemented outside the FPGA. In the case of Xilinx FPGAs several external interfaces are available; however, the Select MAP interface has the highest data throughput. On the other hand, there is only one internal interface named ICAP Xilinx (accessed 2017b). This internal interface can be accessed from the reconfigurable logic of the FPGA and it is a replica of the Select MAP interface. Also scrubbers can be implemented in software or hardware. The scrubbing process can be implemented using a microprocessor with the advantage of a high flexibility to implement different complex scrubbing methodologies but with lower configuration speeds and lower energy efficiency.

2.4 Faults Behavioural Modeling

2.5 Conclusion

The work done so far [1, 2, 3, 4, and 5] evaluated and quantified the SEU effects by performing simulation, emulation, and radiation on an SRAM-based FPGA. Implemented a design, observed its faulty behavior in the presence of SEU and extracted the corresponding fault model. Presented an automated methodology to efficiently used the SEU controller. Discussed the fault injection on the specific subsets rather than random and discussed the selection of the suitable benchmark for FPGA and microprocessor radiations.

CHAPTER 3

PROPOSED APPROACH

This chapter is dedicated to the methodology that we propose to exert to achieve the objectives of this research project, i.e. Methodology and Algorithms for High-level Modelling of Cosmic Radiations Impacts on Electrical Systems. First of all, we will identify four main research axes: (1) Fault emulation platform for sequential circuits to generate signatures; (2) radiation-based experiments; (3) high-level modeling to study radiation impacts on electrical systems; and (4) Simulator- isoneo.

In addition to the development of our research along these axes, we also have a plan to implement a technology demonstrator on FPGA and, fly in an aircraft, e.g., **CMC BEE platform**.

3.1 Sequential Circuit Emulation

3.2 Modeling Faults in Sequential Circuit

3.2.1 Soft-Error Modeling and Analysis in Sequential Circuit

To analyze the faulty behavior of the sequential circuit by using Markov Chain theory is quite obvious choice. Markov chain analysis provide the steady state behavior of the sequential circuit. By using MC analysis we will able to find the number of clock cycles the sequential circuit produce the faulty output.

As mentioned in the section related work, single error rate can be categorized into three different categories a) circuit level b) gate level and c) architectural level. Our work is focused on the architectural level by emulating the fault at circuit level.

Most of the real-time application of the electronic systems are sequential in nature, e.g., random access memories. A typical sequential circuit comprises of combinational logic and flip-flop as shown in Figure. Inputs of the combinational logic, output of the combinational logic, and

inputs and outputs of the flip-flop. There is a temporal correlations between the input signal and the state signal. The state signals are uniquely identified as the function of the input signal and the previous state signal. Due to this; sequential circuit error propagation from the error site, e.g., a bit flip in a flip-flop to the output can observe after several clock cycles. This temporal relationship force to use the more dynamic models than the models availble for the combinational circuits. The sequential circuits models can evolve with the time instances.

We will make our faulty model from the real-time radiation experiment. We will start our analysis by analysing the faulty values. Let us consider the example of the counter as shown in Figure 1 fault free and Figure 2 faulty where output is stuck at 1 .

We will start our analysis by error probability matrix associated with the circuit and the probability that the erros comes in any part of the circuit (Combinational or flip-flop) produce an error at the output.

3.3 Markov Chain Analysis for Faulty Behaviour Model

We used the MC for modeling and analysis of sequential circuits susceptible to soft-errors. We prefer to use Markov chain analysis over the other techniques like BDD/ADD or SAT, As these techniques requires to transform the circuit into their respective tool or mathmatical notation. While, in MC we can directly make the model just looking into the signature values.

The faulty behaviour of a sequential circuit can be analyzed using MC theory. We need to calculate the steady state behaviour of a sequential circuit via MC analysis.

Inorder to do the MC analysis. We need to have a two copies of the original circuit — named Fault-Free circuit, and Faulty circuit (*hit circuit*). The Fault free circuit is used to collect the correct behaviour of the circuit (fault-free outputs and fault free state vectors) and faulty is used to collect the faulty response of the circuit (faulty output and faulty states). From Markov theory, we can define the next state vectors of the fault-free and faulty circuit as:

$NS^{original} = \delta^o = (\delta_1^o, \delta_2^o, \delta_3^o, ..., \delta_m^o)$

$NS^{faulty} = \delta^f = (\delta_1^f, \delta_2^f, \delta_3^f, ..., \delta_m^f)$

$\delta^o =$ Fault-Free circuit output values

$\delta^f =$ Faulty circuit output values

$m =$ Number of output variables

From there I can construct my signature vector:

$\varepsilon_{signature} = \delta^o - \delta^f$

Now, the main goal for the soft error analysis for sequential circuits is to find the transistion probabilities between the signatures from the signature vector $\varepsilon_{signature}$ and from there to determine the faulty behaviour of the sequential circuits when soft-error occurs.

