ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

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

HASSAN ANWAR
DÉPARTEMENT DE GÉNIE ELECTRIQUE
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

RESEARCH PROPOSAL
SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF PHILOSOPHIÆ DOCTOR
(GÉNIE ELECTRIQUE)
(DGA-1032)
DECEMBER 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

UNIVERSITÉ DU QUÉBEC

title of this research proposal :

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

submitted by : ANWAR Hassan
in the context of the : comprehensive examination
to the committee :

THIEBEAULT Claude, Ph.D., advisor
M. SAVARIA Yvon, Ph.D., membre et codirecteur de recherche

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 inflight 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.

TABLE OF CONTENTS

ABSTRACT . . . iii

TABLE OF CONTENTS . . . v

LIST OF FIGURES . . . vii

LIST OF ABBREVIATIONS . . . viii

CHAPTER 1 INTRODUCTION . . . 1
- 1.1 Context and Motivation . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Research Objectives . . . 2
- 1.4 Novelty and Impact . . . 4

CHAPTER 2 LITERATURE REVIEW . . . 5
- 2.1 Radiation Environment . . . 5
- 2.2 Radiation Effects on FPGA . . . 6
- 2.3 Neutron Effects on Electronics . . . 7
- 2.4 Fault Injection . . . 7
  - 2.4.1 Simulation . . . 8
  - 2.4.2 Emulation . . . 9
  - 2.4.3 Radiation testing . . . 10
- 2.5 Fault Models . . . 11
  - 2.5.1 Stuck-at Fault Model . . . 11
  - 2.5.2 Functional Fault Model . . . 11
  - 2.5.3 Open Fault . . . 11
  - 2.5.4 Stuck-On and Stuck-Off . . . 11
- 2.6 Faults Behavioural Modeling . . . 11
- 2.7 Combinational Testing Vs. Sequential Testing . . . 15
  - 2.7.1 Built-In Self-Test (BIST) . . . 15
  - 2.7.2 All Test Pattern Generator (ATPG) . . . 15
  - 2.7.3 Scan Chain Testing . . . 15
- 2.8 Conclusion . . . 15

CHAPTER 3 PROPOSED APPROACH 16

3.1 Research Axis 1: Measuring Sequential Circuit Testing and Reliability 16

3.1.1 Emulation Environment and Framework 16

3.1.2 Sequential Circuit Fault Injection Mechanism 16

3.1.3 Sequential Circuit Test Generation 16

3.2 Research Axis 2: Modeling of Sequential Circuit 16

3.2.1 Modeling of Sequential Circuit with Model Checking 16

3.2.2 Modeling of Sequential Circuit with BDD/ADD 16

3.2.3 Modeling of Sequential Circuit with Monte-carlo Techniques 16

3.2.4 Modeling of Sequential Circuit with Markovian-Chain Analysis 16

3.3 Research Axis 3: Radiation Bombardment 16

3.4 Optional Research Axis: Fault Mitigation 17

3.5 Project Plan 17

CHAPTER 4 PRELIMINARY RESULTS 19

4.1 Relative Sensitivity Based Emulation 19

4.2 High-Level Fault Model 19

4.3 Sequential Circuits Fault model 19

4.4 Conclusion 22

CHAPTER 5 CONCLUSIONS 24

5.1 Work Breakdown Structure 24

5.2 Timetable 26

REFERENCES 27

LIST OF FIGURES

Figure 1.1 Neutron Flux at 40,000 Feet. 3
Figure 2.1 FPGA Structure and Design Flow 6
Figure 2.2 Upset FPGA configuration bits may change the logic and routing. . . . 8
Figure 4.1 Structure of the proposed cache. 20
Figure 4.2 The hash function uses a random number, the address bits, and four XOR stages to produce a random placement. 21
Figure 4.3 Execution Time Measurement 22
Figure 5.1 Work breakdown structure. 25
Figure 5.2 Timetable. 26

LIST OF ABBREVIATIONS

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

The research project is an EU-Canadian joint research project, which is 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 project 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 designed to maintain performance in composite environment without modifying aircraft aerodynamics. Our contribution to this project — ”the study of CR effects on aircraft electrical systems.” This research work will focus on design and implementation of the FPGA-based platform to help to investigate the effects of cosmic radiation (CR) on embedded electronic system of the aircraft. This project also aims to make the higher-level model that will use to investigate the effects of (CR) on the aircraft flying at the altitude of 40,000 feet. This project helps to find; at higher altitude when aircraft gets more exposure to the radiation; need a way to know early in the embedded electronic design of the aircraft if mitigation strategies are required to deal this higher radiation level.

