Single Event Upsets in SRAM FPGA based readout electronics for the Time Projection Chamber in the ALICE experiment

Ketil Røed

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

September 23, 2009

Acknowledgements

My time as a doctoral student has finally come to an end with the submission of this thesis. During the last years I have been given the opportunity to work on the interesting and challenging subject of radiation effects in semiconductor devices. Being part of CERN and the ALICE collaboration has been a door opener to a large group of international experts. This has not only given invaluable input to my thesis but also resulted in many new friends with whom I never would have met otherwise. Even though there have been moments of hard work and lack of motivation, looking back it has still been a great experience.

Many people deserve special thanks for their support and contribution. First of all I would like to thank the Faculty of Engineering at Bergen University College for employing me and giving me this opportunity. The PhD has been carried out in a joint collaboration with the doctoral student program at the Department of Physics and Technology, University of Bergen. Many thanks to my supervisors Kjetil Ullaland at the University of Bergen and Håvard Helstrup and Terje Natås at the Bergen University College.

Dieter Röhrich deserves special attention for his major contributions regardless of not being one of my official supervisors. His experience and network of contacts has been of crucial importance to the progress of my work. Realizing this thesis would have been very difficult without his involvement.

Sharing my time between the two institutions I have to thank Håvard for his continuous effort in making any related administrative aspects of this as transparent as possible. I am also grateful for his positive thinking and support during my PhD thesis work. In particular his encouragement and detailed feedback during the writing phase has been an important contribution to finally completing my thesis.

Already having Kjetil Ullaland as the supervisor of my Master thesis project, I was happy to see him also take on the same responsibility for my PhD. Through our many fruitful discussions he has been a main source of knowledge and inspiration. With his direct approach and know-how he has encouraged me to take responsibility and helped me to stay focused and on track.

Throughout my PhD period I have had the pleasure of working closely with Johan Alme. Sharing not only some of the same tasks, but also a number of offices and Maryland cookies together, there was never a dull moment. He has been an important motivator and friend and I hope that the future will bring new possibilities of working together. I would also like to thank Kenneth Aamodt, Sebastian Bablok, Dominik Fehlker, Kalliopi Kanaki, Dag Toppe Larsen, Matthias Richter, Boris Wagner, and Gaute Øvrebekk for numerous and helpful discussions on topics like ROOT, C++, programming in general, and ALICE physics. Torsten Alt and Gerd Tröger should also be mentioned for their contributions and many interesting discussions. In particular also for the extracurricular activities when visiting Heidelberg. Solfrid Sjåstad Hasund and Rune Fosse are thanked for their contribution to my teaching responsibilities at the Bergen University College. Additionally I would like to thank my other colleagues at the Department of Physics and Technology at the University of Bergen and at the Bergen University College who have contributed in one way or the other.

A special thanks must be given to Henry Tang, Kenneth Rodbell, Conal Murray, Giovanni Fiorenza and their colleagues at the T.J. Watson IBM Research Center in Yorktown, New York, USA. I am grateful to Henry for always giving of his time and being an endless source of information through our many and long discussions. During my 7 months visit in their group I gained a significant amount of knowledge in physics based Monte Carlo simulations and related topics. The Norwegian Research Council should also be thanked for financially supporting this visit through the Leiv Eiriksson mobility programme. Moreover, I have to thank Jennifer Hill for her generous hospitality in addition to the rest of the gang living in Overlook Road. Thank you for making my stay in the US a memorable one.

From the Institute of Experimental Physics SAS, Košice, Slovak Republic and CERN, I have to thank Blahoslav Pastirčák for his generous help and contribution to the Monte Carlo simulations of the radiation environment. Luciano Musa and his group at CERN are also acknowledged as part of the ALICE TPC collaboration and RCU project team in particular. Jon Wikne and Eivind Olsen at the Oslo Cyclotron, University of Oslo, and Alexander Prokofiev at The Svedberg Laboratory, University of Uppsala, should also be thanked for all their help during the many hours of irradiation testing I have participated in over the last years. I would also like to thank my new colleagues at CERN for their understanding during the finalizing part of my PhD thesis.

Special thanks goes to Trud for her support and kindness, and finally I am deeply grateful to my parents for their unconditional help and support during these years.

Ferney-Voltaire, September 2009

Abstract

The front-end electronics of the TPC detector, one of the major detectors of the ALICE experiment at CERN, utilizes an SRAM based FPGA to control the readout of detector data. Compared to traditional ASIC design, an SRAM based FPGA was chosen because it offers the flexibility of in-field programmability. However, when used in radiation exposed environments, FPGAs have shown to be susceptible to radiation induced effects such as single event upsets. A single event upset is induced when a single ionizing particle deposits a sufficient amount of energy to alter the logic state of a memory element. In an SRAM based FPGA the user-programmed functionality depends on the data stored in millions of these memory elements. A single event upset in one or several of these memory cells may result in unexpected and incorrect behaviour. Consequently, for the FPGA in the TPC front-end electronics, this can potentially cause the readout of detector data to temporarily break down. It is therefore important to investigate how radiation induced failures can be reduced or even avoided if possible.

Due to its stochastic nature, a single event upset has to be treated in terms of its probability to occur. This probability is determined by the sensitivity of a memory cell to a specific type of radiation, and further what type of radiation environment it will finally be operated in. Moreover, depending on how a given memory cell is utilized by a system, a single event upset may or may not result in a detectable malfunction of that system.

The main purpose of this thesis has been to investigate these aspects for the SRAM based FPGA in control of data readout in the TPC front-end electronics. By means of Monte Carlo simulations, test procedures and mitigation approaches a major objective has been to qualify this FPGA for reliably operation in the radiation environment produced by particle collisions in the ALICE experiment.

This thesis presents updated Monte Carlo simulation of particle energy spectra and fluences in more precise locations than what has previously been done. Irradiation tests have further been carried out in order to investigate the single event upset sensitivity of the SRAM based FPGA. The results are discussed in light of independent results reported in literature. Combined, Monte Carlo simulation and irradiation test results have been used to predict the single event upset rate expected during operation in the ALICE experiment.

Due to the number of FPGAs utilized in the TPC front-end electronics, single event upsets can be a reliability concern. In order to reduce the probability of system malfunction, a reconfiguration solution was developed that enables the possibility to clear single event upsets in the configuration memory of the FPGA. Irradiation test results show that combined with additional system level mitigation techniques, this reconfiguration solution can be used to finally reduce the functional failure rate of the FPGA.

Because irradiation testing can be time consuming, costly and sometimes even technically difficult, a software based fault injection solution has been implemented without any modification to the existing hardware setup. It provides an alternative and possibly systematic method of testing how a single event upset may impact the operation of the FPGA. Test results show good agreement with comparable irradiation test results.

Finally physics based Monte Carlo simulations are discussed as an additional method to investigate single event upset in memory devices. A general methodology is presented and applied to the specific case study of the TPC front-end electronics FPGA.

Contents

Acknowledgments **i**

Abstract **iii**

1 Introduction **1**

1.1 The ALICE experiment 2

1.1.1 Physics goals 2

1.1.2 Detectors 3

1.2 The TPC front-end electronics 5

1.2.1 Readout Control Unit 5

1.2.2 Choice of FPGA technology 6

1.3 Primary objective and main contributions 8

1.4 Outline 10

2 Radiation effects in the TPC RCU main FPGA **11**

2.1 Single Event Effects 12

2.2 Basic mechanism 14

2.2.1 Charge generation and collection 14

2.2.2 Single event upsets in SRAM memory 15

2.2.3 Single event upsets in the FPGA configuration memory . . . 16

2.2.4 The physics of single event upsets 17

2.3 The TPC radiation environment 20

2.3.1 Particle multiplicity 21

2.3.2 Previous work 22

2.3.3 Updated simulations with new geometry description 22

2.3.4 Discussion 25

2.4 Summary 29

3 RCU radiation tolerant system solution **31**

3.1 Partial reconfiguration 31

3.1.1 Xilinx Virtex-II Pro configuration memory 32
3.1.2 Configuration process 34
3.1.3 Partial reconfiguration on the RCU 37
3.1.4 Limitations of partial reconfiguration 37
3.2 The RCU and reconfiguration network 38
3.2.1 DCS communication and operational modes 39
3.2.2 RCU support FPGA 42
3.3 Measurement of configuration times 45
3.4 Summary 46

4 Accelerated beam testing of the Xilinx Virtex-II Pro 7 and the RCU reconfiguration network 47
4.0.1 Calculating the SEU cross section 47
4.1 Experimental setup 48
4.1.1 Beam line configuration and monitoring 50
4.1.2 Measuring the SEU cross section of the RCU main FPGA . 50
4.1.3 Measuring the mitigation effect of the reconfiguration network 52
4.2 Results 55
4.2.1 SEU cross section 56
4.2.2 The effect of the reconfiguation network under irradiation . . 60
4.2.3 Total ionizing dose 65
4.3 Predicting the SEU rate in the TPC radiation environment 65
4.4 Summary 66

5 Implementing Fault Injection for the RCU main FPGA 69
5.1 Implementation 70
5.1.1 System design considerations 70
5.1.2 Fault injection procedure 71
5.1.3 Software classes 71
5.2 Fault injection case study 75
5.2.1 RCU main FPGA test design 76
5.3 Results 77
5.3.1 Summary and validation results 77
5.3.2 Applications 81
5.4 Discussion and Summary 88

6 Monte Carlo based SEU simulations 91
6.1 General methodology 92
6.1.1 SEU cross section from Monte Carlo simulations 93

6.2 Resolving case study geometry 94
6.2.1 Structural analysis 95
6.2.2 Generic geometry input description 98
6.3 Preparation and setup of simulation tools 99
6.3.1 Fluka specifics 99
6.3.2 Modifications of the SEMM2 model 105
6.3.3 Status 107
6.4 Fluka simulation results 108
6.4.1 Collection Volume variability study 108
6.4.2 Contribution from α-particles and heavy fragments 111
6.4.3 Role of metal interconnect layers 115
6.5 Summary 116

7 Conclusion and outlook 119

A AliRoot simulations of the TPC radiation environment 123
A.1 Previous work 123
A.2 Geometry description 124
A.2.1 Description of front-end cards 124
A.2.2 The RCU scoring region 126
A.2.3 The C++ code of the geometry description 126
A.3 Visual check using energy scoring 130
A.4 Fluence results for the 6 scoring regions 132
A.5 Fluence as a function of energy 138

B Flow diagram of the FRVC procedure 141

C Irradiation test results 143
C.1 SEU cross section results 143
C.1.1 Total dose calculation 146

D Class diagram of fault injection software 147

E SEU Monte Carlo simulation 149
E.1 FPGA geometry analysis 149
E.2 Determining an optimal simulation target area 151
E.3 SEMM2.vBergen setup specifics 155
E.3.1 Tables of simulation parameters 155
E.3.2 Input file example 158

F List of Publications 163

F.1 As main contributor 163

F.2 As collaborator 163

Chapter 1

Introduction

SRAM[1] based Field Programmable Gate Arrays (FPGAs) have become a very attractive alternative in many applications due to the continuous increase in density of user programmable resources and embedded memory. Compared to traditional ASIC[2], design, SRAM based FPGAs offer advantages such as decreased cost and development time. In addition one of the major benefits is considered to be their ability to be programmed in the field. This offers great flexibility as it allows to reprogram already deployed devices with new functionality or improved versions of existing implementations. For complex detector systems like the ALICE[3] experiment where changes may be needed up to the very last minute, reprogammability may prove to be a vital feature. It will also allow continued development to improve functionality after start-up of the experiment. With the use of ASICs, any upgrades would not be possible without the need to replace hardware. Moreover, as the ALICE experiment will be physically inaccessible for most of its operational life time, replacing hardware would require parts of the detector system to be dismantled. The flexibility offered by SRAM FPGAs is therefore a major reason why they were chosen for the project described in this thesis.

A major drawback of SRAM FPGAs is their susceptibility to radiation induced effects [1], in particular single event upsets. Single event upsets are caused by ionizing particles which may deposit enough energy in the device to alter the logic state of a memory element. Since the user-programmed functionality depends on the data stored in millions of these memory elements, a single event upset may give rise to unfavourable effects in the expected functionality.

In the ALICE experiment, beams of particles will be collided at extreme energies to study a state of matter known as quark-gluon plasma [2]. These collisions will

[1] SRAM: static random access memory.

[2] ASIC: Application Specific Integrated Circuit.

[3] ALICE: A Large Ion Collider Experiment. See section 1.1 for further introduction.

generate new particles which can be detected by different sub-detectors of ALICE to determine properties like for instance particle type, energy and momentum. However, particles from these collisions will also pose a reliability risk to the readout electronics of these sub-detectors. For example, in the Time Projection Chamber, which is the main tracking detector of ALICE, the readout of data is controlled by an SRAM based FPGA. If vital functionality should be lost due to a single event upset, large amounts of data may be lost unless the source of the failure can be repaired in time or even avoided in the first place. Due to the dense architecture of the detectors and the fact that the radiation field is dominated by neutrons, physical shielding of the electronics is practically not possible in ALICE. Nor is it wanted as extra material should be avoided if possible in order to minimize the distortion of the physics measurements. The main challenge is therefore to apply the appropriate measures in order to reduce the consequences of such effects. The process or technique applied to reduce the failure probability of a system is often referred to as mitigation of the system.

This thesis is based on the various aspects related to qualifying an SRAM based FPGA for use in the ALICE experiment. The work focuses on investigating the underlying mechanisms responsible for single event upsets in the ALICE environment, the probability of experiencing failures, and a method to repair and test the effects of single event upsets.

1.1 The ALICE experiment

The LHC is a circular accelerator located at CERN[4], the European Organization for Nuclear Research. It has a circumference of about 27 km and located about 100 m below the ground level. With to adjacent beam pipes bunches of particles will travel in opposite directions and collide at dedicated experimental locations. ALICE is one of four main experiments at LHC.

1.1.1 Physics goals

The main purpose of ALICE is to study a state of matter known as quark-gluon plasma, which is believed to have existed soon after the Big Bang. Quarks are bound together into hadrons[5] by the strong force, which is mediated by gluons. As the

[4] CERN: Conseil Européen pour la Recherche Nucléaire.

[5] Hadrons contain two subsets of particles: baryons, made up of 3 quarks, and mesons, made up of a quark-antiquark pair. Common baryons are the protons and neutrons, while pions are typical a example of mesons produced in particle physics experiments.

potential associated with the strong force increases with distance, quarks cannot appear as isolated particles under normal conditions. With the extreme energy density and temperatures that will be achieved in the nucleus-nucleus collisions at the LHC, it is expected that matter will undergo a phase transition into a plasma of unbound quarks and gluons, the quark-gluon plasma. As the quark-gluon plasma expands and cools, it will transform into hadronic matter. The quark-gluon plasma can therefore not be studied directly. Instead a set of observables are identified as indicators of its existence. More details on ALICE physics can be found in [3].

1.1.2 Detectors

The ALICE detector shown in figure 1.1 is optimized for heavy ion collisions. Beams of lead ions and protons will enter from each side of the detector and collide in the interaction point which is at the centre of the detector. The detector has an onion-like structure where the various sub-detectors are arranged in different layers from inside to the outside. Each sub-detector is optimized to study specific properties of the particles produced in the Pb-Pb collisions.

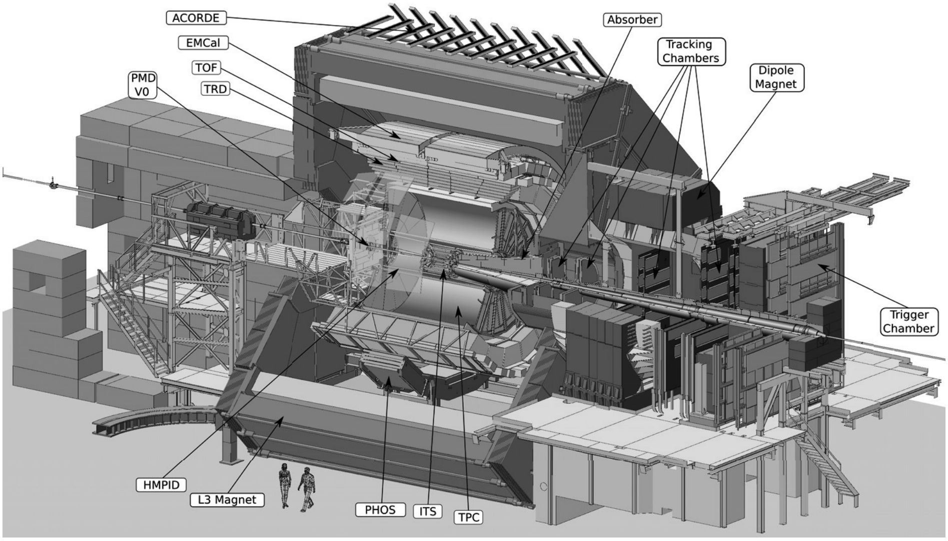


Figure 1.1: Layout of the ALICE detector [2].

The Time Projection Chamber (TPC)

The TPC, shown in figure 1.2, is the main tracking detector in ALICE and is optimized for charged particle momentum measurements. It consists of a cylindrical shaped field cage and has an inner radius of 85 cm, an outer radius of 280 cm, and an overall length along the beam direction of 510 cm. Charged particles produced in the collisions will ionize the gas inside the field cage and electrons will drift in the electric field between the central high voltage electrode and the two end plates. Each of the two end plates is divided in 18 trapezoidal sectors where multi-wire proportional chambers provide the required charge amplification. Each sector is again sub-divided into an inner and outer chamber with slightly different wire geometry spacing and pad sizes. This is done to provide better spatial resolution as the track density is higher closer to the beam line. The full readout plane of the TPC is divided into a total of 570132 pads where each pad corresponds to one data readout channel. As the TPC front-end electronics is located directly behind the two end plates of the field cage, it is expected that radiation induced effects will pose a reliability risk. Thus the radiation environment in this location is further investigated in section 2.3 of this thesis.

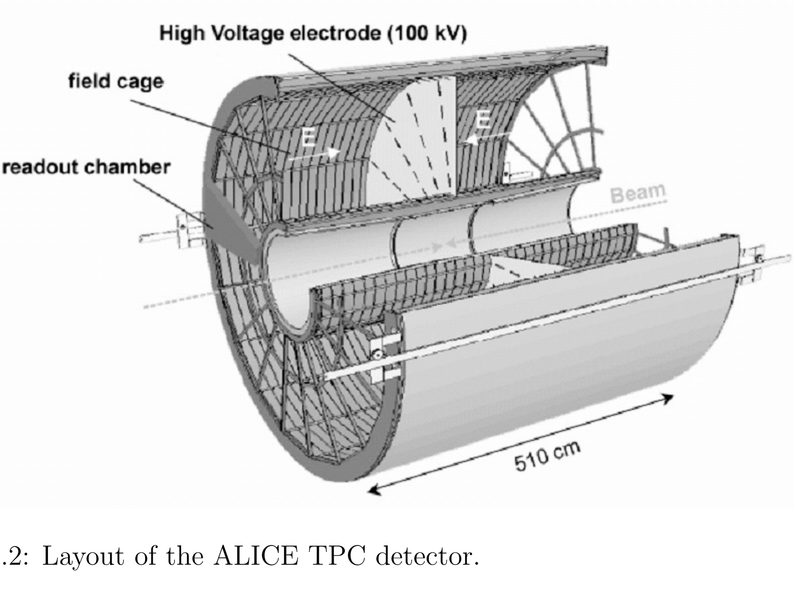


Figure 1.2: Layout of the ALICE TPC detector.

1.2 The TPC front-end electronics

The 18 trapezoidal sectors on each side of the TPC are divided into 6 readout partitions of which 2 are located in the inner chamber and 4 in the outer chamber as shown in figure 1.3. Each readout partition is controlled by the readout control unit which is connected to a number of front-end cards. The main task of the front-end cards is to processes the signals generated by charge deposition on the readout pads. In total 216 readout control units and 4356 front-end cards are needed to read out data from 570132 channels.

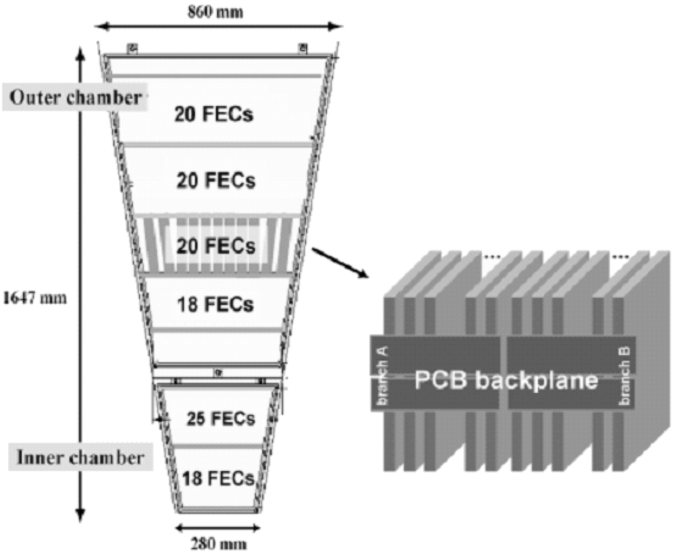


Figure 1.3: Each TPC sector is divided into 6 readout partitions. One RCU per readout partition is in charge of reading out data from the detector through the FECs [4].

1.2.1 Readout Control Unit

The Read Out Control Unit (RCU) consists of the RCU motherboard, the Detector Control System board (DCS board) and the Source Interface Unit (SIU). A Xilinx Virtex-II Pro FPGA, hereafter called the RCU main FPGA, is mounted on the back side of the RCU motherboard. The RCU main FPGA plays a key role in the TPC front-end electronics as it is in charge of data readout from the TPC detector. It is responsible for moving data from the front-end cards to the SIU were the data is optically transmitted via the data detector link to the data acquisition system. In addition it carries out tasks related to configuration and housekeeping of the front-end cards. The RCU is therefore divided into a readout network and a control network with a corresponding readout node and control node in the RCU main FPGA user design. This is illustrated in figure 1.4. Through the DCS bus the control node of the RCU main FPGA is connected to the DCS board which essentially is an embedded computer running Linux. In this way remote access can be provided through a standard Ethernet connection. This option adds flexibility to the system even though the front-end electronics is physically inaccessible. Upgrades of the RCU main FPGA can therefore be carried out after start-up of the experiment. Detailed information about the RCU and front-end electronics can be found in [5], [6], and [7].

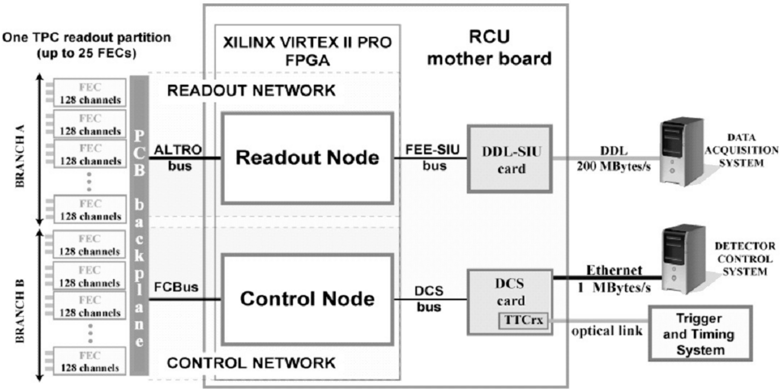


Figure 1.4: The architecture and readout path of the TPC electronics for one readout partition [4].