3.3.1 SER Measurement in Sequential Circuits

3.3.2 Modeling with Probabilistic Calculation Methods

This part of the project is a neutron-induced Single Event Effect test in a commercial FPGA from Xilinx. The primary objective is to investigate the radiation effects reliability for the critical application. We will implement the sequential circuit and data acquisition system. The results we want to achieve to drive signatures for the sequential circuits. Our focus is on the analyzing the impact of multiple errors in state flip-flops, during the cycles following the cycle when faults occur. The following milestones we want to achieve from radiation bombardment experiment.

- Modeling of SEU, MBU and analyzing their effect on logic circuits.
- Evaluation of changes in error rates due to SEUs in sequential circuits.
- Compute the error probability, and signatures from bit-upsets can vary for different outputs and different circuits.

- Evaluation of the impact of multiple flip-flop upsets in sequential circuits.
- Determining the outputs that are most susceptible to errors due to faults in logic.
- Determining the parts of the circuit (gates or gate clusters) that have the largest impact on circuit error probability.
- Estimation of lower and upper bounds of circuit susceptibility to transient.

3.4 Optional: Fault Mitigation

Fault-mitigation can be achieved in two ways: preventing faults from happening and recovering after their occurrence. Fault preventing is achieved by using hardened components and/or shielding. But fault preventative is not a viable solution in terms of a project cost. More complex fault-mitigation methodologies can be implemented at the architectural level. We need to develop some fault-mitigation strategies like triple module redundancy with dynamic reconfiguration of the hardware Jacobs *et al.* (2012) and/or something like the work presented in Jacobs *et al.* (2012) used fault tolerance framework (RFT) that enables system designers to dynamically adjust a system's level of redundancy and fault mitigation based on the varying radiation incurred at different orbital positions. Notably, the reconfigurable fault tolerance framework in Jacobs *et al.* (2012) is based on an upset rate modeling tool that used to capture time-varying radiation effects in a given orbit.

3.5 Project Plan

Summary

Phase 01: The emulation platform will be the starting point of research. We will use the SEUs for the configuration memory upsets. Selection of a suitable benchmark, which is probably ITC'99 Polito (accessed 2017)used for the testing purpose. We will evaluate the bits sensitivity as well. We will implement the prototype.

Phase 02: Evaluate the experimental setup under the neutron radiation at Triumf.

Phase 03: Develop an efficient methodology and high-level model for soft-error of sequential circuits, i.e., Monte-Carlo sampling, approximate approaches, symbolic methods for efficient estimation. The simulator development will keep with all these three phases.

CHAPTER 4

PRELIMINARY RESULTS

In this chapter, we present the preliminary results of this research work. These results are focused on the implementation of a probabilistically analysable instruction and data cache for the Ion MIPS32 processor on FPGA. We developed a random placement and replacement policy that fulfills all the requirements for PTA. Our experiments show that the cache fulfills all the requirements for PTA, and program timing can be determined with arbitrary accuracy. In addition, random placement and replacement improve the observed WCET from 6% to 19% w.r.t. a Least Recently Used policy.

4.1 Relative Sensitivity Based Emulation

This paper presents an FPGA implementation of a probabilistically analyzable cache inspired by the simulation work presented in Kosmidis *et al.* (2013). In this paper we have kept the same approach for the cache behaviour:

1. The cache uses a random replacement policy
2. The cache uses a parametric random placement policy based on a hash function
3. The cache placement is deterministic for each benchmark execution, but randomized across executions
4. We measure end-to-end execution time for a series of benchmarks

4.2 High-Level Fault Model

4.3 Sequential Circuits Fault model

We implemented an instruction and data cache for the Ion MIPS32 processor Open-core (accessed 2015). A completely novel, configurable cache design was implemented in VHDL and

integrated with the Ion core. The cache is completely configurable (bus width, size, block size, policies, etc.) with VHDL generics and could be easily ported to other processor designs. Figure 4.1 shows the main components of our design:

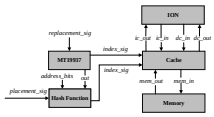


Figure 4.1 Structure of the proposed cache.