Space radiation has two preliminary sources — galactic cosmic radiation and solar energetic particles [32]. 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 largely dominated by the proton. The consequences of these radiations on human health had been studied extensively at national and international levels. A global framework is also available for the addressing of these radiation issues on health particularly for the frequent flyer, e.g., air-crew. Most of the efforts done so far are either on the monitoring, modeling, and measurements of the radiation and improved the air-safety standards. The space radiation is an unavoidable space weather phenomena. The impact and consequences of the high-energy particles and thermalized neutrons on 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. In future, the FPGAs will replace the deterministic computer architecture platform provide more flexibility to flight operations. In FPGAs the configuration bits of the configuration memory that control the resources, user logic, routing resources, LUTs, CLBs, BRAM, DSP, and IOB blocks. If ion hits the FPGA, it can affect the memory resources that lead towards the fault, which may result in a failure. Before need to know to apply the mitigation techniques early in the embedded electronic design, we need to make the higher-level fault model of the systems that facilitate without going into the detailed simulation get the faulty behavior of the component at high-level.

1.2 Problem Statement

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 [19]. Neutron with energy greater than 10 MeV carries sufficient energy to cause single-event effects in SRAM-based FPGAs. An intense neutron environment exists at higher altitudes in the atmosphere, 10 km to 40 km. Long-haul aircraft flying at the altitudes of 40,000 feet nearly 12 km at the latitude of 60° as shown in Figure 1.1 under the influence of greatest neutron flux of all flights — approximately 500 times that a ground-based observer in Newyork City [19]. This high-energy neutron passes through the silicon substrate of a device, and if the charge of these particles is sufficient enough to change the state of the configuration memory of the FPGA results in a drastic consequence. In this work, we will mainly focus on defining the pre-certification strategy that before employing the circuit in a robust condition, realistically evaluate the faulty behaviors of the circuit. The study of CR effects on aircraft at high altitude/latitude to be able to decide on the appropriate protection solution.

1.3 Research Objectives

The main objective of this project is to develop a faulty behaviour model for FPGA-based circuits described at a high-level of abstraction.

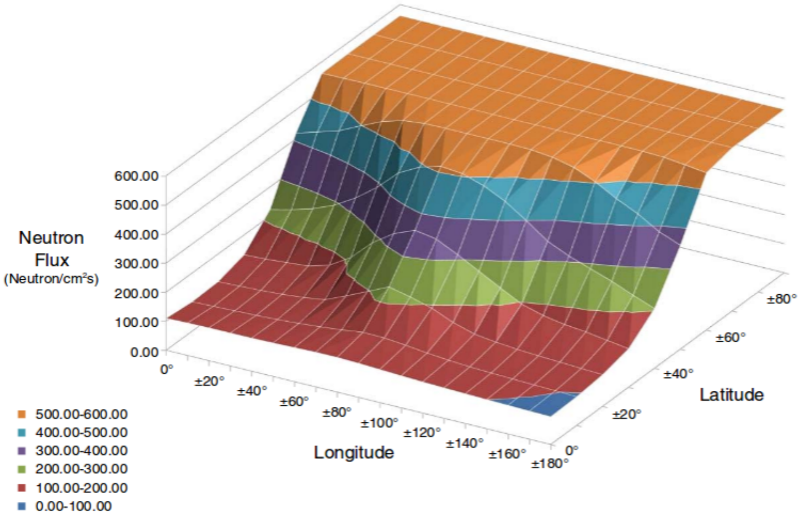


Figure 1.1 Neutron Flux at 40,000 Feet.

By using neural networks, fault behaviour models are developed and their accuracy is validated. 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 goal of this research was to develop an approach for modeling the faulty behaviour of a digital circuit in the presence of cosmic rays.

his ensures that the effect of faulty behaviour 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 more critical parts

The goal of this research was to develop an approach for modeling the faulty behaviour of a digital circuit in the presence of cosmic rays.

To implement effective high-level CR computer model one has to (a) design and implementation of the emulation system, (b) design and implementation of an experimental setup for bombardment, c) feasible for aerospace system, and d) develop a strategy to develop a high-level model from the results (signatures) derived from the emulation and bombardment setup. In the context of the development of a whole simulation methodology including CR environment and CR effects at system and component levels, the objectives of this projects are:

— The goal of this work is to provide a fault injection platform flexible and optimized for FPGA-based systems, allowing emulate configuration faults on SRAM-based FPGAs

— Define CR environment in the context of future aircraft structures at the level of electrical systems