1.2.2 Choice of FPGA technology

Because the RCU main FPGA is a vital part of the data readout path, it is important that it remains fully functional during the operation of the experiment. At the time when the RCU main FPGA device choice was made, different FPGA technologies were therefore evaluated based on their single event upset susceptibility. Reprogrammability was also an important requirement as this would allow to develop and upgrade the user design during and after commissioning of the detector. The standard available technologies considered were:

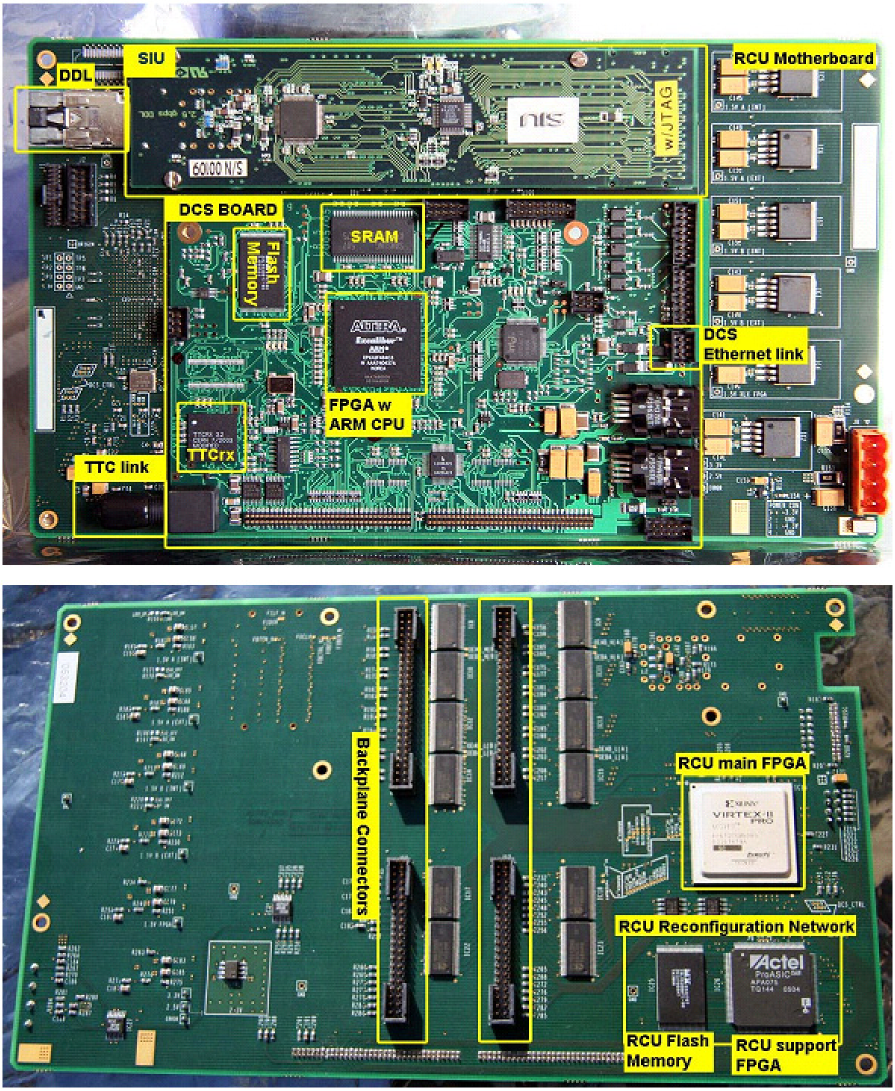


Figure 1.5: Top: The RCU motherboard with the DCS board and SIU card attached: Bottom: The back side of the RCU motherboard where the FPGAs and reconfiguration network are located.

- *SRAM*, where the user-programmed functionality is stored in SRAM memory cell.
- *Flash*, where the memory element is a floating gate transistor that can be turned off by injection charge into the floating gate.
- *Anti-fuse*, where an electrically programmable switch that is initially high impedance forms a low resistance and permanent connection when programmed.

Anti-fuse and Flash based FPGAs are non-volatile. This means that the user-programmed functionality continues to be stored when the power is turned off. No external circuitry is therefore needed to program the FPGA when powered on. Once an anti-fuse switch is programmed, the process can not be reversed. Anti-fuse FPGAs are therefore not susceptible to configuration loss due to single event upsets [8]. Nonetheless, due to their one-time programmability, this technology was not considered as an alternative candidate for the RCU main FPGA. Flash based FPGAs are also considered inherently single event upset tolerant [9] and contrary to anti-fuse FPGAs, they can be reprogrammed. However, at the time when the device choices were made, no Flash based FPGAs were available with enough resources to implement the data acquisition design of the RCU main FPGA. In-field programmability was also limited due to the need of a higher programming voltage than what was available for the readout electronics. These limitations strongly favoured the use of SRAM based FPGAs even though they are known to be sensitive to single event upsets. In the end an SRAM based FPGA from Xilinx was found suitable because it offered the possibility of partial reconfiguration on an operational system. An external system can therefore be designed in order to detect and repair single event upsets in the configuration memory of the Xilinx FPGA without disrupting its operation [10]. This external system is further described in chapter 3 where it is referred to as the reconfiguration network of the RCU main FPGA. Along with the RCU main FPGA it is located on the back side of the RCU motherboard as shown in figure 1.5. In contrast to the RCU main FPGA, fewer resources were required to implement the control functionality of the reconfiguration network. A Flash based FPGA from Actel was therefore chosen together with a Flash memory device to store the configuration data of the RCU main FPGA.

1.3 Primary objective and main contributions

The purpose of this thesis is to investigate the various aspects related to qualifying the Xilinx Virtex-II Pro FPGA for use in the radiation environment of the ALICE experiment. First of all this requires knowledge about the radiation environment at the location where the device is operated. This can be obtained through simulations using Monte Carlo transport codes. Accelerated beam tests are mandatory in order to investigate how sensitive a device is to a certain type of radiation. In relation to single event upsets this sensitivity is referred to as the single event upset cross section, σ_{SEU}. When combined with the knowledge of the radiation environment, it can be used to predict the rate at which single event upsets will occur in the ALICE experiment. The effects and consequences of single event upsets are however difficult to accurately quantify. Single event upsets in FPGAs are not one to one correlated to measurable errors in the expected functionality. Additional testing is therefore needed to investigate this issue, and also to measure the effectiveness of any techniques applied in order to reduce the failure probability. Physics based simulations is another method that can provide additional insight into how different types of radiation and material compositions of a device may contribute to the single event upset rate.

All these aspects are treated in this thesis and the main contributions are listed below:

- Monte Carlo simulations to determine the radiation environment in the location of the RCU main FPGAs, presented in section 2.3.3.

- Development of the RCU support FPGA user design that is in charge of the various reconfiguration procedures for the RCU main FPGA. This design is presented in section 3.2.2 and was developed together with Johan Alme [11]. An important feature of this design is the ability to carry out frame-by-frame read back to verify the integrity of the configuration memory, and further correct single event upsets if detected. It is an important part of the mitigation strategy for the RCU main FPGA.

- Accelerated beam tests to investigate the single event upset sensitivity of the Xilinx Virtex-II Pro FPGA configuration memory in a 29 MeV proton beam at the Oslo Cyclotron (OCL). The results are presented in section 4.2.1.

- Accelerated beam tests to investigate the effectiveness of the frame by frame, read back, verification and correction procedure when combined with other system level error detecting and correcting techniques. The results are presented in section 4.2.2.

- Implementation of a fault injection solution on the RCU as a method to test how single event upsets may effect the functional behaviour of the final RCU main FPGA user design. This method, which is presented in chapter 5, can for instance be used to test the effectiveness of any mitigation strategies applied to the design.

- Preparation of a test case study where physics based simulations will be carried out to investigate the single event upset rate of the Xilinx Virtex-II PRO FPGA. The work is presented in chapter 6.

1.4 Outline

This thesis is divided into seven chapters including this introduction chapter. Chapter 2 starts by giving a brief overview of the basic single event effects related to SRAM memory and FPGAs. The basic mechanism responsible for the single event upsets is then presented along with how single event upsets may affect the SRAM memory cell and FPGA. The second part of the chapter focuses on the radiation environment in the location of the RCU main FPGA, and simulation results are presented. Chapter 3 introduces partial reconfiguration and how it is applied to the RCU main FPGA. Special attention is given to the RCU reconfiguration network and the RCU support FPGA responsible for the different configuration procedures. Chapter 4 presents the results of the accelerated beam tests of the RCU main FPGA. The single event upset cross section for the configuration memory is measured using a 29 MeV proton beam at the Oslo Cyclotron. A standard shift register design is further used to demonstrate how the reconfiguration network in combination with triple modular redundancy will reduced the consequences of SEUs. Finally some predictions are made of the expected number of single event upsets when the RCU main FPGA is operated in the ALICE radiation environment. Chapter 5 describes how fault injection is implemented for the RCU main FPGA. A case study is presented to demonstrate how fault injection can be used to test the effect of single event upsets on the RCU main FPGA user design. This case study uses the same, however somewhat smaller in size, shift register design as for the accelerated beam tests. Chapter 6 describes how physics based simulations are carried out for a case study of the RCU main FPGA. Physics based simulations using Monte Carlo transport codes can be applied to study how the single event upset rate may be affected by different material compositions and types of radiations. xsFinally the thesis is concluded in chapter 7 with and outlook on prospective work.

Chapter 2

Radiation effects in the TPC RCU main FPGA

It has become a well established fact that modern digital integrated circuits can be sensitive to ionizing radiation. When for instance an SRAM cell is exposed to a transient noise pulse, this may change the state of the memory cell from a logical 1 to a logical 0 or vice versa. The basic mechanism responsible is the energy deposition from an incoming ionizing particle traversing the sensitive part of the SRAM cell. This effect is commonly referred to as a single event upset (SEU) or soft error.

SEUs are the main radiation effects of concern for the SRAM based RCU main FPGA in the TPC front-end electronics. They can lead to a variety of undesirable effects where the loss of vital control functionality is the main worry. An SEU may result in breakdown of the readout of data from the TPC detector. In order to avoid or reduce the consequence of an SEU, it is essential to have a good understanding of the basic mechanism responsible, including its potential effects in the FPGA. An overview and introduction to this subject is therefore given in the first part of this chapter. The SEU rate or frequency of a device is strongly dependent on the radiation environment it is exposed to. The last part of the chapter will therefore describe the radiation environment that is expected in the vicinity of the TPC front-end electronics.

SEUs are not the only radiation induced effects in integrated circuits. A number of other radiation effects also exist that may have temporary or even permanent damaging potential. However, in a radiation environment consisting of mainly energetic hadrons, which is the case for the TPC detector, it is the SEU which is of main concern and consequently the focus of this chapter and thesis.

2.1 Single Event Effects

Numerous acronyms are used by the radiation effects community to describe the different types of radiation effects in digital integrated circuits. This section gives a brief overview and description of the most relevant effects related to SRAM based FPGAs. Another compact summary of radiation effects related to Xilinx FPGAs can be found in [12].

The common term for any measurable effect resulting from the deposition of energy from a single ionizing particle strike, is a single event effect (SEE). The most relevant SEEs are:

Single Event Upset (SEU)

The JEDEC standard [13] defines an SEU as a soft error caused by the transient signal induced by a single energetic particle strike. In [14], it is said to occur when a radiation event causes a charge disturbance large enough to reverse or flip the data state of a memory cell, register, latch, or flip-flop.

An SEU can be defined in a number of ways. Essentially it refers to any type of memory cell whose content or value has been changed into an erroneous state due to a radiation event. As a memory cell stores the value of a bit, it is also commonly referred to as a bit flip, meaning the bit value has been flipped or inverted. An SEU can be categorized as a soft error. The error is “soft” because the device is not permanently damaged by the radiation. When new data is written to the struck memory cell, the device will store it correctly [14].

Multiple Bit Upset (MBU)

An MBU is a single radiation event that results in more than a single bit being flipped. Each bit flip is essentially an SEU. An MBU is therefore considered to be a subset of the SEU. They are usually a small fraction of the total number of observed SEUs. The MBU probability is however steadily increasing as geometries shrink [15], [16]. Since most conventional error correcting techniques are only capable of detecting and correcting single bit flips, MBUs are an increasing reliability concern. Still, the conclusion from [16] is that approximately 1-3% of the upsets induced by a 63.3 MeV proton beam are MBUs for the Xilinx Virtex-II Pro FPGA. MBUs will therefore not be the main source leading to functional failures in the RCU main FPGA. Determining the MBU cross section was therefore not within the scope of this thesis.

Single Event Transient (SET)

An SET is a transient pulse in the logic path of an IC. Similar to an SEU, it is induced by a charge deposition of a single ionizing particle. An SET can be propagated along the logical path where it was created. It may be latched into a register, latch or flip-flop causing their output value to change. In the case of the Xilinx FPGAs the FPGA structure by its nature is highly resistant to SETs due to the large capacitive loading of the single path [17]. Compared to SEUs in the FPGA configuration memory, SETs are therefore considered a negligible problem and are not treated in this thesis.

Single Event Functional Interrupt (SEFI)

Xilinx [15] defines SEFI as an SEE that results in the interference of the normal operation of a complex digital circuit. SEFI is typically used to indicate a failure in a support circuit, such as loss of configuration capability, power on reset, JTAG[1] functionality, a region of configuration memory, or the entire configuration. For the Xilinx Virtex-II Pro FPGA used as the RCU main FPGA, the SEFI cross section is typically orders of magnitude lower than the SEU cross section [17]. As for the previously mentioned SETs, further investigation of SEFI rates are not considered for this thesis.

Single Event Latchup (SEL)

The latchup phenomenon occurs when a spurious current pulse activates the parasitic bipolar transistors that are inherent in complimentary metal-oxide-semiconductor (CMOS) structure [18]. The result is an abnormally high current state which may lead to permanent damage if not restored to normal operation by a power cycle. If the origin of the current pulse is an ionizing particle it is called a single event latchup. Because it is a potentially destructive event, it is not categorized as a soft error. SELs are typically a problem for devices operating in environments of energetic heavy ions such as in space. For environments consisting of single hadrons only, such as the ALICE TPC or in terrestrial applications, SELs are considered a negligible problem. No high current events, that needed a power cycle to be restored, were detected during the accelerated beam test presented in chapter 4.

[1] JTAG: Joint Test Action Group.

2.2 Basic mechanism

2.2.1 Charge generation and collection

As a charged particle passes through matter it loses energy through the process known as ionization. This is the starting point of all single event effects. Along its path, the charged particle generates electron-hole pairs through scattering with the atomic electrons of the material. In this process the target nucleus remains at a fixed location due to the small amount of energy transferred. The charged particle is only slightly deflected and its path can therefore be considered as a straight line. If the path traverses a reverse-biased p-n junction, charge carries are collected by the electric field and drifted to the nearby node where a current/voltage transient is created [14]. The process is illustrated in figure 2.1. Charge collection is dominated by the faster drift process followed by a slower diffusion process. An additional effect of the generated column of electron-hole pair, is its distortion of the electric field. A funnel shaped extension of the depletion region enhances the drift collection, and effectively more charge can be collected at the node. The funnel concept was first introduced by Hsieh in [19] and depends on the doping concentration of the substrate. Increasing the doping concentration will decrease the distortion of the electric field lines. The amount of collected charge is a complex combination of factors like the size of the device, biasing of the various circuit nodes, substrate structure and device doping [14]. In addition the type of ion, its energy and trajectory through the node plays an important role.

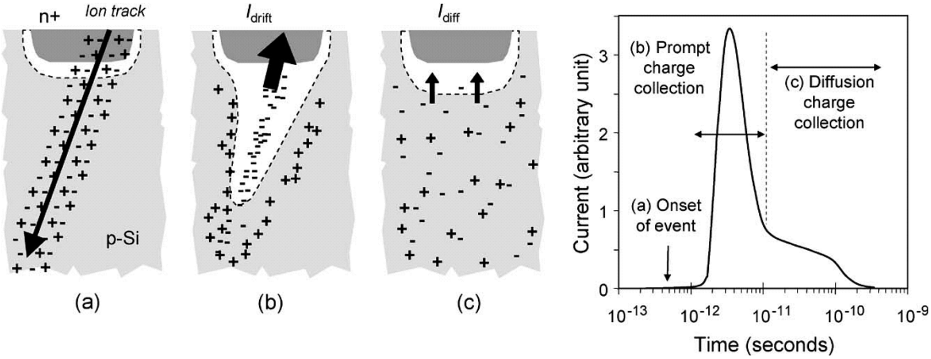


Figure 2.1: Charge generation and collection phases in a reverse-biased junction and the resultant current pulse caused by the passage of a energetic ion [14].

2.2.2 Single event upsets in SRAM memory

A single event upset was previously defined as bit flip in a memory cell caused by the transient signal induced by a single energetic particle strike. Figure 2.2 shows a schematic of a 6-transistor SRAM cell. It consists of cross coupled inverters each with one NMOS and one PMOS transistor. The sensitive regions in an SRAM cell are the drain areas of the transistors in the “OFF” state. These regions correspond to a reverse biased pn-junction capable of collecting the charge carriers. When a particle strikes one of the drain areas, a transient current pulse will be created at the output of the respective inverter. This current pulse is then propagated to the input of the other inverter pair. If the width and amplitude of the pulse are sufficient, the next inverter stage will change its output as a consequence. Effectively, a new value will be loaded/latched in the memory cell. The sensitivity of an SRAM cell depends on factors like the node capacitance, the operating voltage and the speed of the feedback circuit [14]. The node capacitance together with the channel resistance acts as a low pass filter that may reduce the rising slope and magnitude of the induced current pulse. With down scaling of technology and feature sizes, the operational voltage of a device is also decreased. This means that less charge is needed to induce an SEU. Increasing the capacitive load is therefore a known design technique to reduce the sensitivity as the technology nodes gets smaller [17]. Combined, parameters like these define the amount of charge or energy needed to flip the bit of a memory cell. This is also referred to as the critical energy, E$_{crit}$, or

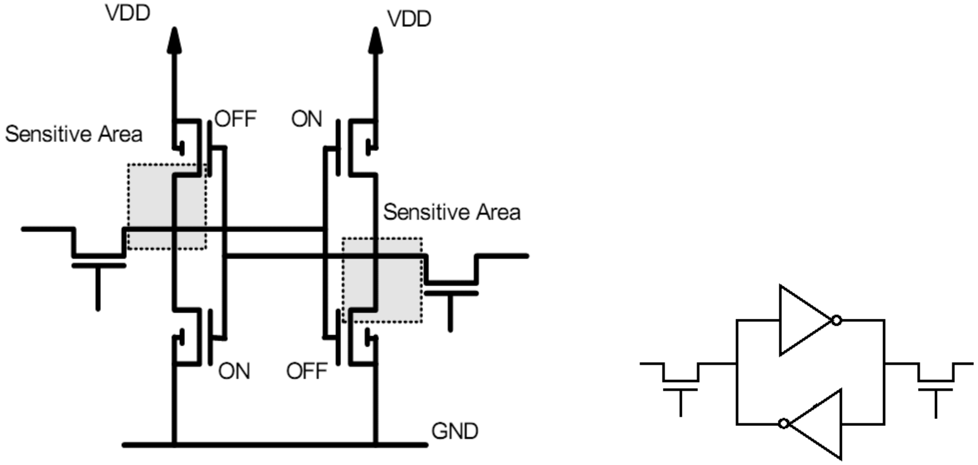


Figure 2.2: 6-transistor SRAM cell [20].

critical charge, Q$_{crit}$. That is, an SEU can be induced if a charge larger than the critical charge has been collected by the sensitive node. Critical charge and critical energy can be used interchangable, provided that one knows how to change the unit from one to the other. If silicon is assumed as the material of the charge collection volume, which is usually the case, then

$$Q_{crit}(\mathrm{infC}) = 44.5 \cdot E_{crit}(\mathrm{inMeV}) \quad [21]. \tag{2.1}$$

2.2.3 Single event upsets in the FPGA configuration memory

The main building blocks of a modern SRAM based FPGA are a number of programmable logic blocks structured in a large matrix. In an FPGA, a function is implemented by mapping it into this pre-existing and programmable logic. This mapping is referred to as its configuration [22] and is stored in a large array of SRAM cells. If an ionizing particle causes one or multiple SRAM cells to change its value during normal operation, this is referred to respectively as an SEU or an MBU. If this corrupted bit controls a logical resource utilized in the implemented design, this may cause a malfunction in the operation of the FPGA. For example, look-up tables are used to implement Boolean functions in SRAM based FPGAs. The content of the look-up table is stored as part of the configuration memory, and if the stored value is corrupted due to an SEU it will no longer store the correct Boolean function. This can result in unwanted and incorrect behaviour. Similarly, other programmable logic such as for example routing multiplexers are controlled by the content of one or several SRAM cells. An SEU can therefore cause a broken connection between two logical blocks and consequently corrupt the flow of data in the FPGA. These types of malfunctions are referred to as functional failures or functional errors in the following.

An FPGA can contain millions of configuration bits and the larger part controls the complex routing network interconnecting all the logic blocks. For a typical FPGA design only a fraction (90-98%) of the configuration bits are used [20]. An SEU in an unused bit will have no effect on the normal operation of the device. This effect was seen during both the accelerated beam tests and fault injection tests presented in chapter 4 and 5. During irradiation several of the SEUs detected did not seem to have any influence on the operation of the FPGA. Thus, making prediction on the expected functional failure rate of the RCU, the relationship between SEUs and functional failures must first be established. This ratio is highly dependent on the implemented design due to how the utilization of logical resources may vary from one design to another. In combination with accelerated beam tests to determine the SEU sensitivity of a device, fault injection is a well suited method to further determine the average number of SEUs per functional failure. Fault injection is therefore discussed in chapter 5.

An SEU in the configuration memory of an FPGA can only be corrected by reconfiguring the device. A standard mitigation technique is therefore to regularly reconfigure the full device to correct any accumulated SEUs. This approach is implemented for the RCU main FPGA and is further described in chapter 3.

2.2.4 The physics of single event upsets

The dominant mechanism of energy loss by a charged particle passing through a material is Coulomb scattering by the atomic electrons of the material. Due to the small size of the nucleus compared to the size of an atom, a collision with the nucleus is far less probable. Nevertheless, nuclear interactions play an important role prior to the creation of an SEU and will be discussed in this section.

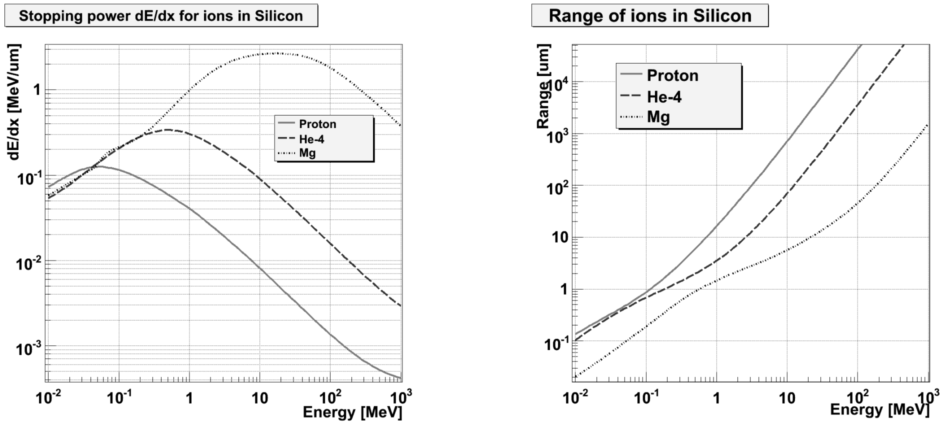
Stopping power and range

A measure for the energy loss of a particle per unit length is the stopping power. The stopping power can be related to the number of electron-hole pairs produced per unit length along the particle's track. For silicon, the conversion factor is 3.6 eV per electron-hole pair [21]. The stopping power is usually expressed in MeV cm^2 mg^{-1}. If multiplied by the material density it becomes MeV/cm, energy per unit length.

Figure 2.3(a) shows the stopping power of a proton, an α-particle and a magnesium ion in silicon. As can be seen, the stopping power is dependent on the kinetic energy and charge of the particle. Except for very low energies, the higher the charge of the ion the higher is the stopping power. Thus a magnesium ion is more ionizing than the α-particle, and the α-particle more than the proton. The stopping power increases with decreasing energy until it reaches a maximum and then starts to decrease. At low energies the ion has a tendency to pick up electrons which lowers its effective charge and thus its stopping power.

After travelling a certain distance in the material, the ion eventually loses all its energy and comes to rest. This distance is referred to as the *range* of the ion. For comparison the range of protons, α-particles and magnesium ions in silicon is plotted as a function of energy in figure 2.3(b).