1. The cache block contains the cache memory proper, as well as the logic to manage the replacement policy (random and least-recently-used)

2. A hash function block that operates on the index signal to the cache, randomizing the mapping between memory blocks and cache blocks

3. A pseudo-random number generator (MT19937)

4. The Ion core, which provides a MIPS32 ISA and controls the whole system

Our cache has three fundamentally novel features that enable probabilistic timing analysis:

1. A random **placement** policy which uses a parametric hash function to shuffle the initial placement of blocks in the cache memory

2. A random **replacement** policy that uses high-quality random numbers to provide statistically-verifiable guarantees that replacement events are uniformly distributed among the available cache blocks

3. A high-quality pseudo-random number generation, with an extremely long period, to generate random bits for the implementation of the cache random policies

In our cache design, we used the Mersenne Twister algorithm to generate random numbers. In particular, we used the MT19937 algorithm, which is considered as a good hardware solution for a random number generation Matsumoto *et al.* (1998). MT19937 provides a uniform pseudo number pattern with a period of $2^{19937-1}$, with a width of 32 or 54 bits. We used the OpenCores implementation of MT19937 Opencores (accessed 2015). The synthesis report shows that the maximum clock frequency the design can achieved is 147.016 MHz, with a throughput of 30 Msamples per second.

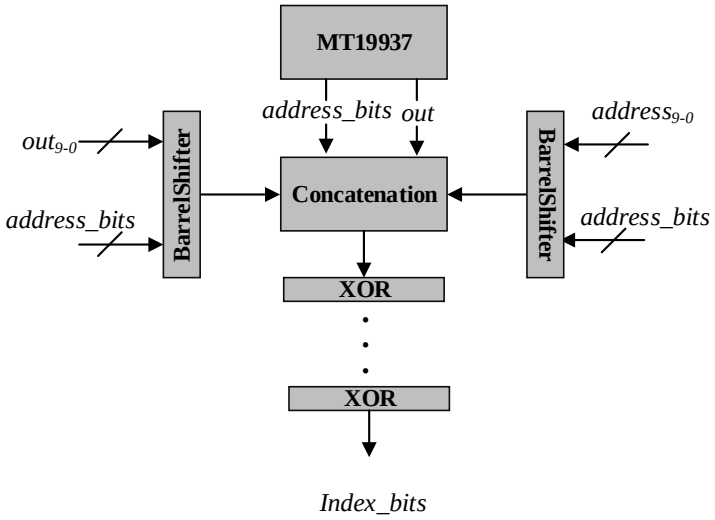


Figure 4.2 The hash function uses a random number, the address bits, and four XOR stages to produce a random placement.

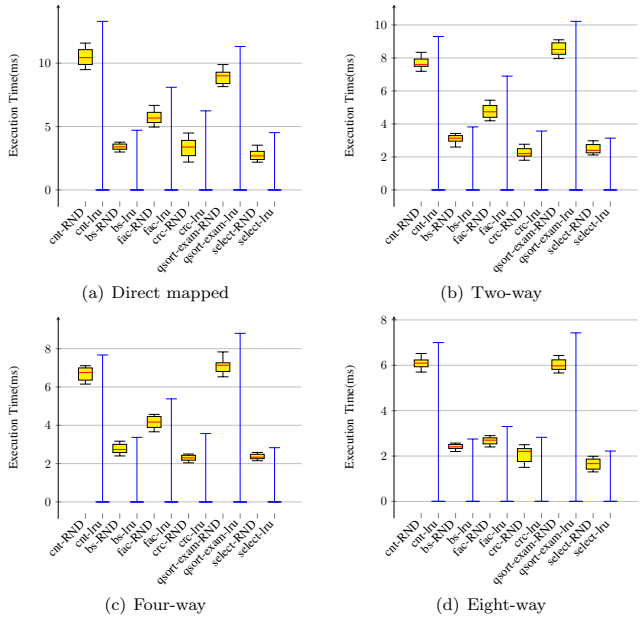
The idea of using a parametric hash function for the implementation of random placement was given by Kosmidis *et al.* (2013). This design is remodelled for this work, replacing their Multiply With Carry (MWC) random number generator with the MT19937, increasing the quality of the random numbers as well as the period. The redesign was driven by the fact that MWC does not pass some statistical normality tests Bandyopadhyay & Bhattacharya (2015), and its period might be insufficient for long running applications Goresky & Klapper (2003).