— Study existing databases of effects of CR at electrical systems level

- Complete the analyses of the result of the CR characteristics recorded and derive the consequences in the aircraft embedded system
- Develop the computer model of the CR effects
- Simulate numerically the effect at component and electrical system level
- Develop a strategy for evaluating the robustness of systems against CR
- Propose update of CR requirements for electrical systems
- Proposed the methodology and the models based on data observed in on-board experiments

The main challenges we foresee are:

- Development of the complex circuits and testing under radiation, e.g., soft-error analysis for sequential circuits
- Make a model at higher-level of abstraction from the data extracted at lower level
- Experimental set-up for bombardment

1.4 Novelty and Impact

- The development and implementation of an early validation strategy at higher abstraction level helps to identify at what extent mitigation strategies are required
- Study the system susceptibility under neutron-induced single even effect
- Compare the neutron induced and proton induced errors
- Signature for the sequential circuit
- Computer model to study CR effects at early in the embedded system design

CHAPTER 2

LITERATURE REVIEW

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 2.1. 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)

Table 2.1 Single Event Effects Summary

Single Event Upset (SEU)	corruption of the information stored in a memory element	Memories, latches in logic devices
Multiple Bit Upset (MBU)	several memory elements corrupted by a single strike	Memories, latches in logic devices
Single Event Functional Interrupt (SEFI)	corruption of a data path	Complex devices with built-in state
Single Hard Error (SHE)	unalterable change of state in a memory element	Memories, latches in logic devices
Single Event Transient (SET)	Impulse response of certain amplitude and duration	Analog and Mixed Signal circuits
Single Event Disturb (SED)	Momentary corruption of the information stored in a bit	combinational logic, latches in logic devices
Single Event Latchup (SEL)	high-current conditions	CMOS, BiCMOS devices
Single Event Snapback (SESB)	high-current conditions	N-channel MOSFET, SOI devices
Single Event Burnout (SEB)	Destructive burnout due to high-current conditions	BJT Power MOSFET
Single Event Gate Rupture (SEGR)	Rupture of gate dielectric due to high electrical field conditions	Power MOSFETs

2.2 Radiation 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 [12] involves the creation of a bitstream to load into the device.

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

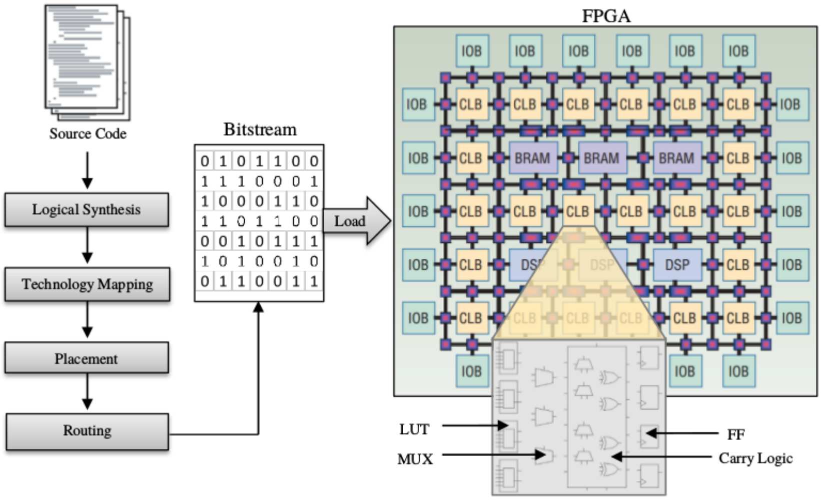


Figure 2.1 FPGA Structure and Design Flow

SRAM-based FPGAs is SEUs within the configuration memory. In such devices, this memory may represent more than 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 [28, 33].

2.3 Neutron Effects on Electronics

2.4 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 [28, 33, 16, 29, 27, 31]. These papers described how they make the faulty behavior of the system to build an accurate representation of the system. The work presented in [27] 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 [16] 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 [31] 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.