In section 2.2.2 the critical charge was introduced as the amount of charge needed to induce a bit flip in a memory cell. In addition a memory cell is associated with its sensitive volume. That is, charge which is deposited within this volume can be considered to be collected by the sensitive node. Knowing its size and shape, and the particle direction and stopping power in the material, the amount of deposited



(a)

(b)

Figure 2.3: Stopping power for various ions in silicon. The plots are generated by data from SRIM 2006 [23].

charge within the sensitive volume can be estimated. To a first order the sensitive volume can be defined by the area and depth of the NMOS or PMOS transistor of the SRAM cell. For a 0.13 μm technology such as the Xilinx Virtex-II Pro, the maximum track length through the sensitive volume is probably[2] in the order of 1 μm and below. The maximum stopping power of a proton is 0.14 MeV/μm which corresponds to approximately 6 fC/μm . This is below the average critical charge of 12 fC [24] for for the Xilinx Virtex-II Pro. Because direct ionization from protons is not expected to be the cause of SEUs, a different mechanism must therefore be responsible.

SEUs from non-elastic interactions

In the case of highly energetic neutrons and protons, they can collide with the nucleus of the target material and induce a nuclear reaction. Compared to the Coulomb interactions nuclear reactions are rare events. However, due to the large number of SRAM cells in modern devices, significant particle fluxes, and the lengthy exposure times, the contribution from non-elastic interctions can not be disregarded.

Interactions between a nucleon (i.e. a proton or neutron) and a nucleus can be either elastic or non-elastic. In an elastic reaction the incident and outgoing particles are the same. Due to the small amount of momentum transfer in an elastic

[2] See section 6.2.

collision they are considered to only play a minor role of SEU rates [21]. For future applications, when the feature size is further downscaled and the critical charge is decreased, this assumption may no longer hold and elastic interactions may have to be reconsidered. In a non-elastic interaction on the other hand, additional particles can be created and emitted from the reaction. Figure 2.4 shows an example of how an energetic particle can enter a device and create a non-elastic interaction in the silicon substrate. One or several of the fragments produced may be emitted with the right direction and enough energy to reach a sensitive transistor. It is typically the recoil ion or an α-particle that possesses enough stopping power to induce an SEU. An example of a reaction channel for a 100 MeV p + Si reaction is given in equation 2.2.

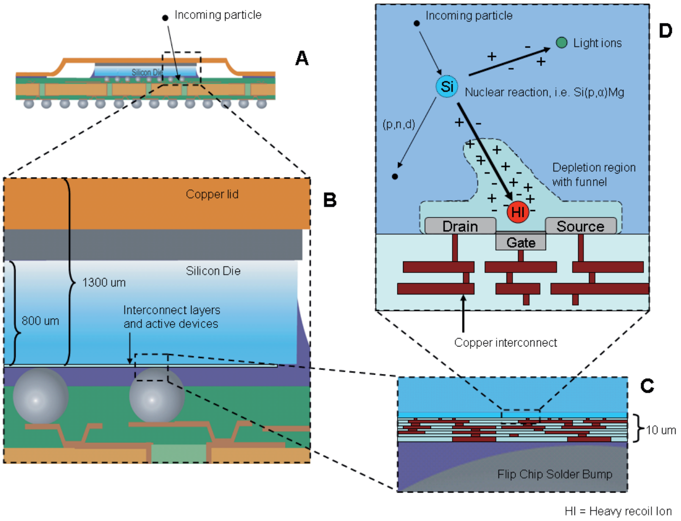


Figure 2.4: Cross section schematic showing the structure of the device. A particle entering from the top may travel through both the copper lid and silicon substrate before it creates a non-elastic interaction nearby a sensitive node. The resulting fragments may have enough energy to induce and SEU if their paths traverses the sensitive node.

$$p +^{28}_{14}\mathrm{Si} \longrightarrow p + 2\gamma +^{4}_{2}\mathrm{He} +^{24}_{12}\mathrm{Mg} \tag{2.2}$$

This produces a 5.4 MeV α-particle and a 4.9 MeV magnesium ion. Their ion energies are similar, but due to the higher charge of the magnesium, its stopping power is approximately 2.3 MeV/μm compared to approximately 0.14 MeV/μm for the α-particle. At the production location, the stopping power of a 5.4 MeV α-

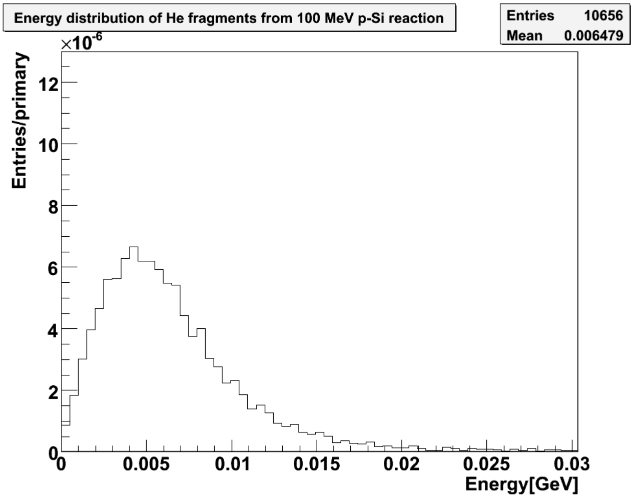


Figure 2.5: Energy distribution of α-particles produced by Fluka simulations of 10^8 protons of 100 MeV on a 100 μm thin silicon target.

particle may not be large enough to cause an SEU. However, as it starts to move its kinetic energy will decrease and consequently its stopping power will increase. With a range close to 28 μm it therefore has potential to cause an SEU at a certain distance from its production location. Figures 2.5 and 2.6 shows the energy distribution of α-particles and magnesium ions produced in a Fluka [25] [26] Monte Carlo simulation of a 100 MeV proton beam on a thin silicon target. The mean energy of the magnesium ions is approximately 2.2 MeV while it is 6.5 MeV for the α-particle. Because a 2.2 MeV magnesium ion has a range of approximately 2.3 μm in silicon, it must be created in close proximity to a sensitive transistor in order to pose a reliability risk. On the other hand, α-particles can potentially contribute even if they are produced as far away as 25-50 μm .

2.3 The TPC radiation environment

When high energetic beams of lead ions collide in the central point of the ALICE experiment, a high primary particle production rate is expected. Many of these particles produce secondaries through hadronic and electromagnetic cascades in absorbers and structural elements of ALICE [2]. This leads to particle fluxes which may pose a reliability risk to the front-end electronics of the TPC detector. An understanding of the TPC radiation environment is therefore important in order to

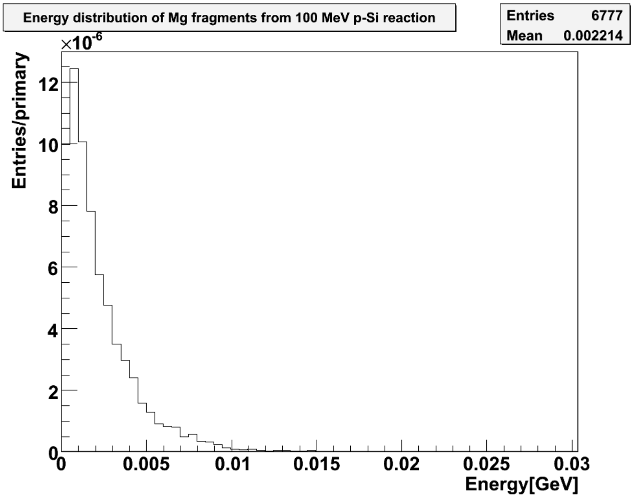


Figure 2.6: Energy distribution of magnesium recoil ions produced by Fluka simulations of 10^8 protons of 100 MeV on a 100 μm thin silicon target.

make predictions of failure rates due to single event upsets.

2.3.1 Particle multiplicity

The event rate for Pb-Pb collisions at the LHC nominal luminosity of 10^{27}cm^{-2}s^{-1} will be about 8000 minimum-bias collisions per second, of which only 5% corresponds to the most central ones [2]. At present the estimates for the multiplicity density (dN/dη) in a central Pb-Pb collision range from 1500-4000 charged particles per unit of pseudorapidity[3] at mid-rapidity. This corresponds to 40000 particles in the worst case scenario. The average particle multiplicity for minimum-bias runs is roughly 1/5 of the multiplicity of a central event [27], [28]. When running simulations of particle fluences in Fluka the result is normalized to the number of primary particles transported and square centimetre, $particles/(cm^2 \cdot primary)$. The particle flux can then be calculated according to equation 2.3

$$Flux = \Phi_{Fluka} \cdot R_{col} \cdot \frac{N_{central}}{5}, \tag{2.3}$$

where Φ_{Fluka} is the fluence result from Fluka , R_{col} is the collision rate, and $N_{central}$ is the particle multiplicity of a central Pb-Pb event.

[3] Pseudorapidity η describes the angle of the particle momentum relative to the beam axis. $\eta = -ln[tan(\theta/2)]$.

Total dose

One month of the year will be dedicated to Pb-Pb collisions, and in [29] the total dose for 10 ALICE years is estimated to approximatley 6 Gy[4] at the TPC locations. For the rest of the year, p-p collisions will run at an interaction rate of 200 kHz [2]. This is a factor 25 higher than for Pb-Pb collisions. Assuming that the particle multiplicity roughly scales with the number of participants (2 for protons and 416 for lead), the total number of particles from 10 ALICE years of p-p collisions is approximately a factor 1.2 compared to Pb-Pb collisions (25 x 10 months x 2/416). The total dose is therefore not expected to be higher than a few kRad.

2.3.2 Previous work

Particle transport simulations have already been carried out to estimate the radiation environment in the TPC detector. In [29] the particle fluence was scored in a 1 mm thick silicon disc representing the front-end electronics. It was divided in four cylindrical scoring regions to study the fluence at different radial distances from the beam line. The results of the simulations are summarized in tables A.1 and A.2 of appendix A.

2.3.3 Updated simulations with new geometry description

In [29] the geometry description of the experiment and detectors was based on ALIFE[5] [30]. It did not contain a description of the actual front-end electronics. The front-end electronics of the TPC detector consist of 216 RCU nodes controlling 4356 front-end cards equally divided between both sides of the TPC. All of these cards are encapsulated in copper envelopes for water cooling and make up extra material between the beam collision point and the RCU main FPGA. Thus it was desirable to implement a more detailed description of the front-end electronics to study whether this would have any impact on the hadron fluence in the locations of the different RCUs. Also, since the present most updated geometry description of the ALICE experiment is available in the AliRoot framework [31], the updated simulations are based on this description. Applying the Virtual Monte Carlo interface (VMC) [32], AliRoot can be combined with Fluka for particle transport.

[4] 1 Gy = 100 Rad

[5] ALIFE is an editor and parser for the Fluka geometry and detector definition.

Geometry description and scoring regions

Figure 2.7 shows how the front-end electronics has been implemented in the geometry description of the TPC detector. The front-end cards are represented by the yellow volumes while the RCUs are represented by the green volumes. A more detailed description of the geometry can be found in appendix A.2. Compared to the scoring scheme in [29], using the RCUs as scoring regions gives a more accurate fluence versus location information. To increase statistics, the RCUs are grouped in 6 rings of 18 cards for each side of the TPC. The RCU scoring rings are labeled 1 through 6, where ring 1 consist of the 18 innermost RCUs (closest to the beam line), and ring 6 the 18 outermost RCUs. Figure 2.8 shows how the location of the 6 scoring rings compares to the scoring scheme used in [29].

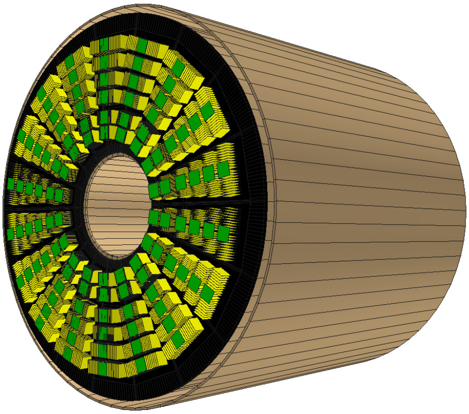


Figure 2.7: A view of the TPC geometry with the added front-end cards (yellow) and RCUs (green).

Results of the preliminary simulation run

In total 15 independent simulations were carried out for the geometry implementation without the front-end cards, and 18 for the geometry implementation with the front-end cards included. For each simulation 10000 primary particles were requested to be transported. The results in figures 2.9(a) and 2.9(b) show that the hadron fluence was not significantly changed by adding the extra geometry. It can be argued whether a tiny shielding effect is observed on the absorber side.

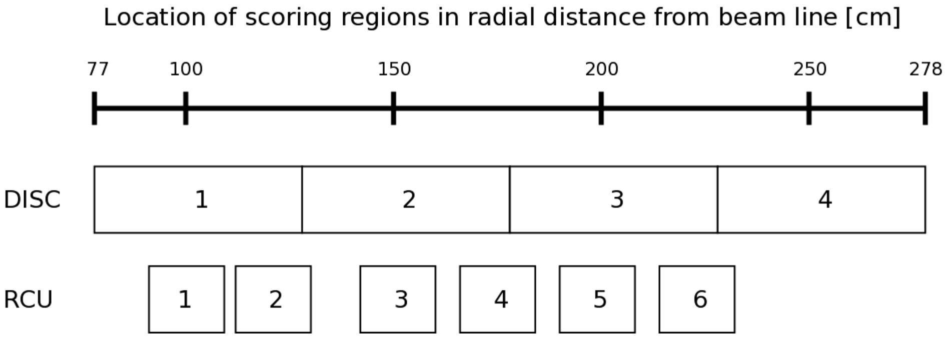
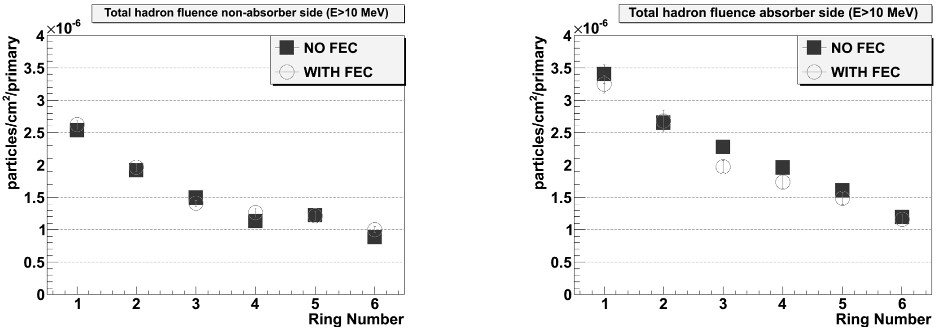

Location of scoring regions in radial distance from beam line [cm]
77
100
150
200
250
278
DISC
1
2
3
4
RCU
1
2
3
4
5
6

Figure 2.8: A comparison showing the location of the RCU scoring rings compared to the scoring scheme used in [29].



(a)

(b)

Figure 2.9: Total fluence of energetic hadrons ($E_{kin} > 10MeV$) for the non-absorber (a) and absorber (b) side as a function of radial distance from the beam line. Error bars represent the standard deviation between the individual runs as produced by Fluka.

This shielding effect is more evident for the low energy neutrons (E<10 MeV) for the absorber side as can be seen in figure A.7. Appendix A also shows additional plots and detailed tables of all the scored particles.

Good agreement is found when comparing the results of the overlapping scoring rings/regions with the previous simulations in [29]. The absorber side has a slightly higher hadron fluence than the non-absorber side, and both sides show the same trend of decreasing hadron fluence with increasing distance from beam line.

Applying equation 2.3, the total hadron flux for each RCU ring can be calculated based on tables A.5 through A.8 in appendix A. The results are shown in tables 2.1 and 2.2 for a particle multiplicity of $N_{central}$=40000, and can be used to calculated the expected SEU rate for all RCUs.

Flux $[part/(cm^2s)]$	Scoring ring absorber side (WITH FEC)					
	1	2	3	4	5	6
p	3 ± 14%	6 ± 19%	5 ± 17%	4 ± 21%	4 ± 19%	2 ± 20%
n	191 ± 4%	147 ± 7%	94 ± 6%	86 ± 7%	70 ± 8%	55 ± 7%
$\pi^{\pm}$	11 ± 13%	16 ± 7%	23 ± 5%	19 ± 7%	19 ± 6%	15 ± 7%
had$^{\pm}$	15 ± 10%	24 ± 7%	31 ± 5%	24 ± 6%	24 ± 6%	19 ± 6%
Sum (had$^{\pm}$ + n)	207 ± 4%	171 ± 6%	126 ± 4%	111 ± 5%	94 ± 6%	74 ± 5%

Table 2.1: Estimated hadron fluxes (particles/cm^2/s) for each scoring ring for a particle multiplicity of $N_{central}$=40000 (absorber side). The numbers are based on the simulation runs with the front-end cards implemented and for $E_{kin} > 10MeV$.

Flux $[part/(cm^2s)]$	Scoring ring non-absorber side (WITH FEC)					
	1	2	3	4	5	6
p	15 ± 12%	8 ± 9%	5 ± 11%	4 ± 15%	4 ± 15%	3 ± 16%
n	85 ± 3%	68 ± 4%	53 ± 4%	50 ± 7%	48 ± 8%	39 ± 5%
$\pi^{\pm}$	63 ± 3%	44 ± 4%	28 ± 5%	25 ± 7%	23 ± 6%	19 ± 6%
had$^{\pm}$	82 ± 3%	56 ± 3%	36 ± 4%	30 ± 6%	29 ± 5%	24 ± 6%
Sum (had$^{\pm}$ + n)	167 ± 2%	125 ± 2%	90 ± 3%	81 ± 5%	77 ± 5%	64 ± 4%

Table 2.2: Estimated hadron fluxes (particles/cm^2/s) for each scoring ring for a particle multiplicity of $N_{central}$=40000 (non-absorber side). The numbers are based on the simulation runs with the front-end cards implemented and for $E_{kin} > 10MeV$.

2.3.4 Discussion

Assuming that only hadrons above 10 MeV are of concern, and further that these hadrons can be treated as the same particle, tables tables 2.1 and 2.2 can be used calculate the expected SEU rate. The basis of these assumptions is briefly discussed in the following paragraphs.

Neutron and protons

In [21] the characteristics of proton-nucleus and neutron-nucleus interactions are considered to be very similar for energies above 100 MeV. Figure 2.10 shows a comparison of the non-elastic interaction cross section of protons and neutrons in silicon. In fact, the interaction cross sections are very similar down to 20-30 MeV. For lower energies the main difference between the neutron and proton is the Coulomb repulsion which decreases the non-elastic cross section of the proton. Lower incident particle energies also reduce the probability of producing light secondary fragments such as the α-particle. This is seen for both neutrons and protons in figures 2.11(a) and 2.11(b). For the p + Si interaction, the α-particle production cross section is significantly reduced below 20-40 MeV. Because neutrons are not affected by Coulomb repulsion they are slightly more effective in producing α-particles at energies below 40 MeV. However, below 10 MeV the α-production from neutrons shows a sharp cut-off with a threshold close to 5-6 MeV. The contribution for neutrons in this energy region is therefore expected to be negligible due to the low fragment production cross section and possible energy transfer.

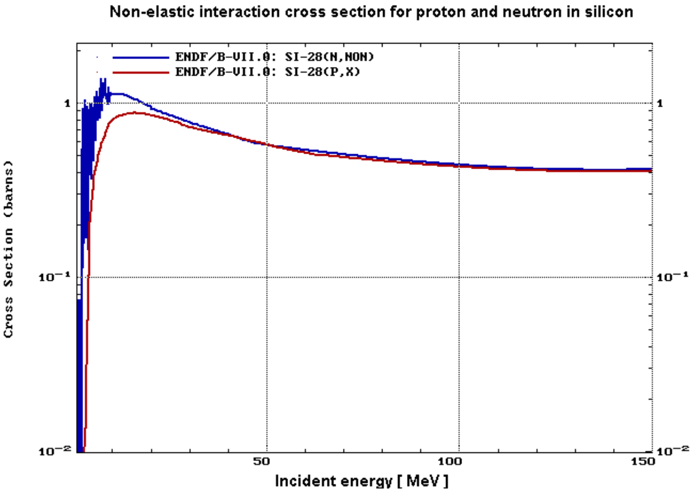
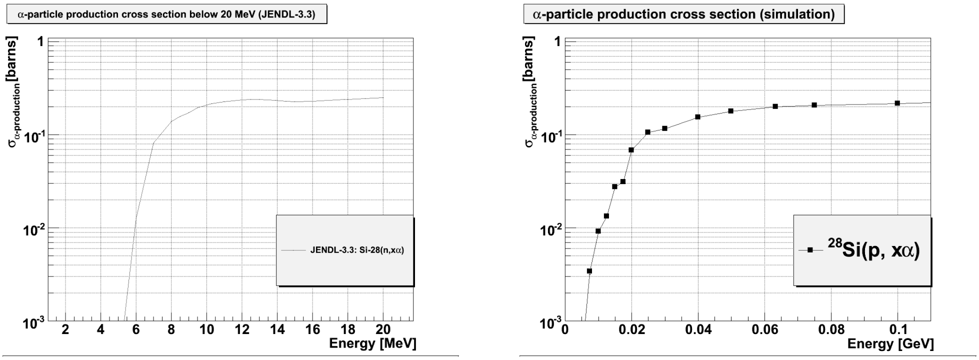


Figure 2.10: Non-elastic reaction cross section of protons and neutrons in silicon [33]. 1 barn = 10^{-24} cm^2.

Figures 2.12 and 2.13 shows the simulated fluence for neutrons and protons as a function of energy summed over all the 6 RCU scoring rings. For both neutrons and protons a peak can be found around 100 MeV. However, while the proton fluence below 10 MeV is effectively negligible, there is a significant amount of lower energy neutrons. Thermal neutrons need to be considered only for the ^{10}B(n,α)^{7}Li reaction which has a high capture cross section. The result of this reaction is a ^{7}Li ion of



(a)

(b)

Figure 2.11: α-particle production cross sections for the n + Si(a) and p + Si(b) interactions. The neutron data is obtained from [33] while the proton result is obtained from a Fluka simulation with protons beams of different incident energies hitting a thin silicon target.

approximately 0.9 MeV[6] and an α-particle of 1.47 MeV, both capable of inducing SEUs. Since ^{10}B has been effectively removed from all Xilinx technologies below 220 nm [17], thermal neutrons are not considered to be an issue for the Xilinx Virtex-II Pro used on the TPC readout control unit.

Pions

While protons and neutrons have comparable cross section characteristics, pions show a slightly different behaviour. Figure 2.15 shows a comparison of cross sections for protons and pions in silicon. The plots are produced from Fluka simulations of proton and pion beams on a thin silicon target. For parts of the energy range of interest in the TPC radiation environment, the α-particle production from pion interactions is a factor 2-3 higher than from proton interactions. Similarly, the π^++Si non-elastic interaction cross section is also slightly higher in a certain energy range. It is therefore not obvious that pions can be treated similarly to protons and neutrons when considering their effectiveness in inducing SEUs. Unfortunately no pion beam SEU cross section data is available for the Xilinx Virtex-II Pro FPGA. Therefore the best approximation is currently to apply proton and neutron SEU

[6] 0.84 MeV in 94% of the time and 1.014 MeV in 6% of the time.

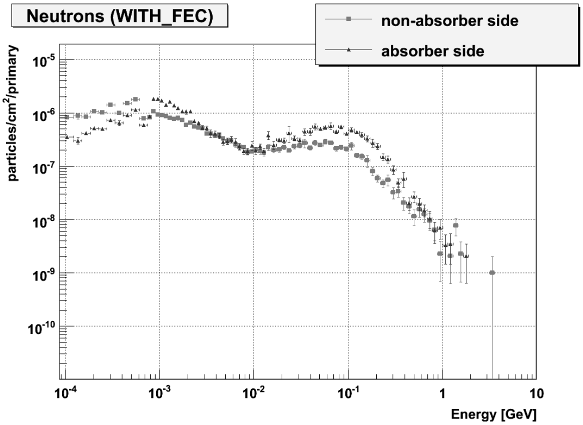


Figure 2.12: Simulated fluence of neutrons as a function of energy. Summed over all scoring regions and showed for the non-absorber side and absorber side with the Front-end cards implemented.