Standard placement assigns sets to cache lines based on the index bits of the memory address. If the placement policy assigns two memory addresses to the same cache set, they will systematically be in conflict. To deal with this deterministic nature while randomizing the timing

behaviour of the placement policy, we use a parametric hash function with a random number as an input. A random number provides a unique and constant cache set mapping for each address. If the random number changes, the cache set in which the address is mapped changes. By changing random number only at a new execution, programs can be analyzed with end-to-end runs assuming that the cache is initially empty. Figure 4.2 shows the structure of the hash function.

Table 4.1 Resource utilization and overhead (Virtex-5)

	LRU	RND			Overhead
		Cache	Hash	MT19937	
LUT Flip Flop	1904	1792	656	117	34.7%
Slice LUTs	6026	5637	660	419	11.6%



(a) Direct mapped (b) Two-way

(c) Four-way (d) Eight-way

Figure 4.3 Execution Time Measurement

The architecture used in our experiments is the OpenCores Ion MIPS32 processor. We integrated both an I-cache and a D-cache, and we implemented the whole system on the Xilinx

ML505 FPGA evaluation board, using the XC5VLX110T chip using Xilinx ISE-14.4 and ModelSim 10.1.a. We used two separate 4 kB cache memories for data and instructions, both with a 32-byte line size. To evaluate our design, we used Mälardalen real-time benchmark Gustafsson *et al.* (2010) suite. We selected six benchmarks: *cnt, bs, fac, crc, qsort-exam and select*. These benchmarks use arrays and matrices, and have nested loops structures which are ideal to test our design Amoset *et al.* (1991). We omitted those benchmarks using external libraries and unstructured code to simplify the software implementation and data collection.

Each benchmark was run on multiple cache configurations profiles, and we derived its execution time profile using MBPTA, with 30 runs per profile to approximate a normal error distribution. To show that our cache generates identically distributed execution times (as required for PTA), we used the Kolmogorov-Smirnov test Boslaugh & Watters (2008), which shows that the null hypothesis (the data are normally distributed) cannot be rejected for all benchmarks at the 5% confidence level ($p > 0.062$). We compared our results (RND) with a standard Least-Recently-Used (LRU) cache policy implementation.

Figures 4.3 show the timing distributions for all benchmarks on our cache from direct-mapped to 8-way associative, respectively. As an added advantage, our random cache shows a 19% improvement in worst case execution time w.r.t to LRU for a direct-mapped cache, 11% for 2-way cache, 8% for a 4-way cache, and 6% for an 8-way cache. As expected, LRU gets closer to RND as the number of ways increases: the number of conflict miss is greatly reduced by additional ways.

4.4 Conclusion

In this paper, we present the RTL model of a randomized L1 data and instruction cache. This cache uses a high-quality random number generator for random placement and replacement. Random placement is obtained with a parametric hash function that shuffles the association between memory addresses and cache blocks. The cache is integrated with the Ion MIPS32 processor, and verified to generate independent and identically distributed timing events, such that

Measurement-Based Probabilistic Timing Analysis is possible (MBPTA). We test our cache and MBPTA approach on a variety of benchmark from the Mälardalen benchmark suite and show a noticeable improvement (5-15%) in terms of measured Worst Case Execution Time (WCET) as well as enabling the identification of safe probabilistic WCET (pWCET) bounds. Future work will consider the implementation of shared randomized caches for multi-core architectures.

CHAPTER 5

APPENDIX EXAMPLE

5.1 First section of the appendix

5.1.1 Figures in annexes

BIBLIOGRAPHY

Abella, J., Hardy, D., Puaut, I., Quinones, E. & Cazorla, F. J. (2014). On the comparison of deterministic and probabilistic WCET estimation techniques. *Real-Time Systems (ECRTS), 2014 26th Euromicro Conference on*, pp. 266–275.

Alderighi, M., Casini, F., D'Angelo, S., Mancini, M., Codinachs, D., Pastore, S., Sorrenti, G., Sterpone, L., Weigand, R. & Violante, M. (2008, Sept). Robustness analysis of soft error accumulation in SRAM-FPGAs using FLIPPER and STAR/RoRA. *Radiation and Its Effects on Components and Systems (RADECS), 2008 European Conference on*, pp. 157-161. doi: 10.1109/RADECS.2008.5782703.