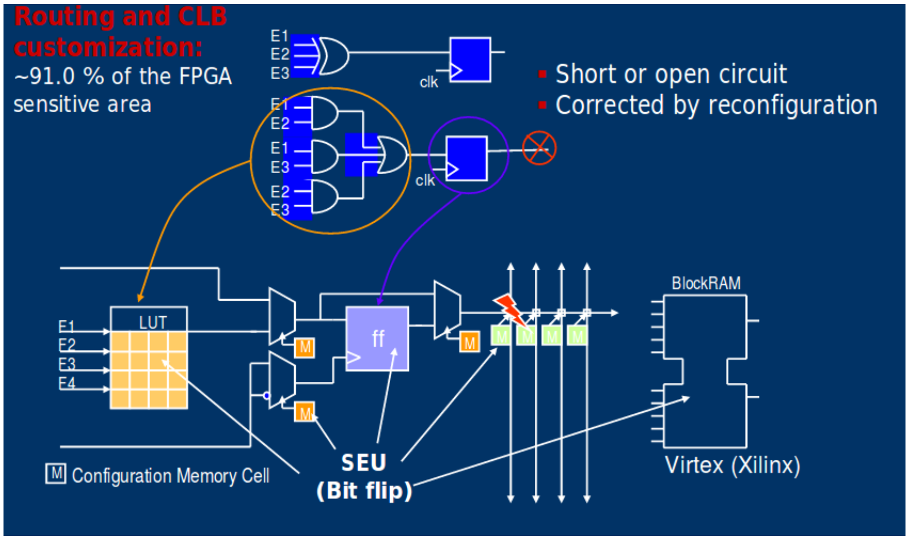

Routing and CLB customization:
~91.0 % of the FPGA sensitive area
E1
E2
E3
clk
Short or open circuit
Corrected by reconfiguration
E1
E2
E1
E3
E2
E3
clk
LUT
E1
E2
E3
E4
M
ff
M
M
M
BlockRAM
Virtex (Xilinx)
Configuration Memory Cell
SEU
(Bit flip)

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

2.4.1 Simulation

The work presented in [29] 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 [16] 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.

2.4.2 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 [15]. The work described 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 [16] and [15] is that; in [15] the authors used the flight control system that is based on a linear plant model. Whereas, in [16]the emulation is performed on the circuits (adder and multiplier). The work presented in [16] 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 [15], 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 [15]. 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 [15]. 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 [31] 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.

2.4.3 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

2.5 Fault Models

2.5.1 Stuck-at Fault Model

2.5.2 Functional Fault Model

2.5.3 Open Fault

2.5.4 Stuck-On and Stuck-Off

2.6 Faults Behavioural Modeling

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 [27]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 [5]. Many techniques have been proposed to provide highly reliable FPGA devices, e.g. radiation-hardened FPGAs [30], 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 [34]. Even on radiation-hardened FPGAs, the SEU rate in a low-earth orbit flight experiment can be up to 16 events per day [26]. 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) [20] and techniques that use scrubbing of the FPGA configuration memory [13]. Also the combination of both techniques has been shown to be able to increase the reliability of FPGA modules significantly [24]. FPGA-based TMR approaches replicate a given module which shall be protected either statically or dynamically [2]. The different granularities of voted replicas are evaluated in [7]. 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 [9], 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 [14]. Scrubbing can be implemented using an internal or external interface as shown in [6]. 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 [35]. 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.

2.7 Combinational Testing Vs. Sequential Testing

2.7.1 Built-In Self-Test (BIST)

2.7.2 All Test Pattern Generator (ATPG)

2.7.3 Scan Chain Testing

2.8 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 Research Axis 1: Measuring Sequential Circuit Testing and Reliability

3.1.1 Emulation Environment and Framework

3.1.2 Sequential Circuit Fault Injection Mechanism

3.1.3 Sequential Circuit Test Generation

3.2 Research Axis 2: Modeling of Sequential Circuit

3.2.1 Modeling of Sequential Circuit with Model Checking

3.2.2 Modeling of Sequential Circuit with BDD/ADD

3.2.3 Modeling of Sequential Circuit with Monte-carlo Techniques

3.2.4 Modeling of Sequential Circuit with Markovian-Chain Analysis

3.3 Research Axis 3: Radiation Bombardment

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 Research Axis: 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 [17] and/or something like the work presented in [17] 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 [17] 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 [25]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 [18]. 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 [22]. 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:

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

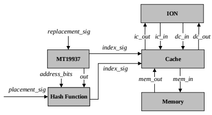


Figure 4.1 Structure of the proposed cache.

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 [21]. MT19937 provides a uniform pseudo number pattern with a period of $2^{19937\text{-}1}$, with a width of 32 or 54 bits. We used the OpenCores implementation of MT19937 [23]. 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.

The idea of using a parametric hash function for the implementation of random placement was given by [18]. 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 [4], and its period might be insufficient for long running applications [10].