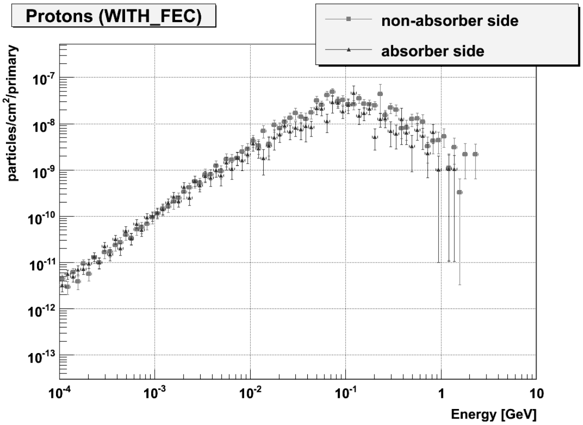


Figure 2.13: Simulated fluence of protons as a function of energy. Summed over all scoring regions and showed for the non-absorber side and absorber side with the Front-end cards implemented.

cross section data when considering pions. Because the relative number of pions to the total hadron flux varies between 5-30%, the potential systematic error of using this approximation is reduced. Some research has been done by others to study the effect of pions in DRAM and SRAM devices [34], [35]. The results are however not unambiguous with a device dependent spread in the experimental data for DRAMs. For SRAMs it is claimed in [35] that it can not be demonstrated that pions are more effective than protons in creating upsets. It should be mentioned that the SRAM devices tested in this study used an older technology process ($0.5\mu m$ - $0.8\mu m$) than what is the case for the Xilinx Virtex-II Pro ($0.13\mu m$).

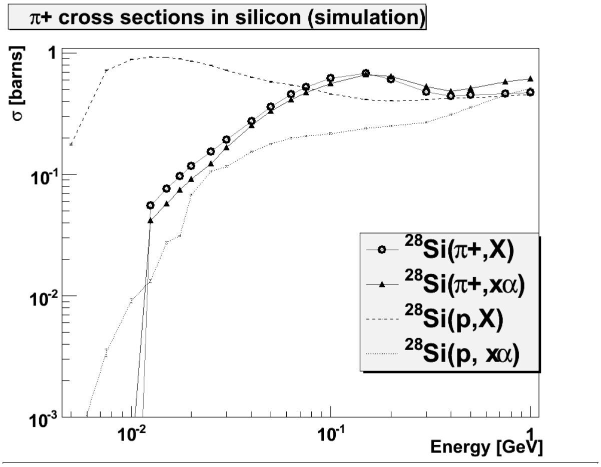


Figure 2.14: Cross sections for π^+ and protons in silicon.

2.4 Summary

Due to the nature of the TPC radiation environment the main radiation effect of concern for the RCU main FPGA are single event upsets. The main mechanism responsible is identified as the energy deposition from fragments produced in non-elastic interactions. Direct ionization from protons is neglected due to their low stopping power.

Based on simulations of Pb-Pb events and transport of the produced primary particles, the hadron flux at the location of the RCU main FPGA has been calculated as presented in tables 2.1 and 2.2. Compared to previous simulations a more

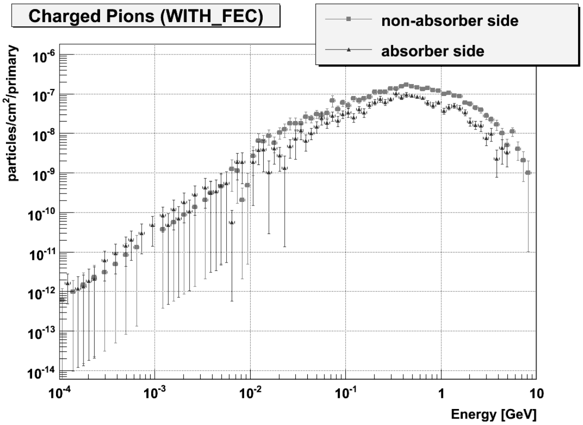


Figure 2.15: Fluence of π^+ as a function of energy summed over all 6 RCU scoring rings.

detailed description of the front-end cards were implemented. The added geometry description did however not significantly influence the hadron flux above 10 MeV.

When calculating the total hadron flux in the different locations of the RCUs, it is assumed that protons, neutrons and pions are equally effective in inducing SEUs. In addition the energy cut-off is made from 10 MeV and below. For the accuracy needed in the final calculations in this thesis work, this approximation is accepted as valid. In cases where higher precision is needed, one should consider to include the contribution from neutrons below 10 MeV and down to the non-elastic interaction threshold. Also, a closer investigation of the potential difference between pions, protons and neutrons could be carried out.

Chapter 3

RCU radiation tolerant system solution

In chapter 1 the RCU motherboard and the DCS board were introduced as parts of the front-end electronics. Because the RCU main FPGA is in charge of data readout for the TPC detector, it is important that this FPGA is kept in operational status during a data taking run. A solution has therefore been implemented that will repair SEUs in the configuration memory of the RCU main FPGA through the use of partial reconfiguration. This method allows reading back a subset of the configuration memory checking for an error in the data. If an error is found, it is corrected by rewriting a correct version of the data.

This chapter will start by introducing partial reconfiguration before describing how it has been applied to correct SEUs in the RCU main FPGA configuration memory. The RCU reconfiguration network will be presented along with the user design of the RCU support FPGA which is the low level controller of the reconfiguration procedures.

3.1 Partial reconfiguration

Partial reconfiguration is defined by Xilinx in [36] to be *"rewriting a subset of configuration frames, either while user design is suspended ("Shutdown" partial reconfiguration) or while the user design is operating ("Active" partial reconfiguration)"*. For the RCU main FPGA it is important that it is kept in continuous operation during data taking. Shutdown partial reconfiguration is therefore not an option because this implies that the FPGA will be inactive for a short moment. Active partial reconfiguration means that certain areas of the device can be reconfigured while others areas remain operational and unaffected by the reprogramming. This is possible due to the special architecture of the Xilinx configuration memory.

3.1.1 Xilinx Virtex-II Pro configuration memory

The Xilinx Virtex-II Pro is a user-programmable gate array of Configurable Logic Blocks (CLBs) and embedded blocks such as user memory (BlockRAM). Each CLB contains four similar slices used to build combinatorial and synchronous logic designs. A simplified schematic of a slice is shown in figure 3.1 where the main building blocks are the Look-Up Tables (LUTs) and configurable register. The LUTs can be used to implement any arbitrarily defined boolean function of four inputs. Additionally it can also be configured as distributed memory or as a 16-bit shift register. A CLB element is tied to a switch matrix with access to the general routing matrix of the FPGA. This extensive routing network is used to interconnect all the configurable elements. It also connects to the BlockRAM user memory through the BlockRAM Interconnets. Input and output pins connects the FPGA to external circuitry and are configured by special Input/Output blocks (IOBs and IOIs). Moreover, dedicated clock pins are used to connect to the Digital Clock Managers (DCMs), clock buffers, and the global clock lines. The global clock resources are configured by a special part of the configuration memory called GCLK. All these

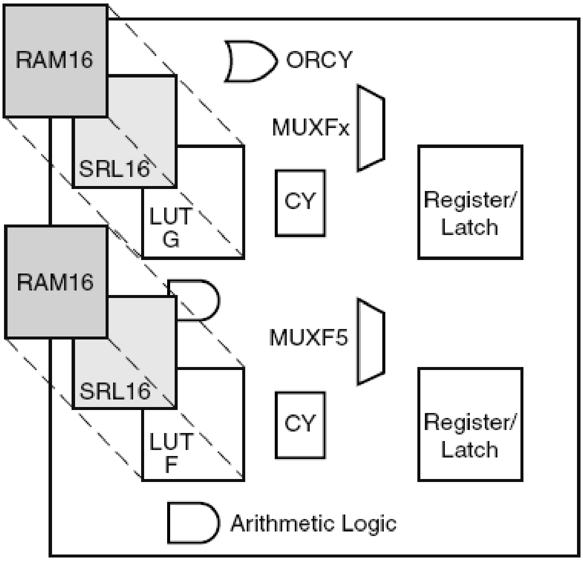


Figure 3.1: The Xilinx Virtex-II Pro slice, a basic building block of the CLBs [37].

programmable elements are controlled by values stored in static memory cells. These values are loaded in the memory cells during configuration and can be reloaded to change the functions of the programmable elements.

The Virtex-II Pro configuration memory is divided into six column types that corresponds roughly to the physical device resources: IOB, IOI, CLB, GCLK, BlockRAM, and BlockRAM Interconnect. These column types are further grouped in three main blocks as illustrated in figure 3.2 [36]. A 3-dimensional address pointer is used to access the configuration memory. The highest level of addressing is the Block Address (BA), and within each block the columns are addressed by their Major Address (MJA). Each column is again sub-divided into a number of minor frames which are the lowest addressable units of the configuration memory. That is, a minor frame is the smallest portion of data that can be read from or written to. In the following a minor frame is referred to as a frame. A frame in the Xilinx Virtex-II Pro FPGA contains 424 bytes of data organized in a 1-bit vertical column from the top to the bottom edge of the FPGA. In total there are 1320 frames, 804 in block 0, 384 in block 1, and 132 in block 2. Block 0 is the largest block as it contains not only columns for configuring the GCLK and IOBs, but also the full matrix of CLBs. Due to the vertical alignment of the frames, the IOI and IOB frames, which are situated on the left and right edges of the FPGA, only controls resources connected to the I/O's on the right and left edges. The I/O's along the top and bottom edges are controlled by a certain number of bytes in the top and bottom part of each CLB frame.

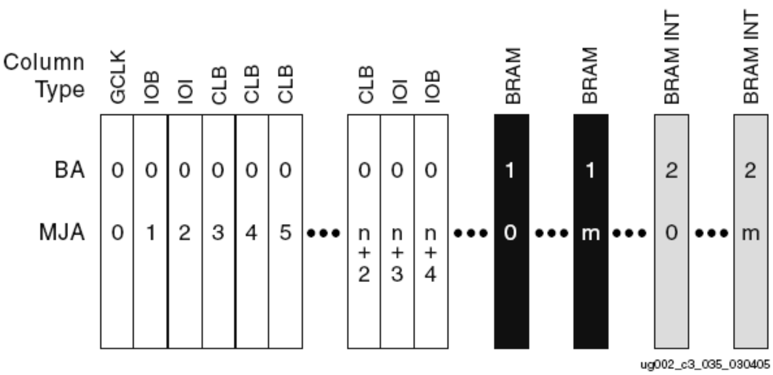


Figure 3.2: The configuration memory map of the Xilinx Virtex-II Pro. BA: Block address, MJA: Major address [36].

The number of columns within a block and frames within a column, varies with the size of the Virtex-II Pro device. More details can be found in Table 4-18 in [36].

3.1.2 Configuration process

There are three main processes that can be used to configure the Xilinx Virtex-II Pro FPGA. Access to the configuration memory is offered through the SelectMAP interface, specific to Xilinx FPGAs only. It is an 8-bit bidirectional data and 5-bit control bus. In the context of this thesis configuration consists of two sub-categories, initial configuration and reconfiguration or scrubbing.

- *Initial configuration* - This involves an erase of the configuration memory before new data is written to the full configuration memory. This is true both if the FPGA already runs with valid configuration data, and if the configuration memory is empty. In any case, the FPGA will turn inactive from the point of erase until new data has been loaded and the device again returns to an active state. The source of configuration data is a standard Xilinx configuration file produced by the Xilinx BitGen[1]tool. Also referred to as a *bitstream file*, it contains configuration data for all the available frames in the FPGA configuration memory.

- *Reconfiguration* - In this context also referred to as *scrubbing*[2], is different with respect to initial configuration in that it does not first erase the configuration memory. This makes it possible to reconfigure the device without interrupting the user design. However, for this to work, the FPGA must be reconfigured with the same data already present in the configuration memory.

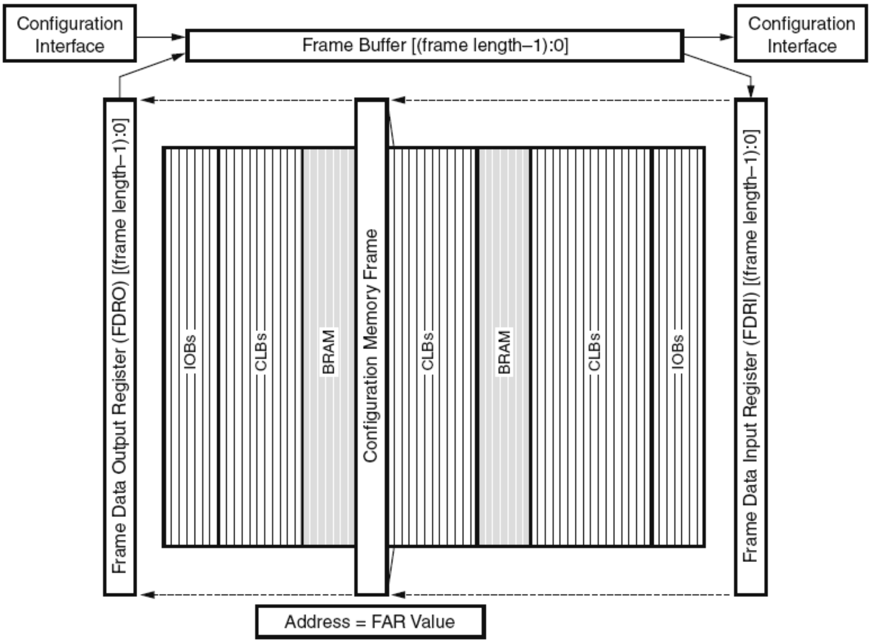
  A reconfiguration file, also referred to a a *scrubbing file* is produced by the Xilinx BitGen tool. In general the scrubbing file is smaller than an initial configuration file because it only contains data for frames which are utilized in the user design. In fact, this is a partial reconfiguration.

The configuration interface of the Xilinx Virtex-II Pro consists of a set of registers used to control the configuration process. Before loading the configuration data, these registers are initialized with information like the number of frames to be written, the size of a frame and the start address of the first frame. The relevant commands are located in the header part of the configuration files. Two additional important flags are the *PERSIST* and *SBITS* flags, located in the Control Register (CTL). Setting the *PERSIST* flag keeps the SelectMAP interface in configuration mode after completion of the initial configuration. If left unset, the SelectMAP interface pins becomes normal user I/O's which prevents the possibility to do reconfiguration or scrubbing. The *SBITS* flags are used to set the security level of the

[1] The Xilinx bitstream generation program

[2] Xilinx [15] defines scrubbing as the process of correcting any configuration cell upsets through FPGA partial reconfiguration. Scrubbing does not interrupt user design function

configuration memory. That is, it can block the possibility to read back the data stored in the configuration memory. An essential part of the work described in this chapter is to check for SEUs in the configuration memory. This can only be done if readback is enabled and the *SBITS* flags should therefore be left unset.



Configuration Interface
Frame Buffer [(frame length–1):0]
Configuration Interface
Frame Data Output Register (FDRO) [(frame length–1):0]
IOBs
CLBs
BRAM
Configuration Memory Frame
CLBs
BRAM
CLBs
IOBs
Frame Data Input Register (FDRI) [(frame length–1):0]
Address = FAR Value

Figure 3.3: The configuration path. Reading or writing frames of data to the configuration memory is pipelined through the frame buffer [36].

After initialization a special register called the *Frame Data Input Register* (FDRI) is used to write data to the configuration memory. This operation is pipelined through the frame buffer such that the first frame of data is written to the configuration memory while the second frame is being shifted through the frame buffer. The process is illustraded in figure 3.3. For this reason the FDRI must be flushed with one frame of pad data at the end of a configuration process to make sure that the last data frame is loaded. After a frame of data has been shifted into the FDRI, it is loaded in parallel into the frame specified by the value previously written to the Frame Address Register (FAR). The last part of a configuration file contains a footer with commands to complete the configuration process in a controlled fashion. For instance, in case of an initial configuration a special start-up sequence must be enabled to bring the FPGA into an active mode. The file structure with a header part, a main body of data, and a footer part, is common for all the configuration files. Different file types are illustrated in figure 3.4. The content of the header and footer differs slightly between an initial configuration file and a scrubbing file. As the scrubbing file contains fewer data frames, this part of the main body is also shorter. However they are both represented by the same structure shown to the left in the figure. The two rightmost file structures in figure 3.4 are generated by tailor made software. While a scrubbing file reconfigures all the utilized frames in the configuration memory, the WriteFrame file is used to reconfigure only a single frame. The main body therefore only contains one frame of data in addition to the pad frame. Header and footer information is modified to accommodate this. The

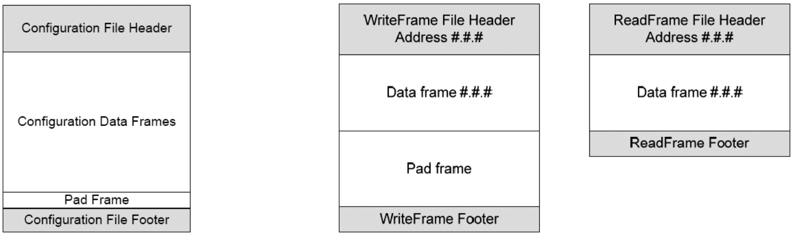


Figure 3.4: Left: Structure of a Xilinx Configuration file. Right: Structure of the generated frame files: write-frame file and read-frame file [11].

ReadFrame file is used to read back a given frame of data. A read process starts by writing the header part of the ReadFrame file to the relevant configuration registers. It contains information about the address and size of the frame to be read. The direction of the SelectMAP data bus is then switched into output mode before the frame data is read from the configuration memory. There is no need for a pad frame in this case. It is therefore omitted from the main body of the ReadFrame file. However, due to the pipelined process using the frame buffer, the first frame read out of the device is not valid data. The smallest possible read process therefore requires two frames of data. After the frame data has been read, the SelectMAP bus is switched back into input mode and the footer part is written to complete the read process. The initial configuration file contains all the frames of data that will be written to the configuration memory. These are stacked in a sequential order. WriteFrame and ReadFrame files are generated by extracting each individual data frame from the initial configuration file. Respective header and footer commands are then added for either a read or a write process. A pad frame is also added to the main body of the WriteFrame.

3.1.3 Partial reconfiguration on the RCU

The main purpose of implementing partial reconfiguration on the RCU is to correct any SEUs that may be induced in the configuration memory. The solution implemented allows this to be done in two ways: continuously overwriting the configuration memory regardless whether an SEU has been induced or not, or first do a read back of the configuration memory to check for an SEU, and reconfigure the part of the memory where the SEU was detected. While the first option generally is referred to as scrubbing, the latter is in this thesis referred to as Frame by frame Readback, Verification and Correction (FRVC). A ReadFrame and WriteFrame file is produced for each data frame in the configuration memory. The main body of the ReadFrame file contains the data which is expected to be read from the configuration memory. Using the ReadFrame files in a sequential order, all frames are read back from the configuration memory, one by one, and compared to the answer book data frame in the main body of the ReadFrame file. If a difference is detected, the corresponding WriteFrame file is used to reconfigure the frame where the SEU were found.

Of course, an SEU could also be corrected through an initial configuration, but as this involves bringing the FPGA into an inactive state, it is not a very efficient method. Except for the limitations mentioned in 3.1.4, the architecture of the Xilinx Virtex-II Pro configuration memory allows all or parts of the memory to be reconfigured without interrupting the design. As the FPGA remains in the active state this is referred to as Active Partial Reconfiguration (APR). This is applied in the FRVC procedure where a partial reconfiguration is carried out only if an SEU has been detected. It also allows to time stamp the SEU and keep track how many SEU has been detected. This information can be used to correlate any abnormal behaviour of the user design to a detected SEU.

3.1.4 Limitations of partial reconfiguration

Reconfiguration of the FPGA puts a few constraints on the user design. The most important is related to implementation of user memory. In the Xilinx Virtex-II Pro the CLB Look Up Tables (LUT) can be configured to be memory elements. This is for instance an efficient way of implementing shift registers without using the register resources of a CLB [36]. It is important that this feature is not utilized in combination with reconfiguration. The LUT are part of the configuration memory and if used as dynamic memory in a shift register design, reconfiguration will overwrite the dynamic data with the default initial data of the configuration file. LUT must therefore strictly be used to implement static logic. However, when developing the partial reconfiguration solution for the RCU main FPGA, this feature was utilized for testing purposes. A LUT was used to implement a standard memory module. During initial configuration of the FPGA, an initial bit pattern was loaded to the implemented LUT memory. During operation this memory is accessible from the user design level an can be overwritten with a different bit patten. If a read back of the memory is then carried out through the SelectMap configuration interface, this new bit pattern will be detected and compared against the initially loaded bit pattern. This method proved to be an efficient way to confirm correct implementation of the SEU detection and reconfiguration procedure.

The BlockRAM resources are also part of the configuration memory and should therefore not be reconfigured. Alternative methods like hamming encoding or other SEU mitigation techniques should be applied to secure the BlockRAM data. An example presented by Xilinx can be found in [38]. Simply just reading back the BlockRAM is also not recommended as the configuration control logic shares the same access interface as the user design. During a readback the user design is therefore not able to access the BlockRAM data. The BlockRAM interconnect frames are on the other hand a static part of the configuration memory. These frames are used to configure how the BlockRAM should be utilized and must be included in the reconfiguration.

During an initial configuration of the FPGA readback of the configuration memory can be disabled by setting the *SBITS* in the Control Register. Combined with encryption of the bitstream it can be used to prevent for instance reverse engineering of the user design. This security measure can not be used in combination with reconfiguration.

3.2 The RCU and reconfiguration network

To control the configuration process of the RCU main FPGA a reconfiguration network has been implemented. Its main components are a Flash based support FPGA (Actel ProASIC^{plus} APA075 [39]) and a Flash memory device [40]. The support FPGA is the main configuration controller and all configuration files needed are stored on the Flash memory device. High level housekeeping tasks are taken care of by the DCS embedded computer which is connected to the reconfiguration network through the DCS bus. A conceptual schematic of the RCU is shown in figure 3.5 where the data path of the readout system is indicated by the black arrows. Besides the initial configuration of the RCU main FPGA at power up, the main purpose of the reconfiguration network is to detect and reconfigure any SEUs and thereby reduce the probability of interrupting the data flow.

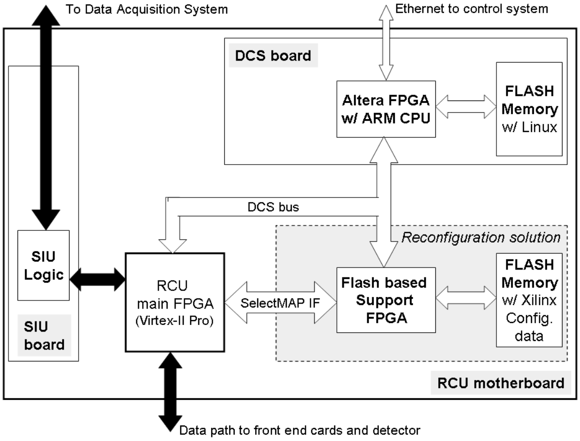

To Data Acquisition System
Ethernet to control system
DCS board
Altera FPGA w/ ARM CPU
FLASH Memory w/ Linux
DCS bus
Reconfiguration solution
SIU Logic
SIU board
RCU main FPGA (Virtex-II Pro)
SelectMAP IF
Flash based Support FPGA
FLASH Memory w/ Xilinx Config. data
RCU motherboard
Data path to front end cards and detector

Figure 3.5: Conceptual schematic of the RCU motherboard. The data path of the system is marked with black arrows. As can be seen it passes through the Xilinx Virtex-II Pro FPGA.

3.2.1 DCS communication and operational modes

The DCS board runs a dedicated Linux operating system on an Altera Excalibur FPGA [41]. This FPGA consists of an embedded ARM processor core in addition to an area of programmable logic. External SRAM and Flash components serve as the memory and hard drive of the Linux system. In the programmable logic part of the FPGA, a special module has been implemented to access external hardware such as the RCU motherboard. The *RCU Communication Module* (RCM)[11] shown in figure 3.6 connects the DCS embedded computer to the RCU motherboard. It is the master of the DCS bus and each slave on the bus has a corresponding communication slave. Three modes of operation are supported by the DCS bus:

- *Normal mode*
- *SelectMAP mode*
- *Flash mode*

A conceptual model of the different modes are shown in figure 3.7. In normal mode, which is the default operational mode, the DCS bus is used in a memory mapped fashion to communicate with the registers in the user design of the RCU main FPGA and the RCU support FPGA. Normal mode operation will further be elaborated for the RCU support FPGA in section 3.2.2.