Altmeyer, S., Davis, R. et al. (2014). On the correctness, optimality and precision of static probabilistic timing analysis. *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pp. 1–6.

Altmeyer, S., Cucu-Grosjean, L. & Davis, R. I. (2015). Static Probabilistic Timing Analysis for Real-time Systems Using Random Replacement Caches. *Real-Time Syst.*, 51(1), 77–123. doi: 10.1007/s11241-014-9218-4.

Amoset, F. et al. (1991). Worst-case Execution Time Analysis for a Java Processor. *Journal of Algorithms*, 12(4), 685-6-99.

Anantaraman, A., Seth, K., Patil, K., Rotenberg, E. & Mueller, F. (2003). Virtual simple architecture (VISA): exceeding the complexity limit in safe real-time systems. *ACM SIGARCH Computer Architecture News*, 31(2), 350–361.

Angermeier, J., Ziener, D., Glas, M. & Teich, J. (2011). Runtime stress-aware replica placement on reconfigurable devices under safety constraints. *Field-Programmable Technology (FPT), 2011 International Conference on*, pp. 1–6.

Anwar, h., Chen, C. & Beltrame, G. (2015). A Probabilistically Analysable Cache Implementation on FPGA. *New Circuit and System (NEWCAS), 2015 15th International Conference on*.

ARM-Research. (accessed 2015). Available: http://www.arm.com/products/processors/cortex-m.

Bandyopadhyay, S. & Bhattacharya, R. (2015). *Discrete and Continuous Simulation: Theory and Practice*. CRC Press.

Bate, I., Conmy, P., Kelly, T. & McDermid, J. (2001). Use of modern processors in safety-critical applications. *The Computer Journal*, 44(6), 531–543.

Bellato, M., Bernardi, P., Bortolato, D., Candelori, A., Ceschia, M., Paccagnella, A., Rebaudengo, M., Sonza Reorda, M., Violante, M. & Zambolin, P. (2004). Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA. *Proceedings of the conference on Design, automation and test in Europe-Volume 1*, pp. 10584.

Berg, C., Engblom, J. & Wilhelm, R. (2004). Requirements for and design of a processor with predictable timing. *Design of Systems with Predictable Behaviour*.

Berg, M., Poivey, C., Petrick, D., Espinosa, D., Lesea, A., LaBel, K., Friendlich, M., Kim, H. & Phan, A. (2008). Effectiveness of internal versus external SEU scrubbing mitigation strategies in a Xilinx FPGA: Design, test, and analysis. *IEEE Transactions on Nuclear Science*, 4(55), 2259–2266.

Bernat, G., Colin, A. & Petters, S. M. (2002). WCET analysis of probabilistic hard real-time systems. *Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE*, pp. 279–288.

Bolchini, C., Miele, A. & Santambrogio, M. D. (2007). TMR and Partial Dynamic Reconfiguration to mitigate SEU faults in FPGAs. *Defect and Fault-Tolerance in VLSI Systems, 2007. DFT'07. 22nd IEEE International Symposium on*, pp. 87–95.

Boslaugh, S. & Watters, P. A. (2008). *Statistics in a nutshell - a desktop quick reference.* O'Reilly.

Cazorla, F. J. et al. (2013). PROARTIS: Probabilistically Analyzable Real-Time Systems. *ACM Trans. Embed. Comput. Syst.*, 12(2s), 94:1–94:26.

Conti, M., Caldari, M., Vece, G. B., Orcioni, S. & Turchetti, C. (2004). Performance analysis of different arbitration algorithms of the AMBA AHB bus. *Proceedings of the 41st annual Design Automation Conference*, pp. 618–621.

Cucu, G. et al. (2012). Measurement-based probabilistic timing analysis for multi-path programs. *Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on*, pp. 91–101.

David, L. & Puaut, I. (2004). Static Determination of Probabilistic Execution Times. *Proceedings of the 16th Euromicro Conference on Real-Time Systems*, (ECRTS '04), 223–230.

Davis, R., Santinelli, L., Altmeyer, S., Maiza, C., Cucu-Grosjean, L. et al. (2013). Analysis of probabilistic cache related pre-emption delays. *Real-Time Systems (ECRTS), 2013 25th Euromicro Conference on*, pp. 168–179.

Delvai, M., Huber, W., Puschner, P. & Steininger, A. (2003). Processor support for temporal predictability-the SPEAR design example. *Real-Time Systems, 2003. Proceedings. 15th Euromicro Conference on*, pp. 169–176.