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

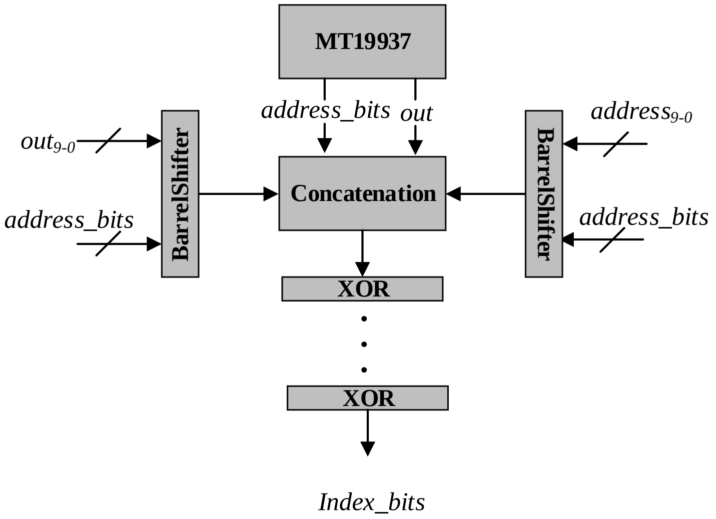


Figure 4.2 The hash function uses a random number, the address bits, and four XOR stages to produce a random placement.

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.

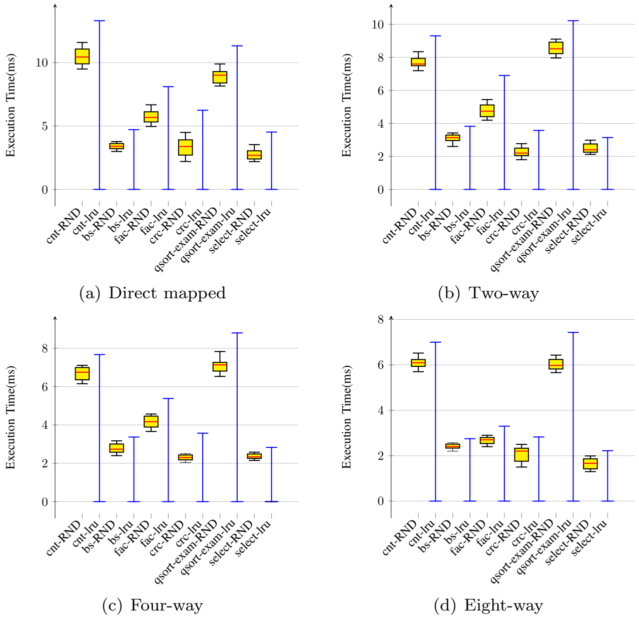
The architecture used in our experiments is the OpenCores Ion MIPS32 processor. We integrated both an I-cache and a D-cache, and we implemented the whole system on the Xilinx ML505 FPGA evaluation board, using the XC5VLX110T chip using Xilinx ISE-14.4 and ModelSim 10.1.a. We used two separate 4 kB cache memories for data and instructions, both with a 32-byte line size. To evaluate our design, we used Mälardalen real-time benchmark [11] 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 [1]. 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 [8], 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-

Table 4.1 Resource utilization and overhead (Virtex-5)

	LRU	RNDRND			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

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

CONCLUSIONS

In this proposal, we have demonstrated our research will be focused on investigation of a design, methodologies, and implementation of a time predictable fault tolerant computing system evaluated by the probabilistic timing analysis (PTA). As our target domain is a real-time industry. We will stress the importance of a worst case execution time (WCET) estimation. Nowadays, the investigations of new timing analysis techniques are an unavoidable need because of the growing complexity of a modern embedded computers and the aerospace industry will especially a benefit from the introduction of a such technologies in terms of reliability and design costs. Our approach will leverage probabilistic approach to enable the timing analysis in computing systems. As a consequence, this research has a potential to make computing systems smarter, more reliable, and easier to design and to program. At the same time, we think that our results will make decisive steps ahead in a fairly unexplored research area - integration of fault-tolerance techniques in time predictable computer architecture.

In our preliminary results, we showed that how probabilistically analysable cache can be integrated in a MIPS processor to make the foundation for a probabilistically analysable computing systems. The research published in [3] proved the effectiveness of a measurement based probabilistic timing analysis (MBPTA). Furthermore, our most recent results demonstrated that probabilistic timing technique is a promising approach for the future timing analysis of a real-time aerospace embedded system.

Through this research, we hope to be able to have an impact on how computer engineers and system designers will think of probabilistic computing in near future, and contribute to create the next generation of a real-time embedded systems for aerospace industry. We will target field-specific international journals such as the “ACM Transactions on Real-time Systems” and the “ACM Transactions on Reconfigurable Technology and Systems”, or conferences including tracks dedicated to the automated design of embedded systems, such as the “Design Automation Conference” and the “Design, Automation & Test in Europe” conference.

5.1 Work Breakdown Structure

Figure 5.1 presents the work breakdown structure (WBS) of our research project. At level 2 of this tree chart, we identify four groups of tasks: the ones related to the acquisition of knowledge; those related to the development of new knowledge; an experimental phase; and, finally, project management tasks.