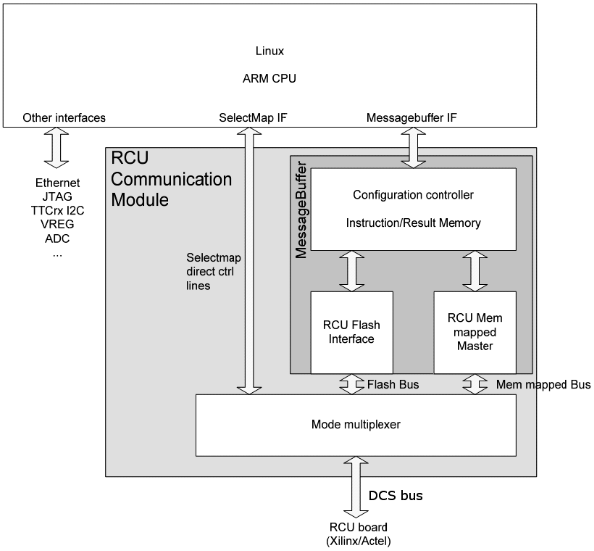


Figure 3.6: System sketch of the DCS configuration [11].

While the RCU main FPGA supports only normal mode operation, the RCU support FPGA also supports the SelectMAP and Flash modes. Two lines on the DCS bus are dedicated to select the mode of operation. These lines are constantly read by the RCU support FPGA in order to switch into the correct mode. In SelectMAP and Flash mode the RCU support FPGA acts as a tunnel for the DCS bus. A direct connection is established from the DCS RCM module to either the SelectMAP interface of the RCU main FPGA, or the Flash memory device. The reason for adding these modes is a combination of optimizing for speed, resource usage and robustness. Using memory mapped communication to communicate with the SelectMAP interface or the Flash memory device is not very efficient. Adding two additional and more direct communication paths from the DCS embedded computer reduce the layers of abstraction in the communication chain and optimize the speed at which the communication can be carried out. In case of Flash communication a write process is more complex than a read process. Running the RCU support FPGA in normal mode, a configuration process only needs read access to the Flash memory. As the logical resources of the RCU support FPGA is limited[3]

[3] The implemented design uses 93.4% of the available logic block of the RCU support FPGA. The synthesis report gives a possible clock frequency of 40.870 MHz which is at the limited compared to the LHC clock frequency of 40.08 MHz

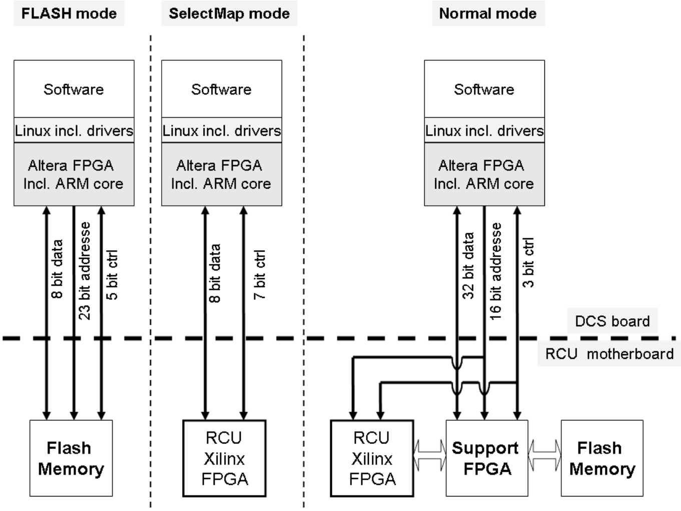


Figure 3.7: Conceptual model showing the different modes of operation in for the RCU system.

when compared to the tasks it will carry out, the *Flash Interface* module was moved up to the DCS level. The purpose of the Flash mode is therefore to add a direct connection to the Flash memory in order to upload RCU main FPGA configuration files.

The architecture of the RCU motherboard does not provide a remote configuration option for the RCU support FPGA. Upgrading the user design will therefore be difficult after the system commissioning has completed and the experiment has started. In case any unforeseen bugs are detected at this stage, correcting them may not be possible. The SelectMAP mode was therefore implemented to add redundancy in the configuration process of the RCU main FPGA. In SelectMAP mode the I/O pins of the SelectMAP interface are directly connected to the DCS embedded computer. A special Linux driver was developed to support the basic SelectMAP communication protocol. This makes it possible to configure the RCU main FPGA directly from DCS software. Different flavours of the RCM module and the Linux driver have further been developed in in [11] and [42] to support other projects using the SelectMAP interface. Additionally it should be mentioned that the SelectMAP mode is an important part of the fault injection solution developed for the RCU main FPGA and described in chapter 5.

3.2.2 RCU support FPGA

The RCU support FPGA is the low level controller of the RCU main FPGA configuration process. It supports three main configuration options:

- *Initial configuration* - This option issues a complete erase of the RCU main FPGA configuration memory before uploading the user design stored in the Flash memory device. The FPGA is inactive during the configuration process.

- *Complete reconfiguration (Scrubbing)* - This option reconfigures the memory with the same design already loaded in the memory. This is done to refresh the memory and correct any SEUs that may have accumulated. Scrubbing is a blind process. That is, it reconfigures the memory regardless of if an SEU is present or not. The FPGA remains fully active during the reconfiguration process.

- *Frame by frame Readback, Verification and Correction (FRVC)* - This options sequentially reads back single frames of data from the configuration memory, verifies the data with the original frame data stored on the Flash memory device through an XOR process, and if a difference is found, the respective frame address is overwritten with the original frame stored on the Flash memory device. The FPGA remains fully active during the FRVC process.

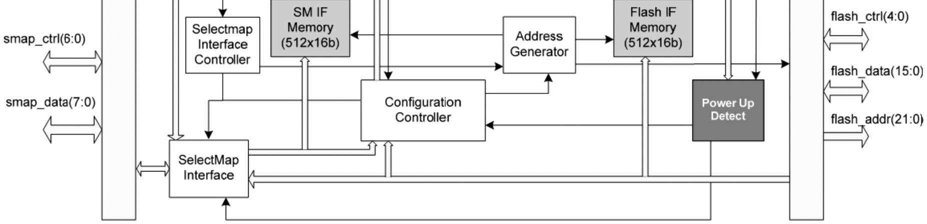


Figure 3.8: Block diagram showing the RCU support FPGA firmware [11].

During normal operation the main task of the RCU support FPGA is to keep the configuration memory of the RCU support FPGA intact with its original data. FRVC is the preferred process as it, in contrast to scrubbing, can keep track of the number of induced SEUs, and further reconfigure only the frame where an SEU has been detected. A block diagram of the RCU support FPGA user design[4] is shown in figure 3.8. Three levels of abstraction are implemented:

1. Detecting and switching between operational modes
2. Decoding of DCS address and control lines
3. Configuration logic

The *Mode Select & Wrapper* module is in charge of setting the correct operational mode based on the value of the two dedicated mode lines on the DCS bus. In SelectMAP and Flash modes the main control logic of the support FPGA is effectively bypassed to establish direct access from the DCS embedded computer to the configuration memory of the RCU main FPGA or the Flash memory. The purpose of these modes are described in section 3.2.1. In normal mode the *Memory Mapped Interface* module is in charge of decoding the address and control lines of the DCS bus. It provides a memory mapped interface to registers used by the configuration logic.

Configuration Controller

The configuration logic is controlled by the *Configuration Controller* module. A simplified block diagram is shown in figure 3.9. An internal state machine is in charge of decoding commands issued by the DCS embedded computer. The state machine has four main global states as can be seen in figure 3.10, an IDLE state and one state for each configuration procedure. A procedure is executed when the DCS embedded computer issues the command enable signal. This triggers the controller state machine to read the command register from *Memory Mapped Interface* module and carry out the requested procedure. Initial configuration and Scrubbing are simple procedures. The respective configuration file is located in the Flash memory and then transferred to the *SelectMAP Interface* module which takes care of the low level communication with the configuration memory of the RCU main FPGA. If initial configuration is requested, the *SelectMAP Interface* module carries out a special initialization procedure which erases the configuration memory of the Xilinx

[4] The user design of the RCU support FPGA has been developed in cooperation with Johan Alme, University of Bergen [11]

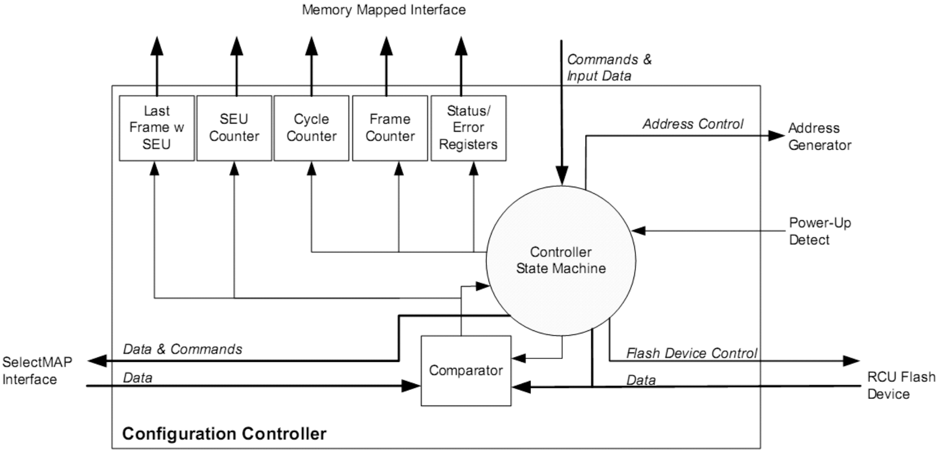


Figure 3.9: Block diagram of the configuration controller in the RCU support FPGA [11].

Virtex-II Pro FPGA and prepares for loading the initial configuration file. This initialization procedure is skipped if scrubbing is requested.

FRVC is a more complex procedure as it in addition to reading data from the Flash memory, also reads back data from the configuration memory of the RCU main FPGA. A simplified flow diagram of the FRVC procedure is shown in figure B.1 of appendix B. Frame by frame data is read back and compared through a simple XOR process. If the frames are found to differ, the SEU counter is increased by the number of differing bits, and the SEU error flag is set. This flag is made available to the DCS embedded computer and the user design of the RCU main FPGA. This means that the SEUs can be time stamped and therefore possibly correlated to a functional failure in the operation of the RCU main FPGA. The SEU error flag indicates that an error has been found in the current frame and this triggers a reconfiguration of this specific frame after which the SEU error flag is cleared.

During the different procedures the controller state machine keeps track of:

- *Frame Counter* - the number of frames read back
- *SEU Counter* - the total number of SEUs detected
- *Last Frame with SEU* - the frame number where the most recent SEU was detected
- *Cycle Counter* - the number of times a procedure has been executed

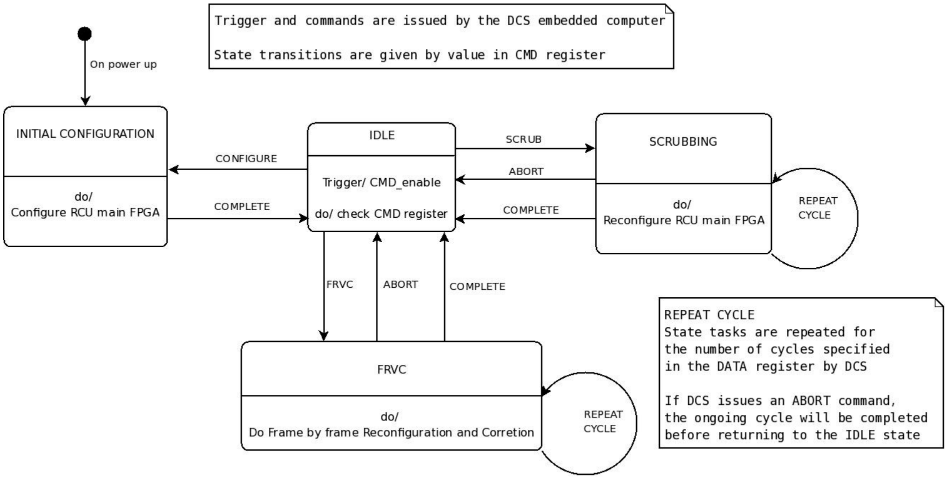

Trigger and commands are issued by the DCS embedded computer
State transitions are given by value in CMD register
On power up
INITIAL CONFIGURATION
do/
Configure RCU main FPGA
CONFIGURE
COMPLETE
IDLE
Trigger/ CMD_enable
do/ check CMD register
SCRUB
ABORT
COMPLETE
SCRUBBING
do/
Reconfigure RCU main FPGA
REPEAT CYCLE
FRVC
ABORT
COMPLETE
FRVC
do/
Do Frame by frame Reconfiguration and Corretion
REPEAT CYCLE
REPEAT CYCLE
State tasks are repeated for
the number of cycles specified
in the DATA register by DCS
If DCS issues an ABORT command,
the ongoing cycle will be completed
before returning to the IDLE state

Figure 3.10: The configuration controller state machine.

In addition a combined status and error register is updated to keep track of the last issued command and any failures in the operation. This information is continuously written to registers accessible from the DCS embedded computer through the *Memory Mapped Interface* (MMI).

A special feature of the Scrubbing and FRVC procedures is that they can be executed in a repetitive mode. This feature is controlled by writing the number of requested repetitions to the data register of the *Memory Mapped Interface* module. A continuous mode can be achieved by setting the data register equal to zero. A scrubbing or FRVC procedure can be aborted by writing the ABORT command to the command register at any point. In order to ensure a safe exit, the ABORT command will only be evaluated after the scrubbing file has been written to the configuration memory, or when all tasks for the current active frame has been completed.

3.3 Measurement of configuration times

Table 3.1 summarizes the measured times of the different configuration procedures. The configuration speed is essentially limited by the access time of the Flash memory and implementation details of the *configuration controller*. In section 4.3 the worst case number $2.4 \cdot 10^{-5}$ SEUs/(FPGA s) is computed for the expected SEU rate. To sufficiently reduce the probability for accumulation of SEUs, Xilinx [10] recommends

Operation	Time	Frequency
Initial configuration	113 ms	-
Scrubbing	77 ms	13 Hz
Read one frame	163 us	-
Write one frame	180 us	-
Read all frames (FRVC)	150 ms	6.6 Hz

Table 3.1: Measured times for the different configuration procedures. Note that the time of the scrubbing is dependent on the design, as the scrubbing file is compressed. Frequency is given for procedures that is meant to run continuous operation.

that the reconfiguration rate should be placed an order of magnitude above the upset rate. A reconfiguration frequency of 6.6 Hz for the FRVC procedure is therefore well within this recommendation. For the purpose of detecting and correcting SEUs, any furhter optimization for speed was therefore not considered for the configuration controller.

3.4 Summary

As the RCU main FPGA will operate in a radiation environment where it can be expected to experience functional failures due to SEUs in the configuration memory, partial reconfiguration is applied as a measure to correct these SEUs. It should be pointed out that this solution will only correct the SEU and not prevent it from occurring in the first place. The essential result is that it will prevent accumulation of SEUs in the configuration memory. This has important implications for functional failure rate of the FPGA and will be further discussed in section 4.1.3.

The configuration controller is implemented to run both reconfiguration of the full[5] configuration memory (scrubbing), and reconfiguration of single frames (FRVC). However, an advantage of the single frame operation is that it is able to keep track of the number of SEUs which has occurred. When enough statistics is accumulated, the result can be compared to the predicted SEU rate in section 4.3. The FRVC procedure also allows to time stamp the SEUs in order to study any correlation to functional failures that may be detected during a run. This type of diagnostics is not available when running the scrubbing procedure. FRVC will therefore be the preferred method during normal operation.

[5] Except for the BlockRAM frames

Chapter 4

Accelerated beam testing of the Xilinx Virtex-II Pro 7 and the RCU reconfiguration network

Accelerated beam testing have been carried out to estimate the SEU cross section of the Xilinx Virtex-II Pro SRAM configuration memory. Combined with information about the TPC detector radiation environment, the SEU cross section is used to predict the number of SEUs expected during an ALICE run. As explained in section 2.2.3, the number of SEUs alone can not be used to predict the functional failure rate at the system level. A shift register test design was therefore also monitored during irradiation to study how the SEUs affected the operation of the design.

The focus of this thesis is to present the results from the 29 MeV proton beam at the Oslo cyclotron where most to systematic studies were carried out. Accelerated beam test have also been carried out using a 180 MeV proton beam at the The Svedberg Laboratory in Uppsala [43].

4.0.1 Calculating the SEU cross section

The probability that a nuclear reaction will occur is given by the nuclear reaction cross section σ. The cross section has the unit of area and is on the order of the square of the nuclear radius R. The nuclear radius is in the order of $10^{-15} - 10^{-14}$ m which results in a small cross section. Thus the cross section is commonly measured in barns where 1 barn $= 10^{-28}\,\mathrm{m}^2 = 10^{-24}\,\mathrm{cm}^2$. The total reaction cross section can be defined [44] as

$$\sigma = \frac{N_R}{N_v \Phi} \tag{4.1}$$

where N_R is the number of reactions that take place, Φ is the number of incident particles per unit area, also known as the fluence, and N_v is the number of scattering centres per unit volume. Accelerated beam tests to experimentally determine the SEU sensitivity of a memory device can be compared to determining the nuclear cross section. The memory device is irradiated with a beam of incident particles I_0 and area A_{beam}. The fluence of the beam is $\Phi = I_0/A_{beam}$ and the area of the beam should be larger than the area of the chip to make sure all memory bits are covered. After irradiation the number of accumulated SEUs, N_{SEU}, in the memory device is counted. To reduce the chance of one memory bit being hit twice within the same experiment, the counted number of SEU should be kept at a fraction of the total number of memory bits. This assumption is true unless the memory is continuously being read during the experimental run. If hit twice, a memory cell will change back to its original value and will not be accounted for. The SEU cross section, that is, the probability of one incident particle causing an SEU in the memory device is determined by

$$\sigma_{SEU}(E) = \frac{N_{SEU}}{\Phi} \tag{4.2}$$

where E is the kinetic energy of the incident particle. It is the amount of deposited charge by a ionizing particle that determines if a memory cell will upset or not. If the deposited charge is larger than a certain critical charge, Q_{crit}, associated with the memory cell, an SEU will occur. The deposited charge is dependent on the ability of the particle to ionize the material (stopping power) which again is related to the kinetic energy of the particle. Thus, the SEU cross section of memory cell is a function of the kinetic energy of the particle. The SEU cross section is commonly measured in the unit of cm^2/bit. This is achieved by normalizing equation 4.2 to the number of memory bits tested, N_B, in the memory device.

$$\sigma_{SEU,bit}(E) = \frac{N_{SEU}}{\Phi}\frac{1}{N_B} \tag{4.3}$$

Equation 4.3 now corresponds well to equation 4.1 assuming that number of bits tested N_B can be compared to the total number of scattering centres per unit volume, N_v. Equation 4.3 is identical to the proton-induced SEU cross section for a mono-energetic proton beam as given in the JEDEC standard [13].

4.1 Experimental setup

The experiments have been carried out using a 29 MeV proton beam at the Oslo Cyclotron (OCL), at the University of Oslo, Norway. A schematic of the experimental setup is shown in figure 4.1. A collimator, located at the beam exit point

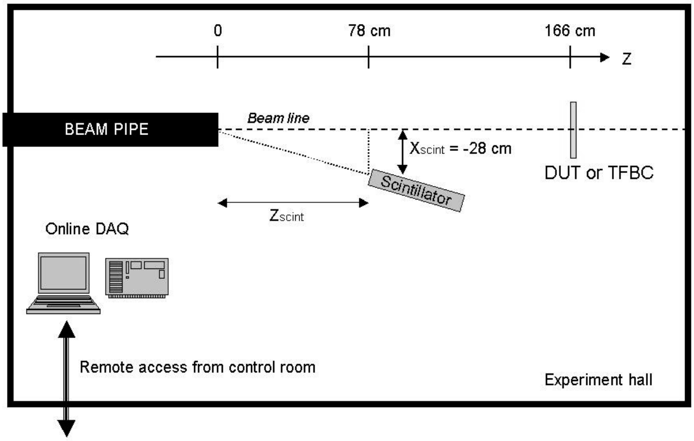


Figure 4.1: Experimental setup area at OCL.

inside the vacuume beam pipe, gives a beam spot diameter of approximately 1 cm. The device under test (DUT) is the RCU main FPGA. It is placed in the beam line 166 cm upstream from the exit point. At this point the beam spot was measured to be large enough to cover the full package of the FPGA which is approximately 2 cm by 2 cm. This gives a uniform beam over the chip die which is approximately 1 cm by 1 cm and centered within the FPGA package.

As the RCU is placed in the beam other components on the RCU can also be exposed to the beam. Even though the beam spot is small and covers mainly the RCU main FPGA, other devices should be protected from the beam if possible. The DCS card is attached in parallel to and on top of the RCU motherboard. A DCS bus extension card was produced to flip the DCS board off the RCU motherboard and out of the most intense part of the beam. Close to the RCU main FPGA on the RCU motherboard are the RCU support FPGA and the Flash memory device. A few centimetres away from the center of the beam line they are at the outskirts of the beam spot. A graphite collimator was therefore placed in front of the RCU motherboard to minimize the exposure to these devices. The opening of the collimator was centered in the beam line to keep the RCU main FPGA fully exposed to the beam. Figure 4.2 shows a picture the RCU in the beam line.

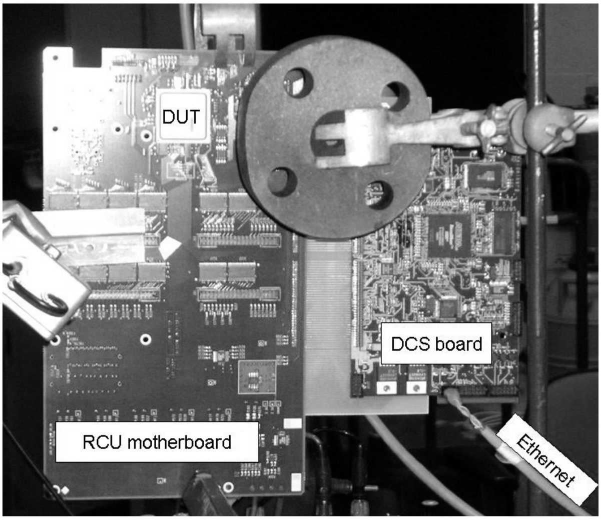


Figure 4.2: The RCU with the DCS board connected through the DCS extension card. The RCU support FPGA and Flash memory is located behind the collimator and is not visible. A laser is used to align the DUT and collimator in the beam line. The beam line is through the upper left opening of the collimator which is 1.5 cm in diameter. This is sufficient to cover the die inside the FPGA package.

4.1.1 Beam line configuration and monitoring

The beam flux is determined using a Thin Film Breakdown Counter (TFBC) [45][46]. During irradiation of the DUT[1], a scintillator is used as a relative beam monitor measuring beam stability and fluence. The fluence measurement is based on a pre-irradiation test calibration with the TFBC. More details on the OCL beam monitoring and experimental area setup can be found in [47] and [48].

4.1.2 Measuring the SEU cross section of the RCU main FPGA

The experimental approach used to measure the SEU cross section of the RCU main FPGA is based on a standard test methodology

1. Determine the energy and flux of the proton beam
2. Load the memory device with a known bit pattern
3. Irradiate the memory device with a given fluence

[1] Device Under Test

4. Read back content of memory device and count number of inverted bits
5. SEU cross section per bit is then calculated by equation 4.3, where the N_{SEU} is the number of inverted bits detected and N_B is the number of bits read back and checked.