Edwards, S. A. & Lee, E. A. (2007). The case for the precision timed (PRET) machine. *Proceedings of the 44th annual Design Automation Conference*, pp. 264–265.

Francisco, C. J. (accessed 2015). Available: http://people.ac.upc.edu/fcazorla/.

Glein, R., Schmidt, B., Rittner, F., Teich, J. & Ziener, D. (2014). A Self-Adaptive SEU Mitigation System for FPGAs with an Internal Block RAM Radiation Particle Sensor. *Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on*, pp. 251–258.

González, A., Valero, M., Topham, N. & Parcerisa, J. M. (1997). Eliminating Cache Conflict Misses Through XOR-based Placement Functions. *Proceedings of the 11th International Conference on Supercomputing*, (ICS '97), 76–83.

Goresky, M. & Klapper, A. (2003). Efficient Multiply-with-carry Random Number Generators with Maximal Period. *ACM Trans. Model. Comput. Simul.*, 13(4), 310–321.

Griffin, D., Lesage, B., Burns, A. & Davis, R. I. (2014). Static Probabilistic Timing Analysis of Random Replacement Caches Using Lossy Compression. *Proceedings of the 22Nd International Conference on Real-Time Networks and Systems*, (RTNS '14), 289:289–289:298. doi: 10.1145/2659787.2659809.

Gumbel, E. J. & Lieblein, J. (1954). *Statistical theory of extreme values and some practical applications: a series of lectures*. US Government Printing Office Washington.

Gupta, S., Ansari, A., Feng, S. & Mahlke, S. (2009). Adaptive online testing for efficient hard fault detection. *2009 IEEE Int. Conf. Comput. Des.*, pp. 343–349. doi: 10.1109/ICCD.2009.5413132.

Gustafsson, J., Betts, A., Ermedahl, A. & Lisper, B. (2010). The Mälardalen WCET Benchmarks – Past, Present and Future. *WCET2010*, pp. 137–147.

Hansen, J., Hissam, S. A. & Moreno, G. A. (2009). Statistical-based wcet estimation and validation. *Proceedings of the 9th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis*.

Hauck, S. & DeHon, A. (2010). *Reconfigurable computing: the theory and practice of FPGA-based computation*. Morgan Kaufmann.

Heckmann, R., Langenbach, M., Thesing, S. & Wilhelm, R. (2003). The influence of processor architecture on the design and the results of WCET tools. *Proceedings of the IEEE*, 91(7), 1038–1054.

Heiner, J., Sellers, B., Wirthlin, M. & Kalb, J. (2009). FPGA partial reconfiguration via configuration scrubbing. *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pp. 99–104.

Herrera-Alzu, I. & Lopez-Vallejo, M. (2013). Design techniques for Xilinx Virtex FPGA configuration memory scrubbers. *Nuclear Science, IEEE Transactions on*, 60(1), 376–385.

Hobeika, C., Pichette, S., Ghodbane, A., Thibeault, C., Audet, Y., Boland, J.-F. & Saad, M. (2013). Flight control fault models based on SEU emulation. *SAE International Journal of Aerospace*, 6(2013-01-2246), 643–649.

Hobeika, C., Pichette, S., Leonard, M., Thibeault, C., Boland, J.-F. & Audet, Y. (2014). Multi-abstraction level signature generation and comparison based on radiation single event

upset. *On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International*, pp. 212–215.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301), 13–30.

Jacobs, A., Cieslewski, G., George, A. D., Gordon-Ross, A. & Lam, H. (2012). Reconfigurable fault tolerance: A comprehensive framework for reliable and adaptive FPGA-based space computing. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 5(4), 21.

Jalle, J., Abella, J., Quinones, E., Fossati, L., Zulianello, M. & Cazorla, F. J. (2014a). AHRB: A high-performance time-composable AMBA AHB bus. *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2014 IEEE 20th*, pp. 225–236.

Jalle, J., Kosmidis, L., Abella, J., Quiñones, E. & Cazorla, F. J. (2014b). Bus Designs for Time-probabilistic Multicore Processors. *Proceedings of the Conference on Design, Automation & Test in Europe*, (DATE '14), 50:1–50:6. Consulted at http://dl.acm.org/citation.cfm?id=2616606.2616668.

Kirner, R. & Puschner, P. (2010). Time-predictable computing. In *Software Technologies for Embedded and Ubiquitous Systems* (pp. 23–34). Springer.