Knowledge acquisition includes the class work done towards the credit requirements of the PhD program and the review of the scientific literature.

The knowledge extension task can be split along the four research axes defined in Chapter 3. The experimental phase goes from the definition of a test plan to the experimental evaluation of our prototype on a CubeSat platform.

Project management tasks involve the writing of conference and journal articles, as well as the preparation of a thesis and its defense.

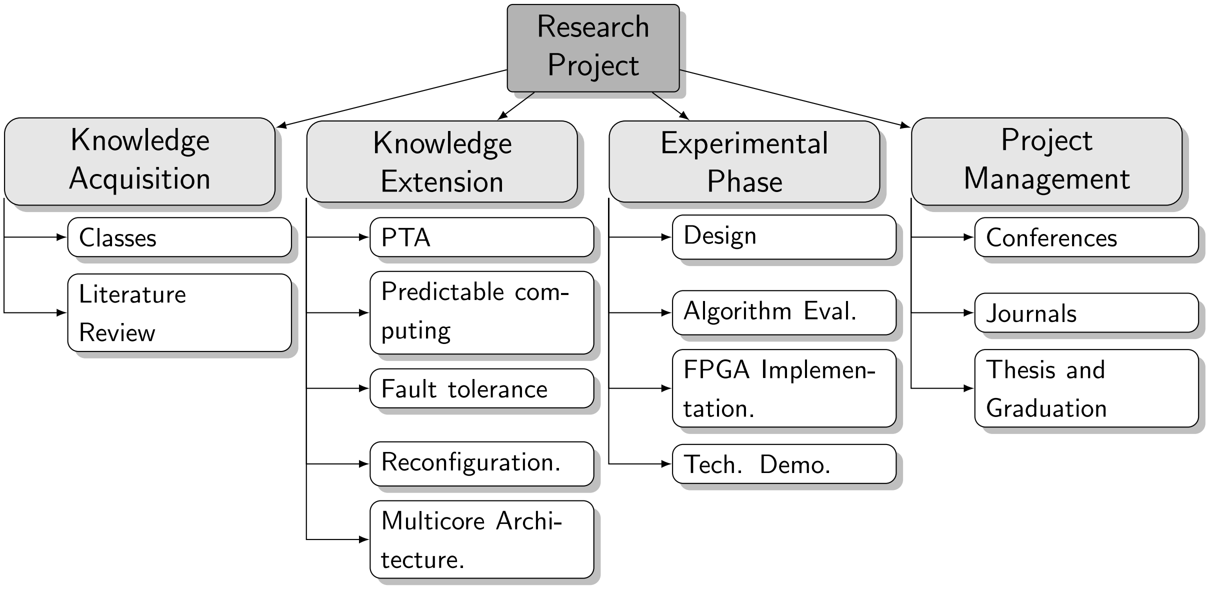


Figure 5.1 Work breakdown structure.

5.2 Timetable

The development of the tasks identified in Chapter 3, and the most important milestones of this project are presented in Figure 5.2. In our intentions, the design of a time predictable computer architecture, the development of novel timing analysis techniques, and the FPGA prototypes implementation will unfold as a series of sequential tasks with relatively small interleaving. Dependability and real-time requirements, on the other hand, should be kept in mind throughout the whole advancement of the project.