This flow is referred to as static testing in the JEDEC standard [13]. If instead the memory is continuously checked during irradiation it can be referred to as dynamic testing. For the RCU reconfiguration network it is possible to apply both static and dynamic testing. Dynamic testing has the advantage that an SEU can be corrected immediately after it has been detected and thus prevent accumulation of SEUs. However, if the effects related to accumulation of SEUs is of interest, static testing has to be applied. Figure 4.3 shows the flow diagram of the dynamic test procedure used for measuring the SEU cross section of the RCU main FPGA. Except

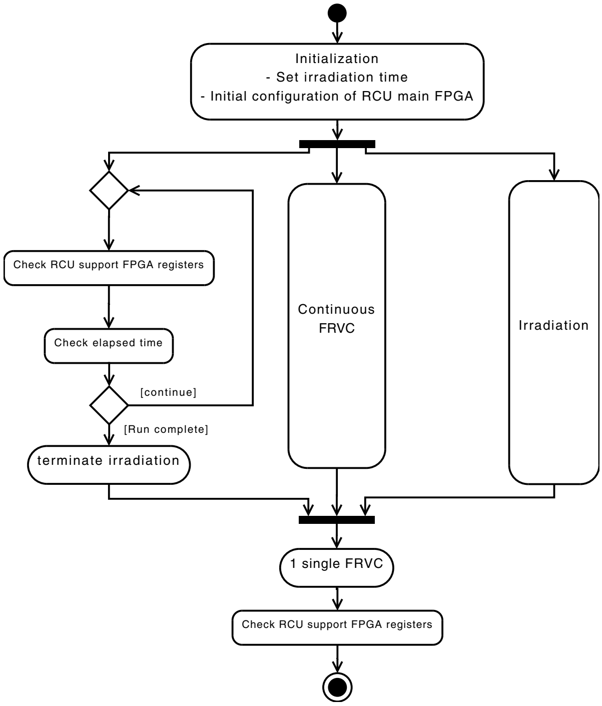


Figure 4.3: Flow diagram of the dynamic irradiation test procedure for the RCU main FPGA.

for starting and stopping the beam, the procedure is automatically controlled by DCS software. At first an initial configuration is carried out to load a pattern of bits to the configuration memory. Parallel to irradiating the RCU main FPGA, the RCU support FPGA continuously reads back the configuration memory to check for SEUs. At regular intervals this information is read by the DCS software, timestamped and stored in a log file.

In the JEDEC standard [13], the word "dead-time" refers to the time between a read of a memory location and a subsequent write to that location. Any upset that occurs in this time window will not be detected since it is overwritten. The total "dead time" during an irradiation test must be small compared to the total irradiation test time. If not, the SEU cross section will be erroneously low. For the RCU main FPGA the "dead-time" is the time it takes to read a single frame plus the time it takes to write a single frame. Using the respective numbers from table 3.1 this gives approximately 350 μs. The total "dead-time" for an irradiation cycle is then

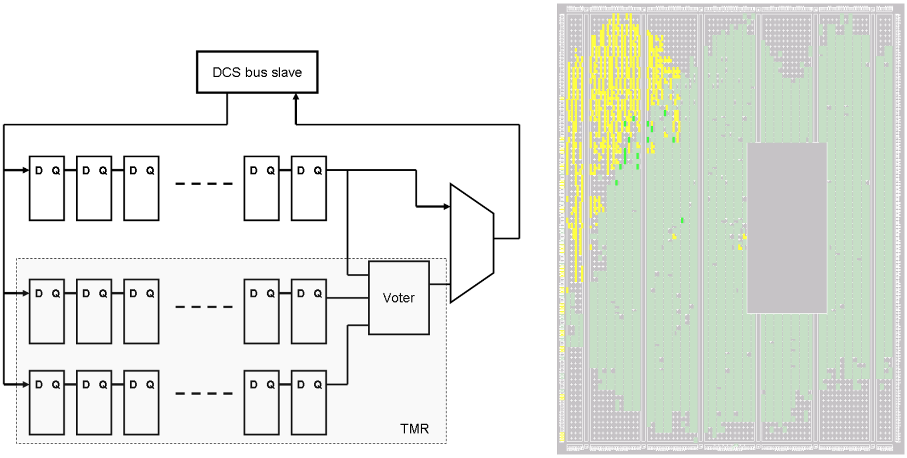
$$t_{dead} = 350\mu s \cdot N_{SEU} \tag{4.4}$$

4.1.3 Measuring the mitigation effect of the reconfiguration network

The purpose of the reconfiguration network is to correct SEUs in the configuration memory of the RCU main FPGA. Before an SEU can be corrected it has to be detected. This means that an SEU will be present in the system from the moment it is detected until it has been corrected by a reconfiguration. Its existence is limited by the best case and worst case correction times. The worst case correction time is the time it takes to carry out the FRVC cycle. In case an SEU occurs in the last frame of the configuration memory at the same time as the FRVC procedure starts on the first frame, it will take 150 ms before the SEU is corrected. The best case correction time is when the FRVC procedure enters a frame where an SEU has just occurred. This SEU will immediately be corrected and only exist for the time it takes to read and reconfigure that frame, which is 350 μs. Summarized, in the RCU main FPGA an SEU has a potential life time spanning from 350 μs to 150 ms. Considering that the FPGA operates on a 40 MHz clock (25 ns clock cycle), it is clear that an SEU has the potential to cause operational failures before it can be corrected. In [11] it has been suggested that an operational failure, if existing for more than a few tens of ms, has the potential to abort the ongoing run. It is therefore important to ensure that the probability of having an operational failure is minimized. Reconfiguration can not prevent an SEU from occurring, but it can reduce its life time. Futhermore, it prevents accumulation of SEUs in the configuration memory. Given that the reconfiguration time is considerably shorter than the average expected SEU rate, this reduces the probability of having more that one SEU present at any time. By carefully implementing additional mitigation on the FPGA user design level, an operational failure caused by this single SEU can be masked out. Some of the irradiation tests carried out during this thesis work were dedicated to study this effect.

Firmware test design

The firmware test design used during the irradiation test was a basic shift register extended with a configurable Triple Modular Redundancy (TMR) solution. TMR is a commonly used mitigation for FPGAs where three copies of the logic is implemented to operate in parallel [49][50]. A majority voter is placed at the output to identify the correct value. Figure 4.4(a) illustrates how this has been implemented in for the shift register test design. Configurable TMR means that the design is extended with a simple multiplexer that can either "turn off" TMR by forwarding the output directly from the primary shift register, or "turn on" TMR by forwarding the output from the voter. A simplified version of the *Memory Mapped Interface*



(a)

(b)

Figure 4.4: (a) Conceptual schematic of the shift register test design (b) Layout of the shift register test design from the Xilinx Floorplanner layout manager. Large green area is the triplicated shift register. Yellow identifies the DCS bus slave. Sharp green is the majority voter.

module described in section 3.2 is implemented as the DCS bus slave. The DCS bus slave takes care of the communication between the DCS test software and the shift register design.

The shift register used for the irradiation tests is 32 bit wide and 70 bit long. Three instances of the same shift register are used to implement the TMR. In total 72% of the total available register resources of the Xilinx Virtex-II Pro 7 FPGA is occupied by the TMR shift register. The corresponding value for the DCS bus slave is 6%. Figure 4.4(b) shows the layout of the implemented design. The greater part of the FPGA is covered by the TMR shift register while the smaller yellow part in the top left corner represents the DCS bus slave. The few sharp green colored boxes make up the majority voter logic.

Test strategy

The purpose of the irradiation tests is to see how the reconfiguration and TMR affects the operation of the shift register design. Figure 4.5 shows a conceptual flow diagram of the test procedure. Similar to the procedure of the SEU cross section measurements, an initial configuration of the RCU main FPGA is first carried out. The actual irradiation period is however divided in three parts. During the first part the RCU main FPGA, neither reconfiguration nor TMR is enabled. As a consequence, SEUs will accumulate in the configuration memory. In parallel, DCS software shifts through and reads out a bit pattern from the shift register in the RCU main FPGA. If the outcome is different than expected, and this difference is persistent, it is regarded as an operational failure due to an SEU in the configuration memory. For the second and intermediate period, reconfiguration is activated by enabling continuous cycles of FRVC. For the first round of FRVC, the number of accumulated SEUs will be counted and corrected by the RCU support FPGA. Any errors in the readout of the shift register will consequently be corrected as well. If an SEU should occur during the second period, it may still cause a an operational failure in the shift register, but due to the reconfiguration it will only be present for a short time. For the third and last period the TMR option is enabled. The outputs of the shift register are now fed through the majority voter and compared with the identical outputs of the two extra TMR shift registers. If an SEU causes an operational failure in one of the shift registers, the majority voter will mask this failure out, and it will not be visible to the DCS software. If the next failure occurs before the first has been fixed, it can only be masked out if it effects the already failing shift register. In case it affects one of the two other shift registers, the majority voter will vote through the incorrect value. To avoid this situation it is important to keep the rate at which the reconfiguration is carried out at a higher frequency than the expected SEU rate. Still, there may be situations were a failure caused by a single SEU can not be corrected. For instance, if all shift registers share the same clock tree, an SEU in the root of this tree will be spread to all shift registers. When doing TMR solutions of sequential logic, isolating the replicas in three different clock domains will therefore improve the failure rate. Moreover, an SEU related to the voter logic can of course not be corrected. Considering the small size of the voter logic compared to the logic it is meant to protect, its probability of failure is relatively small.

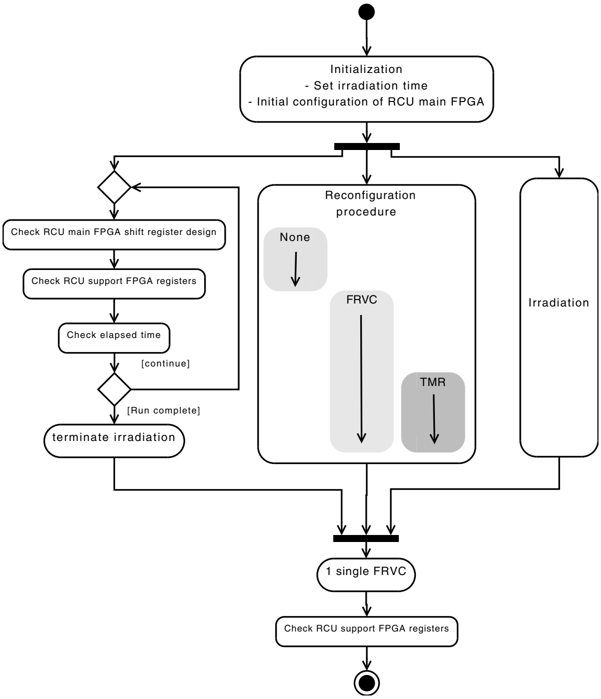

Initialization
- Set irradiation time
- Initial configuration of RCU main FPGA
Check RCU main FPGA shift register design
Check RCU support FPGA registers
Check elapsed time
[continue]
[Run complete]
terminate irradiation
Reconfiguration procedure
None
FRVC
TMR
Irradiation
1 single FRVC
Check RCU support FPGA registers

Figure 4.5: Flow diagram of the dynamic irradiation test procedure for the shift register test design.

4.2 Results

The irradiation tests of the RCU main FPGA are divided in two main irradiation test periods. Each period contains a number of individual runs using the same experimental setup. The main purpose of the irradiation tests were to measure the SEU cross section and the mitigation effect of reconfiguration network. However, no high current event in need of a power cycle to be restored, indicating an SEL, was detected during the irradiation tests.

4.2.1 SEU cross section

In total 61 individual runs were carried out with fluxes in the order of $10^6 - 10^7$ p/(cm^2 s). At these fluxes the SEU count per second ranged from 0.2 - 2.5. That is, the average period between an SEU is from 0.4 - 5 seconds, which is at least tree orders of magnitude higher than the dead time specified in equation 4.4. A correction due to the dead time will be less than 0.1% and can therefore be neglected. One cycle of FRVC is measured to approximately 150 ms, see table 3.1. The probability of having more than 1 SEU in the configuration memory at any time during a run is therefore small. The duration of the individual runs were typically 100-1000 seconds. During almost 7 hours of irradiation, 20750 SEUs were detected. In total 936 frames of 424 bytes each were continuously read back and checked. Only the BlockRAM frames were not checked for reasons explained in section 3.1.4. Figure 4.6 shows that the SEUs seems to be evenly distributed in between the 936 frames. This indicates that the FPGA has been fully covered by a uniform beam. As explained in section 4.1.3 some of the runs were dedicated to study the mitigation effect of the reconfiguration network. During these runs continuous FRVC could be

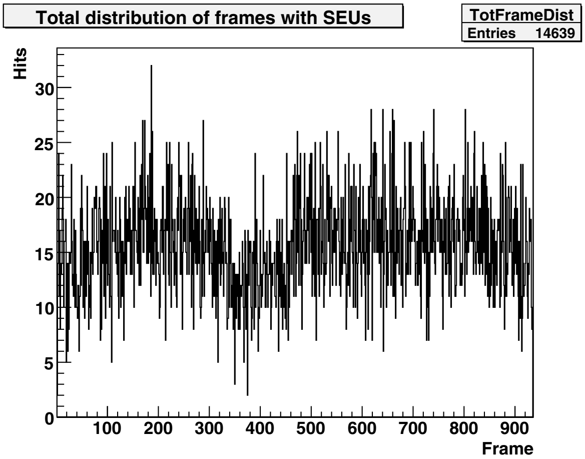
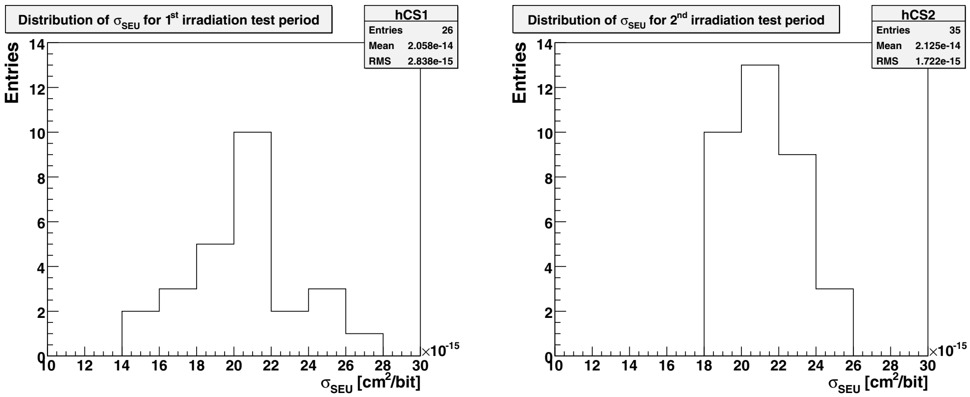


Figure 4.6: Frame distribution of SEUs.

disabled for fixed periods of time which resulted in accumulation of SEUs. When corrected during the first cycle of continuous FRVC, it is not possible to identify each single SEU to a frame number. This explains why the number of entries in the frame distribution plot is smaller than the total number of SEUs for all runs.



(a)

(b)

Figure 4.7: Histograms showing the distribution of the measured SEU cross section for test periods 1 (a) and 2 (b)

The SEU cross sections are calculated for the individual runs and finally binned in two histograms, figures 4.7(a) and 4.7(b), representing the first and second test period[2]. Even though the mean results correspond well, there is a wider distribution in the results from the first test period. This may indicate differences in experimental factors between the two periods. While the positioning of the DUT, TFBC and scintillator were identical for both periods, the alignment and tuning of the beam may have given slightly different beam characteristics.

A single event upset has a statistical nature and is expected to be proportional to the beam flux. That is, if the beam flux increases it is expected that the SEU count will increase, and vice versa. It could therefore also be used as beam flux monitor like the scintillator. Changes in the beam flux should be reflected in both the scintillator counts and the SEU counts. The correlation plot in figure 4.8 shows this linear dependency which also confirms that the scintillator is well suited as a relative monitor.

When trying to fit a straight line to each of the two periods of counting data, a small mismatch can be seen. The SEU cross section calculations are based on

[2] The individual results are listed in the tables of appendix C

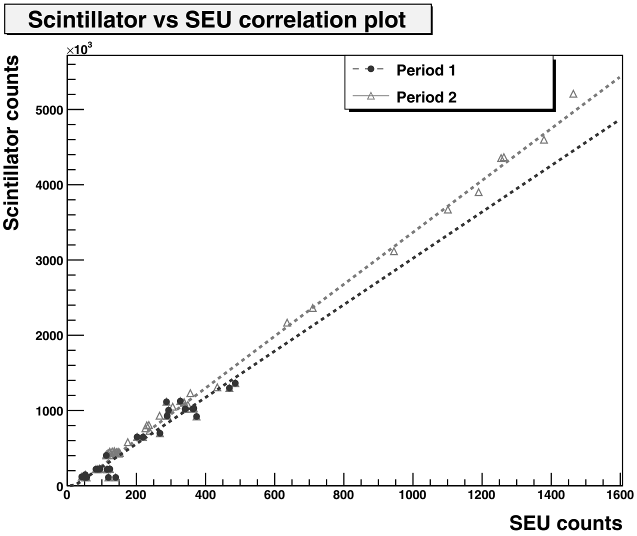


Figure 4.8: Correlation between scintillator and SEU counts.

the calibrated relationship between the TFBC and the scintillator. Therefore, if the correlation plot of the two periods do not exactly overlap, this can be related to the difference in the SEU cross section distribution. In figures 4.9 and 4.10 the SEU and scintillator counts are plotted in chronological order as a function of the run number. Each curve is normalized to the time of the run and to the highest detected count value of all the runs. This makes it possible to compare the curves in a better way. During the second period there is a very close relationship between the SEU and scintillator counts. For the first period the general trend is still present but there is a mismatch for individual runs. In particular this is evident for the first few runs. This is a probable explanation for the wider distribution of the SEU cross section during the first test period. Regardless of this difference, the mean result and standard deviation of the two periods overlap. The results can therefore be merged into a single histogram as shown in figure 4.11, and giving an SEU cross section of $2.1 \cdot 10^{-14} \pm 0.2 \cdot 10^{-14}$ cm^2/bit. The bin width is set to reflect the standard deviation.

Comparison to other results

The SEU cross section for the Xilinx Virtex-II Pro have previously been determined by others using both neutron and proton beams. For terrestrial application which

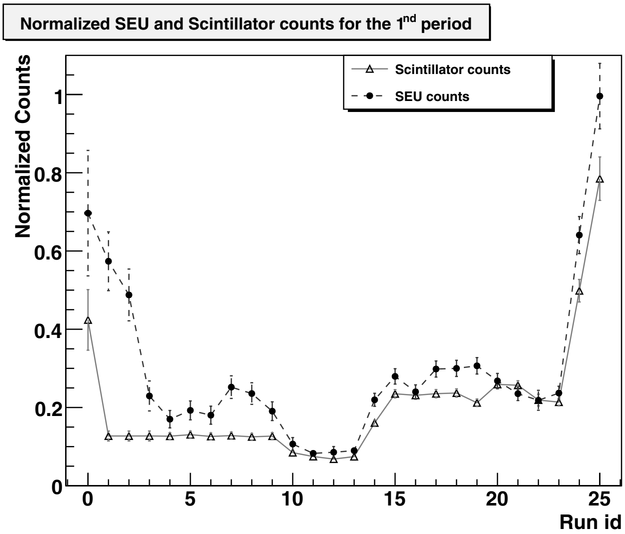


Figure 4.9: Normalized scintillator and SEU count curves for test period 1.

is the focus of the Rosetta study in [17], the Xilinx Virtex-II Pro has been tested using a neutron beam with energy distribution similar to the atmospheric (Hess) spectrum (1 MeV to 600 MeV). The reported SEU cross section for neutron energies above 10 MeV is $2.98 \cdot 10^{-14}$ cm^2/bit. In [16] a 63.3 MeV proton beam has been used to measure a SEU cross section of $3.68 \cdot 10^{-14}$ cm^2/bit. Both these results are higher than the result measured using the 29 MeV proton beam at OCL. This is mainly explained by the low energy of the proton beam used at OCL. From the point where the proton beam leaves the beam pipe at the OCL experimental setup, it travels through approximately 166 cm of air before it hits the surface of the FPGA. To reach the area of sensitive devices within the FPGA, the protons additionally have to travel though a 450 μm thick copper lid and the 800 μm thick silicon substrate. This simple geometry setup was simulated in Fluka to study the attenuation of the beam energy. Figure 4.12 shows that within a distance of approximately ±100 μm relative to the sensitive area, the beam energy has been reduced to approximately 15 MeV. It is therefore fragments produced by non-elastic nuclear interactions induced by protons at this energy that may lead to SEUs. From figure 2.11(b) it can be seen that at 15 MeV the probability of producing α-particles is a factor 6-7 lower than at 63.3 MeV. Furthermore, the possible energy transfer to the produced fragments is less than at 63 MeV. As can be seen from figures 2.3(a) and 2.6, the typical recoil energy is already below the maximum ionization energy

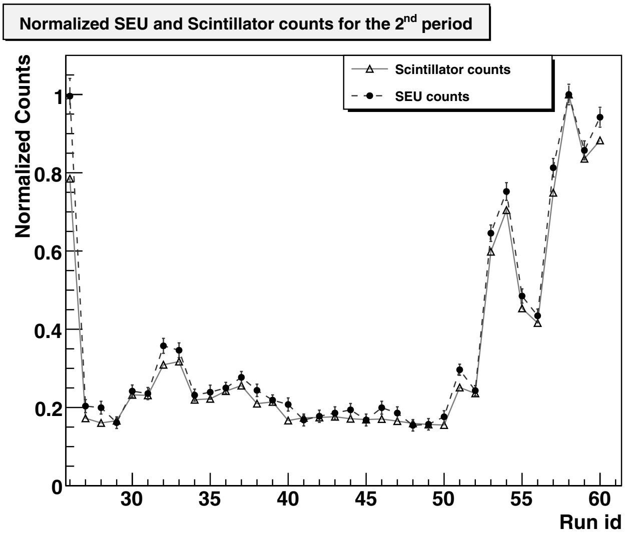


Figure 4.10: Normalized scintillator and SEU count curves for test period 2.

for a primary beam energy of 100 MeV. Thus, reducing the energy transfer and thereby the recoil energy, results in a lower dE/dx which in turn means a lower SEU probability. This effect is therfore expected to dominate even though the nonelastic interaction cross section increases slightly at lower energies before finally dropping to zero.

4.2.2 The effect of the reconfiguation network under irradiation

To study how the shift register behaves under irradiation a special output error plot is produced. If the bit pattern being shifted through differs from the expected value at the output of the shift register, this indicates an operational failure. In figures 4.13(a) and 4.13(b) the status of each output is plotted as a function of time for two individual runs. An erroneous output is indicated by an entry (black dot) in the plot at the corresponding output number on the y-axis. If an error is persistent, a continuous number of black dots can be identified as a continuous line for the corresponding output. During the 200 first seconds of irradiation, the shift register design runs without any mitigation. That is, both reconfiguration and TMR is disabled. This is clearly shown in the plots as the number of erroneous outputs increases with time. After 200 seconds continuous FRVC is enabled and the configuration memory is reconfigured.

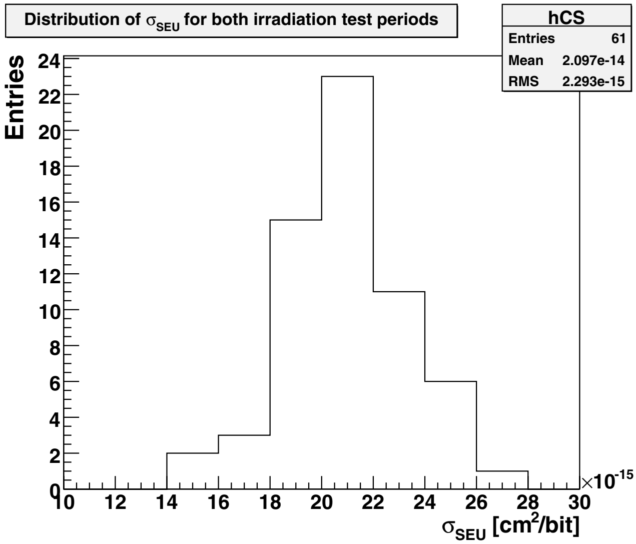


Figure 4.11: Histogram showing distribution of the measured SEU cross sections.

Both the number of scintillation counts and the number of SEUs detected by the RCU support FPGA, are continuously monitored during the test. Normalized scintillator and SEU counting values are plotted as a function of time in figures 4.13(c) and 4.13(d). For the first 200 seconds no SEUs are detected because the FRVC procedure of the RCU support FPGA is not enabled. At the moment FRVC is enabled, all the accumulated SEUs are detected and corrected. This explains the sudden increase in SEU counts after 200 seconds. Both curves show a strong linear dependency which is a strong indication of a stable beam and the stochastic nature of SEUs. Even though SEUs in the configuration memory are continuously corrected, the output error plot still shows that errors are detected in the shift register. As explained in section 4.1.3 this can be due to the fact that an SEU can exist long enough to corrupt the output of the shift register. When it finally is corrected, the output returns to giving the correct value and the operational failure becomes temporary. It should be noted that a single SEU can induce errors in several shift register outputs at the same time. The number of corrupted outputs is therefore not directly linked to the number of SEUs.