Kosmidis, L., Abella, J., Wartel, F., Quinones, E., Colin, A. & Cazorla, F. J. (2014a). PUB: path upper-bounding for measurement-based probabilistic timing analysis. *Real-Time Systems (ECRTS), 2014 26th Euromicro Conference on*, pp. 276–287.

Kosmidis, L., Quinones, E., Abella, J., Vardanega, T., Broster, I. & Cazorla, F. J. (2014b). Measurement-based probabilistic timing analysis and its impact on processor architecture. *Digital System Design (DSD), 2014 17th Euromicro Conference on*, pp. 401–410.

Kosmidis, L. et al. (2013). A Cache Design for Probabilistically Analysable Real-time Systems. *Proceedings of the Conference on Design, Automation and Test in Europe*, (DATE '13), 513–518.

Lahiri, K., Raghunathan, A. & Lakshminarayana, G. (2001). LOTTERYBUS: A new high-performance communication architecture for system-on-chip designs. *Proceedings of the 38th annual Design Automation Conference*, pp. 15–20.

Lamport, L. (1994). *LaTeX: A Document Preparation System* (ed. 2nd). Reading, Massachusetts: Addison-Wesley.

Lesea, A., Drimer, S., Fabula, J. J., Carmichael, C. & Alfke, P. (2005). The rosetta experiment: atmospheric soft error rate testing in differing technology FPGAs. *IEEE Transactions on Device and Materials Reliability*, 5(3), 317–328.

Lyons, R. E. & Vanderkulk, W. (1962). The use of triple-modular redundancy to improve computer reliability. *IBM Journal of Research and Development*, 6(2), 200–209.

Malik, S., Martonosi, M. & Li, Y.-T. S. (1997). Static Timing Analysis of Embedded Software. *Proceedings of the 34th Annual Design Automation Conference*, (DAC '97), 147–152. doi: 10.1145/266021.266052.

Matsumoto, M. et al. (1998). Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator. *ACM Trans. Model. Comput. Simul.*, 8(1), 3–30.

Mezzetti, E., Ziccardi, M., Vardanega, T., Abella, J., Quiñones, E. & Cazorla, F. J. (2015). Randomized Caches Can Be Pretty Useful to Hard Real-Time Systems. *Leibniz Transactions on Embedded Systems*, 2(1), 01–1.

Open-core. (accessed 2015). Available: https://github.com/jaruiz/ION.

Opencores. (accessed 2015). Available: http://opencores.org/project/complex-gaussian-pseudo-random-number-generator.

Ostler, P. S., Caffrey, M. P., Gibelyou, D. S., Graham, P. S., Morgan, K. S., Pratt, B. H., Quinn, H. M. & Wirthlin, M. J. (2009). SRAM FPGA reliability analysis for harsh radiation environments. *Nuclear Science, IEEE Transactions on*, 56(6), 3519–3526.

Petersen, E. (2011). *Single event effects in aerospace*.

Polito. (accessed 2017). Available: http://www.cad.polito.it/downloads/tools/itc99.html.

Quinn, H. & Wirthlin, M. (2015). Validation techniques for fault emulation of SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*, 62(4), 1487–1500.

Quinn, H., Graham, P., Morgan, K., Baker, Z., Caffrey, M., Smith, D. & Bell, R. (2012). On-orbit results for the Xilinx Virtex-4 FPGA. *2012 IEEE Radiation Effects Data Workshop*, pp. 1–8.

Quinn, H., Robinson, W. H., Rech, P., Aguirre, M., Barnard, A., Desogus, M., Entrena, L., Garcia-Valderas, M., Guertin, S. M., Kaeli, D. et al. (2015). Using benchmarks for radiation testing of microprocessors and FPGAs. *IEEE Transactions on Nuclear Science*, 62(6), 2547–2554.

Reineke, J., Grund, D., Berg, C. & Wilhelm, R. (2007). Timing Predictability of Cache Replacement Policies. *Real-Time Syst.*, 37(2), 99–122.

Research, G. (accessed 2015). Available: http://www.gaisler.com/index.php/products/processors/leon3.

Robache, R., Boland, J.-F., Thibeault, C. & Savaria, Y. (2013). A methodology for system-level fault injection based on gate-level faulty behavior. *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International*, pp. 1–4.

Rockett, L., Patel, D., Danziger, S., Cronquist, B. & Wang, J. (2007). Radiation hardened FPGA technology for space applications. *Aerospace Conference, 2007 IEEE*, pp. 1–7.