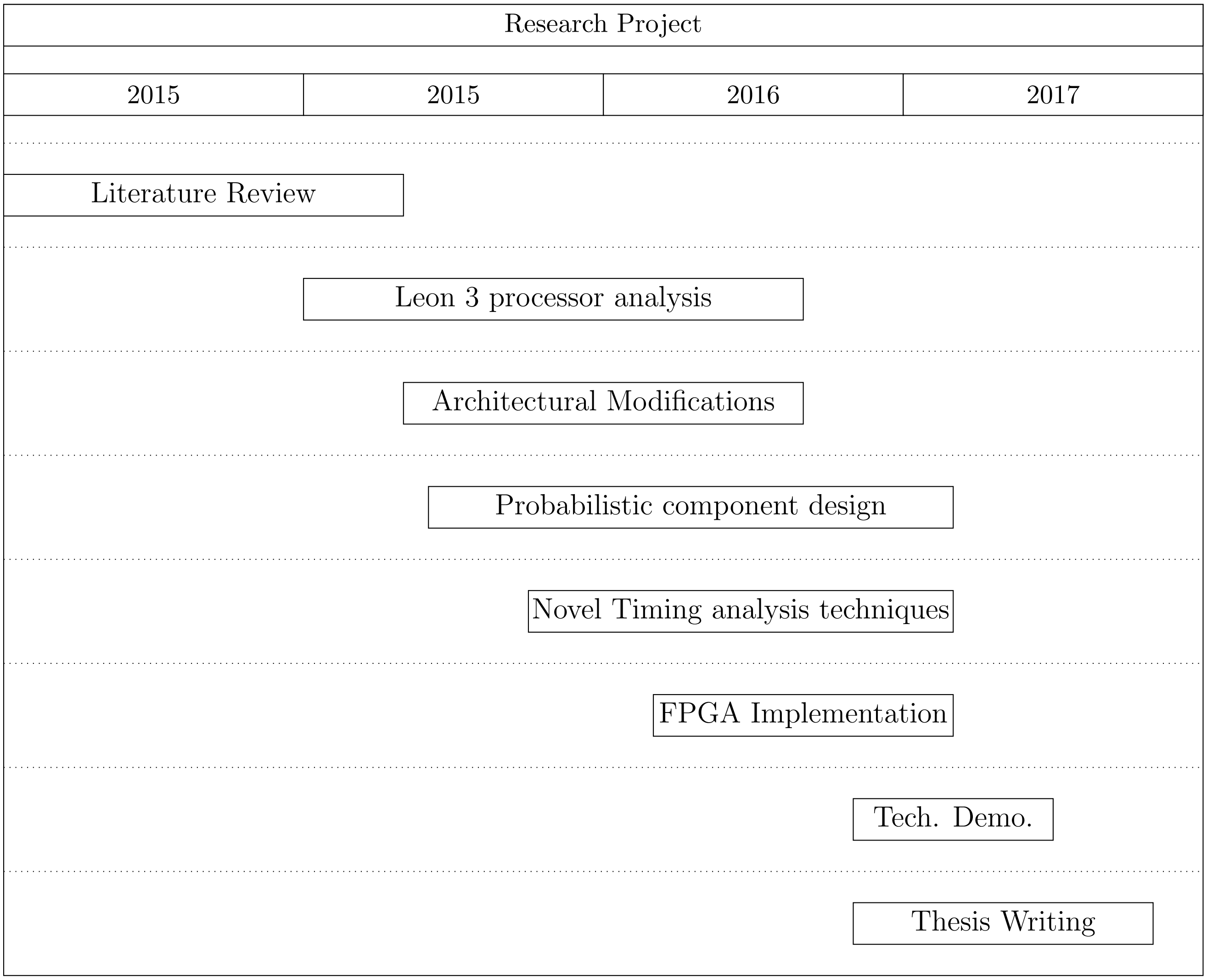


Figure 5.2 Timetable.

REFERENCES

[1] AMOSET, F. *ET AL.* (1991). Worst-case execution time analysis for a java processor. *Journal of Algorithms*, 12, 685–6–99.

[2] ANGERMEIER, J., ZIENER, D., GLAS, M. and TEICH, J. (2011). Runtime stress-aware replica placement on reconfigurable devices under safety constraints. *Field-Programmable Technology (FPT), 2011 International Conference on*. IEEE, 1–6.

[3] ANWAR, H., CHEN, C. and BELTRAME, G. (2015). A probabilistically analysable cache implementation on fpga. *New Circuit and System (NEWCAS), 2015 15th International Conference on*. IEEE.

[4] BANDYOPADHYAY, S. and BHATTACHARYA, R. (2015). *Discrete and Continuous Simulation: Theory and Practice*. CRC Press.

[5] BELLATO, M., BERNARDI, P., BORTOLATO, D., CANDELORI, A., CESCHIA, M., PACCAGNELLA, A., REBAUDENGO, M., SONZA REORDA, M., VIOLANTE, M. and 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*. IEEE Computer Society, 10584.

[6] BERG, M., POIVEY, C., PETRICK, D., ESPINOSA, D., LESEA, A., LABEL, K., FRIENDLICH, M., KIM, H. and 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, 2259–2266.

[7] BOLCHINI, C., MIELE, A. and 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*. IEEE, 87–95.

[8] BOSLAUGH, S. and WATTERS, P. A. (2008). *Statistics in a nutshell - a desktop quick reference*. O'Reilly.

[9] GLEIN, R., SCHMIDT, B., RITTNER, F., TEICH, J. and 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*. IEEE, 251–258.

[10] GORESKY, M. and KLAPPER, A. (2003). Efficient multiply-with-carry random number generators with maximal period. *ACM Trans. Model. Comput. Simul.*, 13, 310–321.