During the last 200 seconds of the irradiation test, the TMR option is enabled. Given that an SEU affects the outputs of only one of the TMR replicas, the majority voter should mask out the erroneous outputs of the corrupted replica. The result should be an error free shift register. Looking at the plots, there is a clear difference

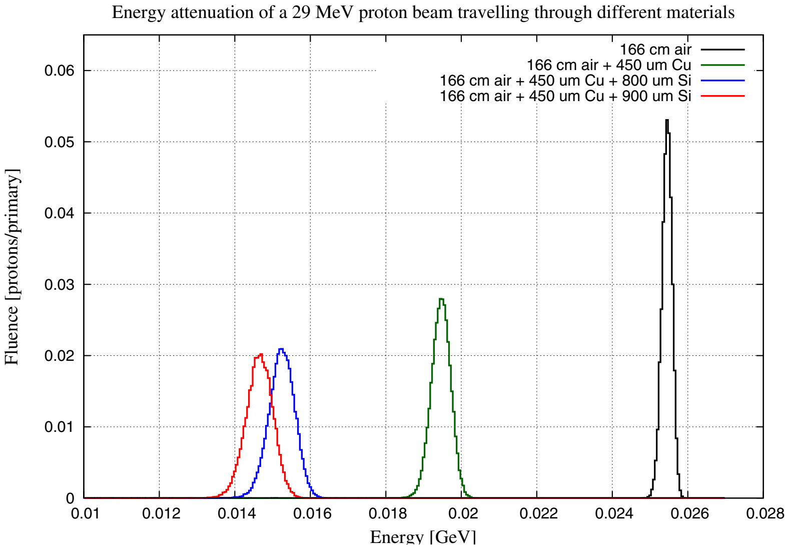
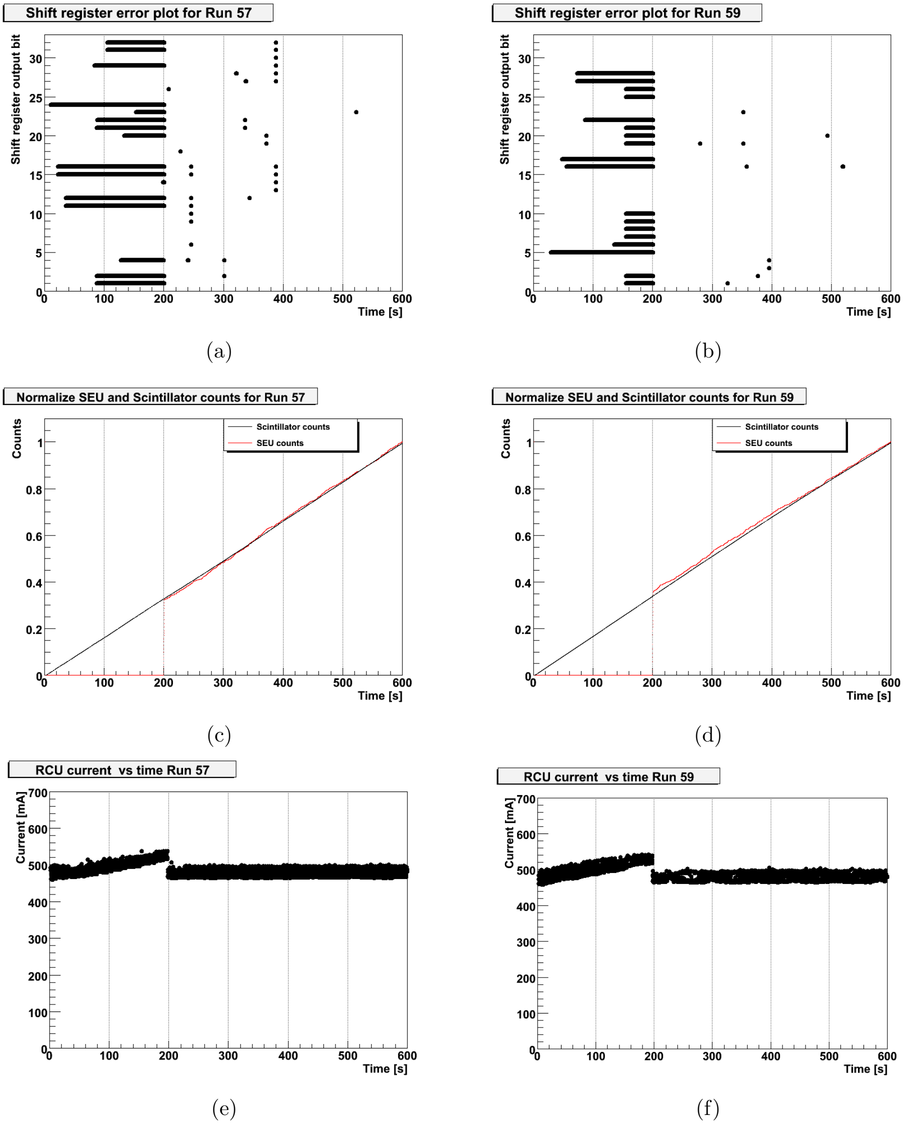


Figure 4.12: Energy attenuation simulation for a simple geometry representing the irradiation test setup. At the device entry point the beam energy has been reduced to 26 MeV while after 450 μm of copper and 800 μm the beam energy has been reduced to approximately 15 MeV.

between the 200 first seconds and the 200 last seconds of the runs. There is also a visible improvement from from the middle period to the last period. This shows that the TMR combined with continuous reconfiguration (or scrubbing) can improve the failure rate of a design, and consequently the mean time between failure. Still, a few errors are still seen during the last period. This is believed to mainly be due to how the TMR is implemented. Because the main purpose of the irradiation test was to get a preliminary indication on the effect of the combined reconfiguration and TMR mitigation solution, a minimum of effort was put into improving the TMR solution. For instance, the three shift register replicas are heavily interwoven and not physically separated in the layout. They therefore also share that same clock tree. Because of this, a single configuration bit may control resources connected to more than one replica of the shift register. Consequently, the TMR solution is likely to fail for an SEU in this type of configuration bit.

During the second irradiation test period, eight runs were dedicated to the shift register mitigation test. The result can be quantified by comparing the average number of SEUs needed to cause a functional failure for the three mitigation scenarios. The number of erroneous outputs is continuously monitored during a run.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.13: The plots are produced from the individual runs number 57 and 59. (a) and (b) shows the error plot of the shift register output where a black dot indicates an erroneous output. (c) and (d) are normalized values of scintillator and SEU counts. (e) and (f) are the current consumption of the RCU. For the first 200 s the shift register is run without FRVC and TMR. From 200-400 s FRVC is enabled. From 400-600 s both FRVC and TMR are enabled.

A functional failure is defined as an increase in the number of erroneous output. Table 4.1 shows the summarized results. Since the beam flux is stable during a run, one would expect the same number of SEUs to accumulate within each of the three equally long periods. This is also confirmed by observation. Due to the limited number of runs the accumulated statistics for the number of functional failures is low. Nevertheless, the result indicates that the combination of FRVC and TMR gives an improvement in the average number of SEUs per functional failure compared to no mitigation. Due to the interwoven nature of the shift register it was expected to show failures when applying FRVC and TMR mitigation. It should also be mentioned that the DCS bus slave was not mitigated at all. As the full chip was irradiated, any SEUs in this part of the chip could therefore corrupt the readout of the shift register. However, the DCS bus slave utilizes roughly ten times less resources than the shift register.

Run Type	SEUs	FF	Average SEU/FF
No mitigation	2628±51	71±8	37±4
FRVC	2632±51	67±8	39±5
FRVC + TMR	2599±51	14±4	186±50

Table 4.1: The average number of SEUs needed to induce a functional failure (FF) are calculated for the three different mitigation scenarios. The uncertainties are produced from counting statistics only.

Applying only FRVC alone does not improve the result compared to no mitigation. The readout frequency of the shift register is approximately 100 Hz, and thereby higher than the FRVC frequency. This difference is sufficient for a failure in any node of the shift register to propagate a number of nodes before the corrupted node is corrected. It will therefore be detected as a temporary failure.

During irradiation, when accumulating SEUs, the current consumption of the RCU was observed to increase. Immediately after reconfiguration was enabled, it dropped down to its initial level as can be seen in figures 4.13(e) and 4.13(f). This is an effect which also has been reported in [51] where it is explained to be due to internal contention by the accumulated upsets. Since not all the configuration bits may be used for a given design, a number of bits remains "unprogrammed". In case these bits are "programmed" by an SEU, some of them might for instance connect the clock tree to unused logic and induce more activity which again increases the current consumption. When corrected by a reconfiguration, this unwanted activity is removed and the current consumption is reduced to its initial level. If the current consumption did not decrease after a reconfiguration, this could indicate a Single Event Latch-up. This situation was not observed during the two test periods.

The qualitative result of the mitigated shift register irradiation test, is that if carefully implemented, the combination of reconfiguration and extra design level mitigation can be used to improve the failure rate the final RCU firmware design.

4.2.3 Total ionizing dose

Total dose effects are typically related to malfunction due to long term energy deposition in active semiconductor regions. During the OCL irradiation tests the one single FPGA was exposed to approximately $4 \cdot 10^{10}$ protons. This corresponds to a dose[3] of approximately 160 Gy which is significantly higher than the expexted total dose in ALICE. The FPGA was fully functional after the irradiation tests, and no significant change in the current consumption was registered except for that related to the temporary SEUs.

4.3 Predicting the SEU rate in the TPC radiation environment

Based on simulations of the TPC radiation environment the hadron flux above 10 MeV was calculated and presented in tables 2.1 and 2.2. The SEU cross section is a measure for the probability that an incoming particle will induce an SEU. Multiplying the SEU cross section by the hadron flux and the number of configuration bits[4], therefore gives an estimate for the SEU rate per second and per FPGA.

$$N_{SEU} = \sigma_{SEU,bit}(E) \cdot Flux \cdot N_B \tag{4.5}$$

As previously discussed, the OCL result is probably on the low side and would therefore slightly underpredict the SEU rate. The following calculations are therefore based on using the SEU cross section reported for the 63.3 MeV proton beam in [16]. The worst case SEU rate is calculated to $2.4 \cdot 10^{-5}$ SEUs/(FPGA s) for the innermost ring of RCUs on the absorber side. A data taking run in ALICE is expected to last four hours. Using the worst case estimate this gives 75 SEUs for an ALICE run counting all 216 FPGAs together. This estimate can be somewhat moderated to 42 SEUs by instead applying the differentiated flux over all the RCU scoring rings. As discussed in section 2.2.3 not every SEU will result in a measurable functional failure in the user design of the FPGA. Given the example of the shift register design, an average of 37 SEUs were needed before a functional failure was

[3] See appendix C.1.1 for dose calculation

[4] $936 \cdot 424 \cdot 8$ configuration bits are used for the Xilinx Virtex-II Pro 7. This corresponds to all the frames in the configuration memory except for the BlockRAM frames.

detected. To predict the expected functional failure rate of the FPGA, the SEU rate therefore has to be scaled down. The shift register result can however not be applied to the case of the final data acquisition user design being implemented in the the RCU main FPGA. This design is of higher complexity and separate testing is needed to determine the corresponding scaling factor. Nevertheless, a worst case prediction can be made as Xilinx [17][20] recommends a conservative scaling factor of 10. Based on the previous calculations 4-8 functional failures can then be expected per run.

4.4 Summary

The SEU cross section for the configuration memory of the RCU main FPGA was measured to $2.1 \cdot 10^{-14} \pm 0.2 \cdot 10^{-14}$ cm^2/bit in a 29 MeV proton beam at OCL. This is a factor 1.5-2 lower than the result reported for a atmospheric neutron spectrum in [17] and a 63.3 MeV proton beam in [16]. The reason is explained to be that at 29 MeV the energy is effectively attenuated to around the threshold for non-elastic interactions. This leads to both fewer fragments produced, and in particular recoil fragments with lower ionization power. Even though higher energies are generally recommended for accurate determination of SEU cross sections, the 29 MeV beam at OCL is still capable of inducing acceptable upset rates well suited for the specific testing applications described in this thesis work.

For the TPC radiation environment the worst case SEU rate was calculated to $2.4 \cdot 10^{-5}$ SEUs/(FPGA sec). This result corresponds to the location of the RCUs closest to the beam line on the absorber side. It is expected that the SEU rate will slightly decrease moving out towards the outermost RCUs. Considering the result of the simulated radiation environment and counting all 216 FPGAs together, one can expect to detect 40-80 SEUs during a 4 hour run in ALICE. This corresponds to approximately 4-8 functional failures during the same run if a conservative scaling factor of 10 is used. A concluding remark is therefore that experiencing a failure in the readout functionality of an RCU main FPGA is a realistic scenario during an ALICE run. It can however not be specified if the failure is of a severe or negligible type. Only testing of the final RCU main FPGA user design can give indications of expected failure types.

Chapter 5

Implementing Fault Injection for the RCU main FPGA

Fault injection (FI) is an analysis technique that injects faults and errors into a system in order to see what impact they may have. It is a method to simulate errors that can occur during normal operation and thereby learn how the system behaves when something goes wrong. A basic introduction to fault injection and how it is applied to software is given in [52]. When related to FPGAs, fault injection can for instance be applied during simulation of VHDL behavioral-level models as described in [53]. Due to the increased use of FPGAs within radiation exposed environments, fault injection can also be used to simulate the effects of SEUs in the configuration memory. This type of fault injection is more often referred to as accelerated beam testing or irradiation testing. Accelerated beam testing captures the actual physical mechanisms that are responsible for an SEU. It is the only method of testing to determine how sensitive an SRAM cell is to a certain type of radiation. That is, to determine the SEU cross section. However, for extensive testing to study the behavioural effect in the FPGA user design, accelerated beam testing can be inconvenient. To achieve significant coverage and statistics, a complex set of test vectors or stimuli is needed. This can be time consuming and therefore not compliant with the fact that access to accelerated beam facilities often is both limited and costly. Also, due to the stochastic nature of SEUs, it is difficult to achieve controlled fault injection with accelerated beam testing. That is, limited by the accelerator specifications, the upset rate can to some extent be controlled, but the bit location is still random.

The work presented in this chapter is motivated by applying partial reconfiguration as a utility to carry out fault injection. This method does not require access to accelerated beam facilities, and it can be carried out in a controlled fashion by simply manipulating the configuration bitstream of the FPGA. It has also proved to be successful in a number of other cases [54][55][56]. Even though a more general injection platform like the Flipper tool [57] exists, it can not directly be applied to the RCU main FPGA during operation in the data readout path of the TPC detector. To achieve this, the fault injection solution has to be implemented as part of the already existing hardware system. This chapter will describe how this done in addition to presenting the resuls of a test case study carried out to validate this implementation.

The terminology used to describe fault injection throughout this chapter varies slightly. Fault injection will also be referred to as injecting a bit flip or injecting an error in the configuration memory. In some situations this terminology better describes the process. The impact fault injection has on the behaviour of the FPGA user design is referred to as an operational failure or a functional failure.

5.1 Implementation

5.1.1 System design considerations

Initially fault injection was not included as a feature of the RCU reconfiguration network. Implementing it at a later stage meant that a major limitation had to be overcome. The preferred solution would be to add fault injection as an option in the configuration state machine of the RCU support FPGA. This solution was disqualified due to the limited available resources of the RCU support FPGA. Great effort had already been put into optimization in order to fit the design into the RCU support FPGA. Implementing additional functionality was simply not feasible. Also, since fault injection was introduced as a possible test method at the stage when system commissioning was started, redesigning the configuration controller to include fault injection would impose an extra delay. The alternative solution chosen was to take advantage of the available operational modes of the DCS bus. SelectMAP mode offers a direct connection from the SelectMAP configuration pins to the DCS embedded computer. Since a special Linux driver (Virtex driver) already had been developed to carry out initial configuration from DCS software, only a simple modification was needed to support partial reconfiguration. When writing either a scrubbing file or a single frame file, the special *Prog_b* line for clearing the configuration memory was not activated. This means that similar to partial reconfiguration (FRVC) done by the RCU support FPGA, the DCS software can write a frame of configuration data to the RCU main FPGA without interrupting the operation of the user design running on the FPGA.

The final solution is therefore a combination of software to control the fault injection procedure and switching of DCS bus modes, and the normal functionality of the RCU support FPGA to detect and correct the injected bit flip.

5.1.2 Fault injection procedure

In order to carry out the normal reconfiguration tasks controlled by the RCU support FPGA, a number of WriteFrame and ReadFrame files have to be produced and downloaded to the RCU Flash memory device. This has already been described in chapter 3. During fault injection the task of the DCS software is limited to only writing frames of configuration data to the RCU main FPGA. Therefore, the same WriteFrame files stored on the Flash memory device are also kept in the file system of the DCS embedded computer. When fault injection is applied, the DCS selects one of the WriteFrameFiles, reads the content, inverts one of the data bits, and finally writes the frame content containing a corrupt bit position to the configuration memory of the RCU main FPGA. The specific frame and bit location to invert, can of course be decided by the user of the software. In order to carry out this operation, the system has to be switched into the SelectMAP mode. After the frame content has been written, the system is again switched back into normal mode. In normal mode stimuli are given to the implemented user design in the RCU main FPGA to check for any abnormal behaviour. The response is logged for further analysis. Depending on whether the operator of the fault injection test wants to correct the injected bit, a command can be sent to the controller state machine of the RCU support FPGA to issue a single cycle of FRVC. If accumulation of bit flips are requested, the FRVC cycle is skipped. The procedure is illustrated in figure 5.1 and can be repeated for every requested bit flip.

5.1.3 Software classes

Four main software classes were developed to make it possible to run fault injection from the DCS embedded computer. A brief description of each class will be given in this section and a corresponding class diagram can be found in appendix D.

XilinxTest

This is the top level class of the fault injection software. It contains methods related to different fault injection test procedures. In the present version, the following methods are implemented based on slightly different test strategies.

- *FlipAllBits* - This method flips every single bit within a frame one by one in a sequential order. By specifying the frame number of the first and last frame,

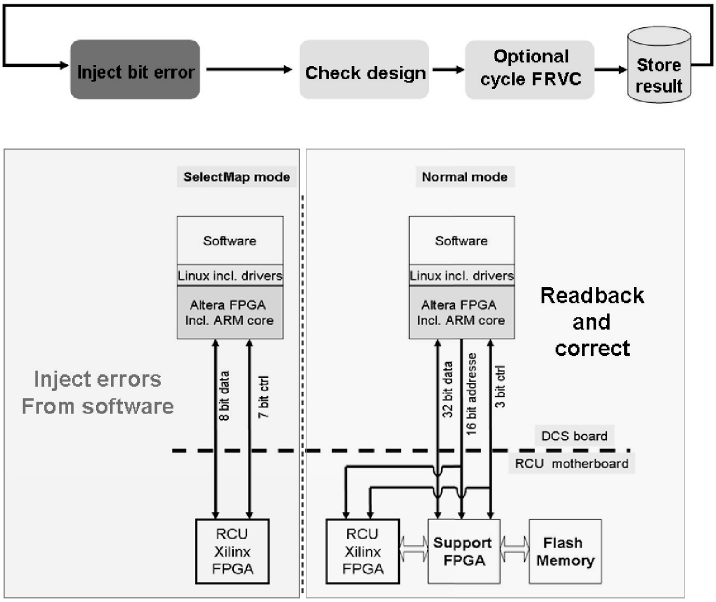


Figure 5.1: Direct access in SelectMAP mode is used to inject errors in the configuration memory of the Xilinx FPGA. The system is then switched back into normal mode to check the design and reconfigure the configuration memory. FRVC: Frame by frame Readback, Verification and Correction.

this can be carried out over a number of frames. In case fault injection should be carried out on a number of frames in a non-chronological order, the method can be called several times for a single frame each time. The purpose of this method is to carry out fault injection in a systematical fashion. It can for instance be used to map sensitive bits in a given design. Of course, methods for testing the behaviour of the FPGA user design has to be added as well.

- *RunToFirstFailure* - The basic purpose of this method is to empirically determine the average number of fault injections needed in the configuration memory before a failure is detected in the operation of the FPGA user design. It is specifically designed for the case study presented in section 5.2 but is not limited by this and can be used for other case studies as well with small adjustments.

- *FlipSingleBit* - This method is implemented to increase the flexibility of the fault injection software. It uses the *SetFlipData* and *doFI* methods of the *XilinxFaultInjection* class to inject a single bit flip in a specified frame of the configuration memory.

While the first two methods are tailor made to carry out a specific test procedure, the *FlipSingleBit* is a simple method doing one operation. It can for instance easily be utilized to make a small test program to manually inject a single bit flip. In addition to the public methods described above, the class contains a number of private methods. These are specifically implemented to carry out the test procedures in the *FlipAllBits* and *RunToFirstFailure* methods.

XilinxFaultInjection

This method takes care of the main operations related to injecting bit flips in the configuration memory of the FPGA. Before fault injection can be carried out, the user has to prepare a special configuration file. This file contains a list of the frames that will be used by the fault injection software. The location of this file is specified by the *SetFramePath* method. This is also the location of all the frame files. The use of a configuration file allows a more dynamic use of the fault injection software since the individual frames do not have to be hard coded. A limitation to this is that the specified frames in the configuration files have to correspond to the frame files loaded in the RCU Flash memory device. If not, a mismatch will exist between the frames where a bit flip can be injected and the frames that will be read back and corrected by the RCU support FPGA.

The main available methods in the *XilinxFaultInjection* class are listed below.

- *doFI* - This is the method that controls how the fault injection should be carried out. Its main tasks are:
  - Open and read the content of the specified frame file.
  - Locate and invert the bit value of the specified bit.
  - Write the content with the corrupted bit position to the configuration memory of the RCU main FPGA.

  Two options exist for deciding which frame and related bit to invert. In case the user wants to invert a specific bit, this can be done by first using the *SetFlipData* method, followed by calling the *doFI* method with the parameter *random=FALSE*. By setting *random=TRUE*, a random frame file will be selected from the available frame files specified in the configuration file. When the content of the frame file has been read into the *frameFileContent* table, a random byte location will be selected followed by inverting a random bit location within this byte. Accumulation of bit flips is supported by setting *accumulation=TRUE*. Whenever a bit value is flipped, the bit location information is stored in the *flippedBitTable*. If accumulation is activated for the subsequent bit flip, the bit locations already stored in this table will be inverted in the same fault injection cycle as well.

- *SetFramePath* - Sets the path to where the frame files are located in the DCS file system.
- *SetFlipData* - Sets the information related to the specific bit location of the fault injection.
- *ReadFrameConfigurationFile* - Reads the configuration file containing information about which frames that can be used for fault injection.

A frame can be addressed by using either a combination of the block, major and minor number, or by a single number identifying the frame by its order among all the utilized frames. A frame info file is used by the software to correlate the block, major and minor number to the single frame number, and get the information about how many frames are used for the fault injection.

As described above, accumulation of bit flips is supported by implementing a table keeping track of all the previously injected bit flips. The reason for this solution is to keep one set of WriteFrame files only. Alternatively the original WriteFrame file has to be used for the first injected bit flip, then the corrupted frame content has to be written to a new and corrupted frame file. The next time a fault injection is carried out and accumulation is activated, the corrupt frame file has to be loaded and updated with the additional corrupted bit.

Flipping multiple bits at the same time has not been specifically implemented in the present version of the software. However, only minor modifications are need to accomplish this. An additional method can be added to the *XilinxFaultInjection* method that allows to fill the *flippedBitTable* with the number of bit flips requested. By calling the *doFI* method with accumulation activated, all the bits specified in the *flippedBitTable* will be inverted in the respective frame. Applying this method, the effect of multiple bit upsets (MBU) can be studied. A limitation governed by the architecture of the Xilinx configuration memory is that only bits within the same frame can be flipped at the same moment in time. If multiple bit flips should take place between different frames, a short delay between the bit flips will be present. This is because the frame data input register in the Xilinx configuration state machine only can latch one frame of data into the configuration memory at a time.

SelectMapIF

The main task of the *SelectMapIF* class is to activate and control the operation of the Linux driver communicating with the SelectMapIF on the RCU main FPGA. It contains the public method *WriteFrameDataToDevice* that controls activities related to switching the DCS bus between normal mode and SelectMAP mode, opening and closing the correct device of the Virtex driver, and of course writing the frame content to the SelectMAP interface of the RCU main FPGA.