Santinelli, L. & Cucu-Grosjean, L. (2011). Toward Probabilistic Real-time Calculus. *SIGBED Rev.*, 8(1), 54–61. doi: 10.1145/1967021.1967028.

Schoeberl, M. (2009a). Time-predictable Computer Architecture. *EURASIP J. Embedded Syst.*, 2009, 2:1–2:17. doi: 10.1155/2009/758480.

Schoeberl, M. (2009b). Time-predictable computer architecture. *EURASIP Journal on Embedded Systems*, 2009, 2.

Schoeberl, M., Puffitsch, W., Pedersen, R. U. & Huber, B. (2010). Worst-case Execution Time Analysis for a Java Processor. *Softw. Pract. Exper.*, 40(6), 507–542.

Schoeberl, M., Schleuniger, P., Puffitsch, W., Brandner, F., Probst, C. W., Karlsson, S. & Thorn, T. (2011). Towards a time-predictable dual-issue microprocessor: The Patmos approach. *Bringing Theory to Practice: Predictability and Performance in Embedded Systems*, 18, 11–21.

Schoeberl, M., Chong, D. V., Puffitsch, W. & Sparsø, J. (2014). A time-predictable memory network-on-chip. *14th International Workshop on Worst-Case Execution Time Analysis*, pp. 53.

Seznec, A. (1993). A Case for Two-way Skewed-associative Caches. *Proceedings of the 20th Annual International Symposium on Computer Architecture*, (ISCA '93), 169–178.

Souari, A., Thibeault, C., Blaquière, Y. & Velazco, R. (2015). Optimization of SEU emulation on SRAM FPGAs based on sensitiveness analysis. *On-Line Testing Symposium (IOLTS), 2015 IEEE 21st International*, pp. 36–39.

Stroock, D. W. (2012). *An introduction to the theory of large deviations*. Springer Science & Business Media.

Thiele, L. & Wilhelm, R. (2004). Design for timing predictability. *Real-Time Systems*, 28(2-3), 157–177.

Tobiska, W. K., Atwell, W., Beck, P., Benton, E., Copeland, K., Dyer, C., Gersey, B., Getley, I., Hands, A., Holland, M., Hong, S., Hwang, J., Jones, B., Malone, K., Meier, M. M., Mertens, C., Phillips, T., Ryden, K., Schwadron, N., Wender, S. A., Wilkins, R. & Xapsos, M. A. (2015). Advances in Atmospheric Radiation Measurements and Modeling Needed to Improve Air Safety. *Space Weather*, 13(4), 202–210. doi: 10.1002/2015SW001169. 2015SW001169.

Topham, N. & González, A. (1999). Randomized Cache Placement for Eliminating Conflicts. *IEEE Trans. Comput.*, 48(2), 185–192.

Udipi, A. N., Muralimanohar, N. & Balasubramonian, R. (2010). Towards scalable, energy-efficient, bus-based on-chip networks. *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pp. 1–12.

Violante, M., Sterpone, L., Ceschia, M., Bortolato, D., Bernardi, P., Reorda, M. S. & Paccagnella, A. (2004). Simulation-based analysis of SEU effects in SRAM-based FPGAs. *IEEE Transactions on Nuclear science*, 51(6), 3354–3359.

Whitham, J. (2008). *Real-time processor architectures for worst case execution time reduction*. (Ph.D. thesis, University of York).

Wilhelm, R. et al. (2008a). The Worst-case Execution-time Problem - Overview of Methods and Survey of Tools. *ACM Trans. Embed. Comput. Syst.*, 7(3), 36:1–36:53.

Wilhelm, R., Engblom, J., Ermedahl, A., Holsti, N., Thesing, S., Whalley, D., Bernat, G., Ferdinand, C., Heckmann, R., Mitra, T. et al. (2008b). The worst-case execution-time problem—overview of methods and survey of tools. *ACM Transactions on Embedded Computing Systems (TECS)*, 7(3), 36.

Wirthlin, M., Takai, H. & Harding, A. (2014). Soft error rate estimations of the Kintex-7 FPGA within the ATLAS Liquid Argon (LAr) Calorimeter. *Journal of Instrumentation*, 9(01), C01025.

Xilinx. (accessed 2017a). Available: https://www.xilinx.com/xapp1073.

Xilinx. (accessed 2017b). Available: http://www.xilinx.com.