[11] GUSTAFSSON, J., BETTS, A., ERMEDAHL, A. and LISPER, B. (2010). The Mälardalen WCET benchmarks – past, present and future. B. Lisper, editor, *WCET2010*. OCG, Brussels, Belgium, 137–147.

[12] HAUCK, S. and DEHON, A. (2010). *Reconfigurable computing: the theory and practice of FPGA-based computation*, vol. 1. Morgan Kaufmann.

[13] HEINER, J., SELLERS, B., WIRTHLIN, M. and KALB, J. (2009). Fpga partial reconfiguration via configuration scrubbing. *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*. IEEE, 99–104.

[14] HERRERA-ALZU, I. and LOPEZ-VALLEJO, M. (2013). Design techniques for xilinx virtex fpga configuration memory scrubbers. *Nuclear Science, IEEE Transactions on*, 60, 376–385.

[15] HOBEIKA, C., PICHETTE, S., GHODBANE, A., THIBEAULT, C., AUDET, Y., BOLAND, J.-F. and SAAD, M. (2013). Flight control fault models based on seu emulation. *SAE International Journal of Aerospace*, 6, 643–649.

[16] HOBEIKA, C., PICHETTE, S., LEONARD, M., THIBEAULT, C., BOLAND, J.-F. and 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*. IEEE, 212–215.

[17] JACOBS, A., CIESLEWSKI, G., GEORGE, A. D., GORDON-ROSS, A. and 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, 21.

[18] 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.

[19] LESEA, A., DRIMER, S., FABULA, J. J., CARMICHAEL, C. and ALFKE, P. (2005). The rosetta experiment: atmospheric soft error rate testing in differing technology fpgas. *IEEE Transactions on Device and Materials Reliability*, 5, 317–328.

[20] LYONS, R. E. and VANDERKULK, W. (1962). The use of triple-modular redundancy to improve computer reliability. *IBM Journal of Research and Development*, 6, 200–209.

[21] MATSUMOTO, M. *ET AL.* (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8, 3–30.

[22] OPEN-CORE (accessed 2015). Available: https://github.com/jaruiz/ION.

[23] OPENCORES (accessed 2015). Available: http://opencores.org/project/complex-gaussian-pseudo-random-number-generator.

[24] OSTLER, P. S., CAFFREY, M. P., GIBELYOU, D. S., GRAHAM, P. S., MORGAN, K. S., PRATT, B. H., QUINN, H. M. and WIRTHLIN, M. J. (2009). Sram fpga reliability analysis for harsh radiation environments. *Nuclear Science, IEEE Transactions on*, 56, 3519–3526.

[25] POLITO (accessed 2017). Available: http://www.cad.polito.it/downloads/tools/itc99.html.

[26] QUINN, H., GRAHAM, P., MORGAN, K., BAKER, Z., CAFFREY, M., SMITH, D. and BELL, R. (2012). On-orbit results for the xilinx virtex-4 fpga. *2012 IEEE Radiation Effects Data Workshop*. 1–8.

[27] 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, 2547–2554.

[28] QUINN, H. and WIRTHLIN, M. (2015). Validation techniques for fault emulation of sram-based fpgas. *IEEE Transactions on Nuclear Science*, 62, 1487–1500.

[29] ROBACHE, R., BOLAND, J.-F., THIBEAULT, C. and 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*. IEEE, 1–4.

[30] ROCKETT, L., PATEL, D., DANZIGER, S., CRONQUIST, B. and WANG, J. (2007). Radiation hardened fpga technology for space applications. *Aerospace Conference, 2007 IEEE*. IEEE, 1–7.

[31] SOUARI, A., THIBEAULT, C., BLAQUIÈRE, Y. and VELAZCO, R. (2015). Optimization of seu emulation on sram fpgas based on sensitiveness analysis. *On-Line Testing Symposium (IOLTS), 2015 IEEE 21st International*. IEEE, 36–39.

[32] 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. and XAPSOS, M. A. (2015). Advances in atmospheric radiation measurements and modeling needed to improve air safety. *Space Weather*, 13, 202–210. 2015SW001169.

[33] VIOLANTE, M., STERPONE, L., CESCHIA, M., BORTOLATO, D., BERNARDI, P., REORDA, M. S. and PACCAGNELLA, A. (2004). Simulation-based analysis of seu effects in sram-based fpgas. *IEEE Transactions on Nuclear science*, 51, 3354–3359.

[34] WIRTHLIN, M., TAKAI, H. and HARDING, A. (2014). Soft error rate estimations of the kintex-7 fpga within the atlas liquid argon (lar) calorimeter. *Journal of Instrumentation*, 9, C01025.

[35] XILINX (accessed 2017). Available: http://www.xilinx.com.