NormalModeIF

During normal mode operation of the DCS bus, the *NormalModeIF* class contains methods to simplify the communication which is needed to carry out fault injection. The standard communication tasks needed are

- *InitActelRegisters* - This method is called at the start of a fault injection to clear and set the command, status, error, and other registers into a default ready mode.
- *InitActelRegisters* - This methods reads the present value of all registers in the RCU support FPGA related to the reconfiguration processes. For instance, the number of SEUs detected and the frame number of the last frame where an SEU was detected. It is called by the *XilinxTest* class whenever this information is needed and stored in a log file for further analysis.
- *WriteReg* - General method that can write requested data to a user design register specified by the register address.
- *ReadReg* - General method that can read data from a user design register specified by the register address.

This class uses the *MessageBufferIF* module which is previously developed [58] to control the low level communication over the DCS bus during normal mode operation. Like for the *SelectMapIF* class, it is associated with a specific Linux driver to handle the communication with the hardware modules. The communication line of these two interfaces has previously been shown in chapter 3 figure 3.6.

5.2 Fault injection case study

A case study was carried out to validate the implementation of fault injection for the RCU main FPGA. To show the potential use of fault injection, a specific test procedure was developed. The basic purpose was to determine the average number of fault injections needed before a test design running on the RCU main FPGA experienced a functional failure. Except for some modifications, this strategy is identical to the test carried out during the accelerated beam test described in chapter 4. Therefore, a similar design as the shift register used for the accelerated beam test was used for the fault injection as well. This also allows to better compare the results from the fault injection results to the results from the accelerated beam test.

5.2.1 RCU main FPGA test design

For the accelerated beam test a 32 bit wide and 70 bit long shift register was used. As figure 4.4(b) shows the layout of the shift register covered almost the full FPGA. All the 936 frames covering the GCLK, IOB, IOI, CLB and BlockRAM interconnects were downloaded to the RCU Flash memory device and used during the reconfiguration process. Compared to writing a single frame of data from the RCU Flash memory device to the SelectMAP interface using the configuration controller of the RCU support FPGA, writing a frame of data from DCS software using the SelectMAP mode of the DCS bus is a significantly slower process. Writing the actual data frame is measured to approximately 1 ms, but with an additional software overhead of 60 ms. Compared to the approximately 180 μs it takes to carry out the same operation for the RCU support FPGA, this is a factor 340 slower. The cause of this overhead is not specifically identified, but is believed to be connected to the software accessing drivers and switching the DCS bus mode. Running the *FlipAllBits* method in the *XilinxTest* class for all 936 frames, the estimated total time of this procedure is approximately 185 hours or 1 week. This includes running cycles of FRVC (150 ms) to correct each injected bit flip. For this case study it was decided to reduce the test time by decreasing the number of frames used. Therefore, the length of the shift register was reduced to 8 bits. For this study, the purpose was to test the effect of the mitigation technique implemented for the shift register. From the resource point of view, the size of the shift register is comparable to the DCS bus slave. It is therefore expected that the contribution from the DCS bus slave will be significant, in particular for the mitigated versions of the shift register. Therefore, fault injection is only carried out for the shift register and not the DCS bus slave. This is achieved by carefully placing the modules in different frames using placement constraint settings. The shift register and majority voter is placed within 66 frames in the middle part of the FPGA, while the DCS bus slave is placed within 88 first CLB frames to the left. In addition to the shift register, the relevant IOB, IOI and GCLK frames were also tested (30 frames). In total 96 frames were exposed to fault injection.

5.3 Results

The main purpose of the case study was to validate proper operation of the fault injection implementation. One method of validation is to compare the results of the fault injection to the accelerated beam tests. Since fault injection is meant to be an alternative to beam testing when studying the behavioural effects of the RCU main FPGA user design, similar results should be required. A case study based on fault injections in random locations was therefore carried out. With only a few modifications the same shift register test design used during the beam tests were also used for fault injection.

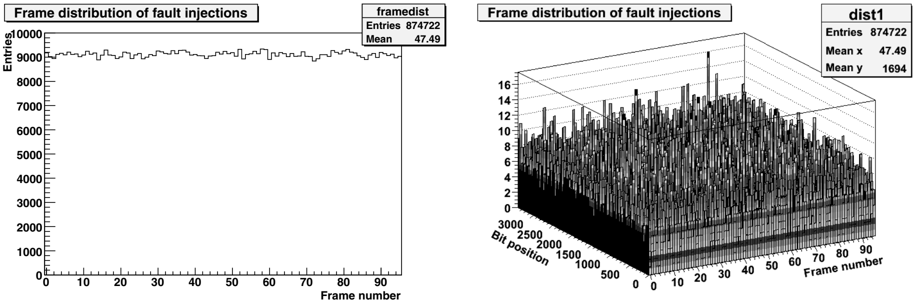
5.3.1 Summary and validation results

Table 5.1 summarizes the main observed results from the randomized fault injection runs. Fault injection was carried for similar mitigation scenarios as during the accelerated beam tests. Bit flips were injected in random locations of the utilized configuration memory until a functional failure in the shift register was detected. Both the number of fault injections and the number bit flips detected by the RCU support FPGA was then recorded. An initial configuration of the RCU main FPGA was then carried out and the test was repeated until in total 2600 functional failures were detected. In total 874722 fault injects were carried out over the 96 available frames. Figures 5.2(a) and 5.2(b) shows that an even distribution in frame and bit location has been achieved. Out of totally 325632 available bit locations, 28373 were not sampled during this test. This results in a total bit location coverage of approximately 90%. The individual coverage for each mitigation scenario is listed in table 5.1. To increase the coverage, the runs have to be either extended, or a systematic testing approach has to be taken. For this test randomized fault injections were carried out since this best compares to the accelerated beam testa.

Type of mitigation	Non	FRVC	TMR	FRVC + TMR	Total
Functional failures (FF)	2600	2600	2600	2600	10400
Injected bit flips	100223	102891	247300	424308	874722
Detected bit flips	95683	98056	235337	404391	833467
Bit locations not sampled	246672	245220	162708	97189	28373
Coverage (Bits sampled)	24%	25%	50%	70%	91%
Detected bit flips / FF	37±1	38±1	91±2	156±3	-

Table 5.1: Summary of all test runs for each of the four scenarios. The uncertainties are produced from counting statistics only.

Also, during the beam test in total 20750 SEUs were detected for in total 936 frames. This gives a bit location coverage of 0.7%.



(a)

(b)

Figure 5.2: 31478 bit positions are not covered in this distribution. This is approximately 10% of the all bit locations in the 96 frame files.

Nonexistent memory locations

If a fault injection can be detected during a cycle of FRVC by the RCU support FPGA, this is a clear indication that the technical part of the fault injection works properly. While this has clearly been successful, a certain discrepancy between the number of injected bit flips and the number of detected bit flips was observed. Of the total number of injected bit flips, 4.7% were not detected by the RCU support FPGA. In the configuration section of the Xilinx Virtex-II Pro User Guide [36], it is reported that the Virtex-II Pro has nonexistent memory locations. These are bit locations that are not connected to any logical resources of the FPGA and thus can not affect the behaviour of the user design if struck by an SEU. The Xilinx BitGen tool can produce a special mask file that contains a bitmap of these nonexistent memory locations. In addition it also contains mask bits for all the user memory (BlockRAM) bits. A '1' in the mask file indicates that a bit location should be masked out. An analysis of the mask file produced for the design used during fault injection was carried out and the result is given in table 5.2. The CLB66 entry gives the number of mask bits in the CLB frames used for the shift register design. When analysing the FRVC and combined FRVC and TMR mitigation run log files, it is possible to map the injected bit location which does not result in a detected SEU by the RCU support FPGA. These bits are referred to as stuck bits in the following and can be interpreted as nonexistent memory cells. The results are listed

Frame type	Mask bits	Number of frames	Total frame bits	% of mask bits
GCLK	13166	4	13568	97%
IOB1	128	4	13568	0.9%
IOI1	1904	22	74624	2.6%
CLB	178080	748	2537216	7.0%
CLB66	2040	66	223872	0.9%
IOI2	1904	22	74624	2.6%
IOB2	128	4	13568	0.9%
BRAMINT	113848	132	447744	25.4%
BRAM	800628	384	1302528	61.5%

Table 5.2: The table shows the number of mask bits for a given frame type. The numbers are produced from an analysis of the mask file produced by the Xilinx BitGen tool.

in table 5.3. This shows that for the GCLK and IOB1 frames, the percentage of stuck bits compares well with the percentage of mask bits for those frames. It is thus in the IOI1 and CLB66 frames the discrepancy is found. This indicates that some of the bit locations that is requested to be masked out by the mask file, can be accessed and therefore are existing memory cells. The assumption is then that these memory cells exist, but that they are not connected and therefore do not control any logical resources of the FPGA. These bits are referred to as soft bits in the following. To further investigate the discrepancy, the injected bit locations for the FRVC and the combined FRVC and TMR runs were compared to the bit locations in the mask file. If any of the injected bit locations corresponded to a mask bit location, a check was made to see if it was either a stuck bit or a soft bit. The results are listed in table 5.4. This shows that of the injected bit locations matching a mask bit location in the mask file, only 56.7% and 46.7% of the injected bits locations are detected

Frame type	Injected bits	Detected stuck bits	% stuck bits	Coverage
GCLK	21903	21126	96.5%	77%
IOB1	22160	210	0.9%	78%
IOI1	120484	1664	1.4%	77%
CLB66	362652	1561	0.4%	78%

Table 5.3: The table shows the relationship between the number of fault injections that were carried out and the number of times it did not result in a detectable bit flip. The bit location coverage was 77-78% which means that not all bit locations were tested.

Frame type	stuck bits	soft bits	% stuck bits	%soft bits	Mask bit coverage
GCLK	10131	0	100%	0	77%
IOB1	102	0	100%	0	80%
IOI1	815	622	56.7%	43.3%	75%
CLB66	739	852	46.4%	53.6%	78%

Table 5.4: If a fault injection was carried out in a location corresponding to a mask bit in the mask file, it was check whether fault injection resulted in a detectable bit flip or not. For the GCLK and IOB1 frames, only stuck bits were detected. In the case of the IOI1 and CLB66 frames, roughly half of the locations did not result in a bit flip while other half resulted in a detectable bit flip. The bit location coverage was from 75-80% which means that some mask bit locations were not tested.

as stuck bits for the IOI1 and CLB66 frames respectively. Accounting for both the detected stuck bits and soft bits, the discrepancy to the number of mask bits is no longer significant. The conclusion is therefore that the difference between the total number of fault injections and the total number of detected bit flips in table 5.1, is not caused by an incorrect implementation of the software, but by the nonexistent memory locations.

Comparison to accelerated beam test

When running fault injections without any mitigation enabled, in average 37 configuration upsets are needed to induce a functional failure in the design. Applying both FRVC and TMR the number of bit flips per fault injection is increased by a factor of 4. These numbers compares well to the corresponding results from the accelerated beam tests reported in section 4.2.2. In principle the fault injection and the accelerated beam tests ran the same test design on the RCU main FPGA. The only differences were the decreased shift register length, and that the DCS bus slave was not exposed during the fault injection tests. Because the shift register is approximately ten times larger than the DCS bus interface in the accelerated beam case, its contribution is expected to be relatively low. Comparing the results from the fault injection and the accelerated beam tests is therefore still acceptable. At least this is true when running non mitigated tests. For the runs using TMR or a combination of TMR and FRVC, the relative contribution from the DCS might have to be considered. In fact this can explain why the number of bit flips per functional failure for the combined FRVC and TMR case is higher in the accelerated beam test. Another explanation for this difference can of course be related to the poor statistics gathered during the accelerated beam test.

A fault injection run was also carried out in order to see if similar plots to the ones shown in figure 4.13 could be produced from fault injection. The result is seen in figure 5.3.1 and shows similar behaviour as for the accelerated beam tests. For the accelerated beam test results in figure 5.3.1 the SEU rate was approximately 2.4 SEUs/sec compared to 0.9 emulated SEUs/sec during fault injections. This explains why fewer erroneous outputs are seen during the same period of time for fault injection. The fact that the current consumptions also increases during fault injection, supports the explanation that this increase is caused by "programming" of unused bits, and thereby possibly increasing the activity in the FPGA. Due to the lower upset rate, the increase is less visible for this fault injection test. Still, there is a clear boundary at the point where FRVC is enabled and the injected and accumulated bit flips are corrected.

The overall results validates that the fault injection implementation works correctly. It can therefore be used as an alternative to extensive beam testing when studying the behavioural effect of SEUs on the final RCU main FPGA user design.

5.3.2 Applications

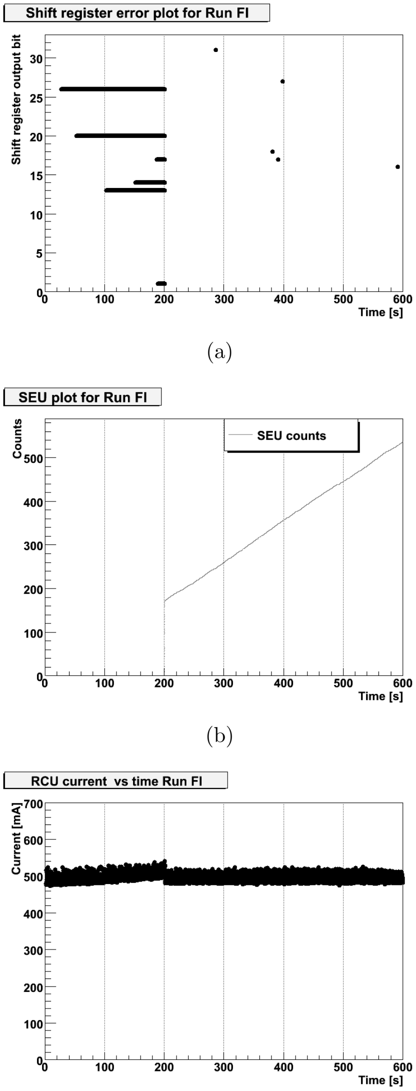
To allow flexible connections of the logical resources in an FPGA requires an extensive network of routing lines. For a specific design only a fraction of the possible routing solutions are used. Also, depending on the complexity of a design, not all of the available logical resources are used. The result is that a large number of configuration bits remain "inactive". That is, they are not controlling any resources that are used for the implemented design. In [20] it is reported that only 1 out of every 10-40 configuration memory cells are utilized in a typical design. This number reflects the number of SEUs needed to cause a functional failure in a design. When studying the impacts of SEUs, Xilinx [17] refers to this number as the single event upset probability impact (SEUPI). If combined with the SEU cross section, it can be used to predict the expected functional failure rate for a design in a given radiation environment.

$$\sigma_{FF} = \sigma_{SEU} \cdot \frac{1}{SEUPI} \tag{5.1}$$

This scaling factor is also briefly discussed in chapter 4.

Fault injection is a well suited method to determine the SEUPI number for the final RCU main FPGA, and thereby estimate the expected failure rate of the design for a given radiation environment. Since the final RCU main FPGA design was not available during testing of the fault injection solution, the shift register design was used to show the potential of fault injection.

Table 5.1 lists the average number of detected bit flips per functional failure for



(a)

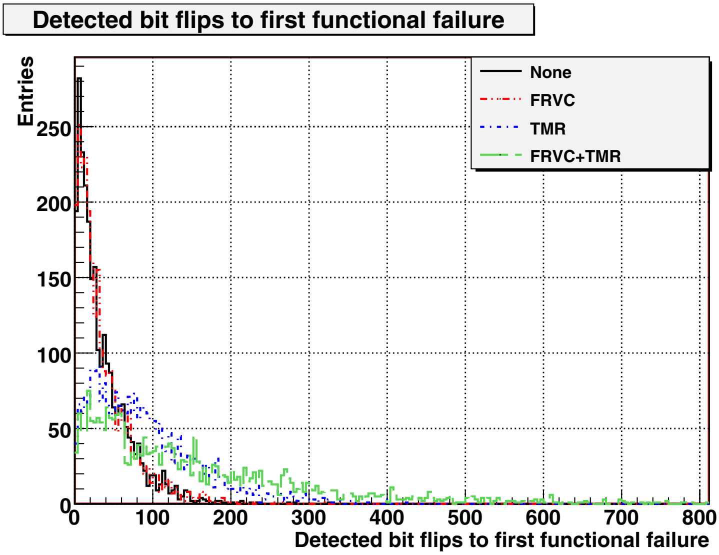
(b)

(c)

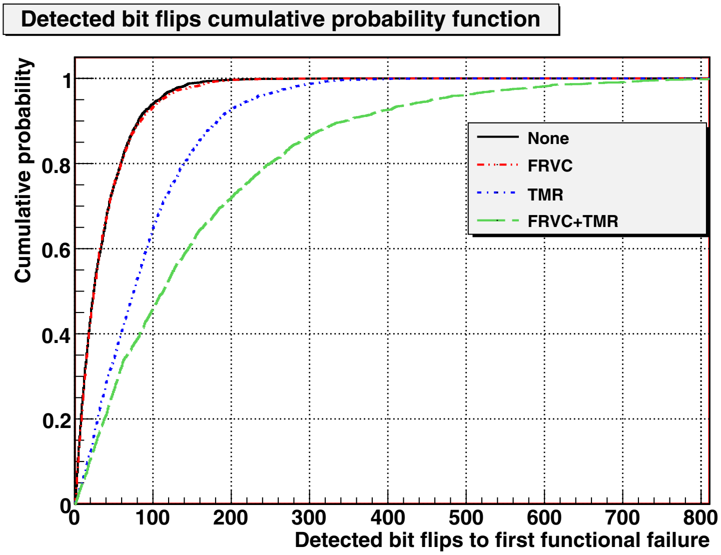
Figure 5.3: a) Shift register error plot. b) Detected bit flips by the RCU support FPGA. c) RCU current consumption during fault injection.0-200s: no mitigation, 200-400s: FRVC, 400-600s: FRVC+TMR.

the different mitigation approaches of the shift register design. As can be seen the number for the non-mitigated design compares well to the typical number of used configuration memory cells in [20]. However, the average number is computed from a distribution, and with a sufficient amount of data, a frequency distribution plot can be produced for the different mitigation scenarios applied to the shift register. The result is shown in figure 5.4(a). Even though the combined TMR and FRVC mitigation approach has increased the average SEUPI number, there are still a considerable amount of entries for low SEUPI numbers. This can be explained by the simplistic implementation of the TMR mitigation. In addition no approach was taken to mitigate any failures induced in the GCLK, IOI, or IOB frames. A weak link in the TMR solution is of course the majority voter itself. An approach that combined can improve the result is to move the majority voter out of the FPGA. This will also include using triplicated output pins and could thereby reduce the influence of SEUs in the IOI and IOB blocks. For a 32 bit wide shift register a solution like this may be difficult or even impossible due to limited available I/O pins. In cases where fewer I/O's are needed, for instance a serial communication line as I^{2}C, it can prove an effective method.

The average SEUPI numbers for the non-mitigated and the FRVC mitigated run in table 5.1 are very similar. Comparing the distribution plots in figure 5.4(a), this similarity becomes even more clear. This is due to the procedure of the fault injection test. For the FRVC runs, a bit flip is first injected in the configuration memory. This is followed by checking the shift register for any subsequent failure caused by the injected bit flip. FRVC is then carried out first after this check. Compared to the non-mitigated approach there is really no difference in the testing procedure expect for the accumulation of bit flips for the non-mitigated approach. Similar distribution curves are therefore expected. If a difference could be seen, this would indicate an effect related to the accumulation of bit flips. For instance, while two or more bit flips in combination could cause a functional failure, flipped individually no failures would be seen. Because of the millions of available configuration bits, if such an effect is existing, characterizing it would be difficult due to the large number of possible combinations. It would demand a very extensive and tedious test which was beyond the scope of this thesis work. The fact that the non-mitigated and FRVC mitigated runs show similar results for this test, can not be considered a universal truth. It very much depends on the operation of the user design. For instance, it could be interesting to check whether FRVC can improve the result in cases were the injected bit flips are corrected before the data in the shift register is shifted. For instance if the shift register clock is slower than the FRVC detection and correction time. In [11] it was predicted that several tens of readout operations can be carried out by the final RCU main FPGA design during



(a)



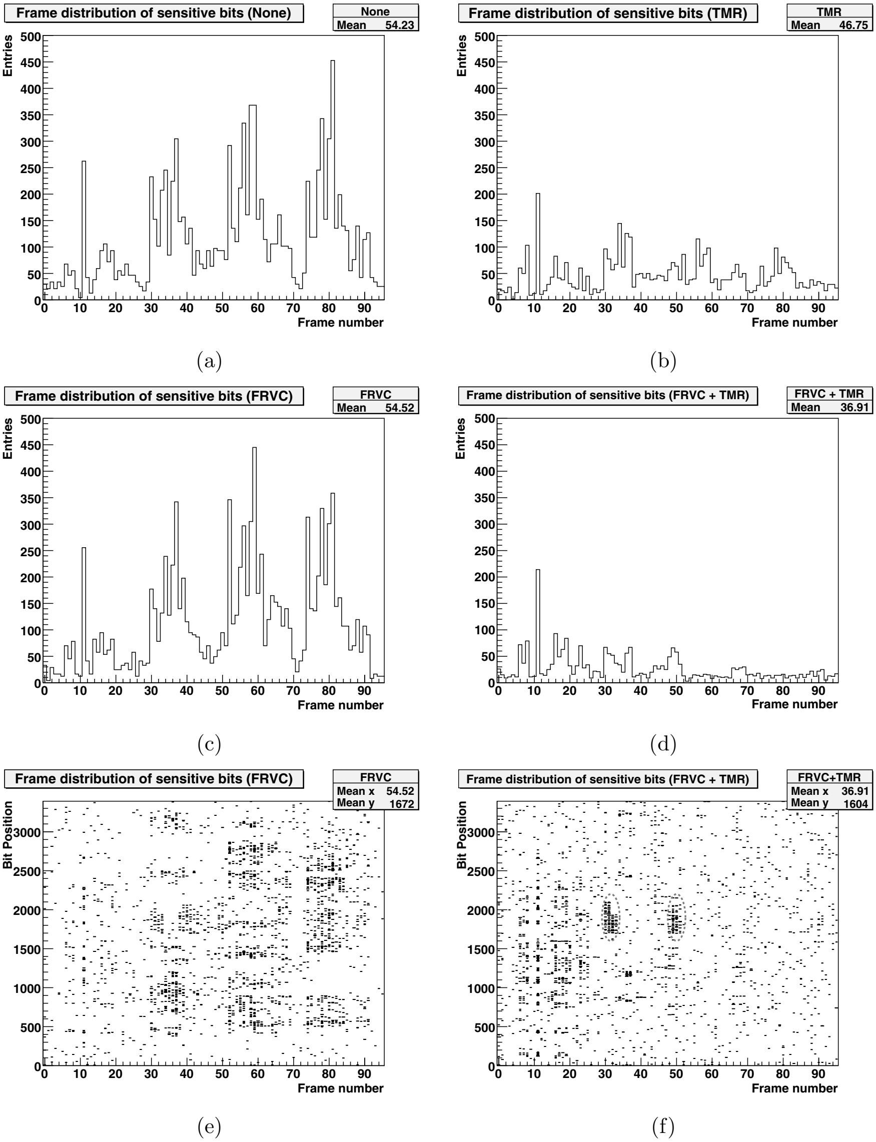
(b)

Figure 5.4: a) shows the distribution in number of fault injections needed to induce a functional failure in the shift register design. Each mitigation scenario were run until 2600 functional failures were detected. The cumulative probability function is shown in b).

this time. It is therefore not expected that FRVC alone will considerably improve the failure rate. As demonstrated in the distribution plot, an improvement is more likely by combining the TMR and FRVC mitigation approaches. However this can only be determined when a fault injection test is carried out on the final RCU main FPGA user design. A similar distribution plot could then be produced. By also generating the cumulative distribution plot, the probability of a failure during an ALICE run can be estimated. As an example the cumulative distribution plot has been produced from the fault injections of the shift register design. In chapter 4 the worst case number of expected SEUs during a standard ALICE run is estimated to 75 for all RCUs. In case the shift register design was meant to operate in this radiation environment, the chance that one of the RCUs would fail during that run is 90% for the non-mitigated version, 50% for the TMR version and 35% for the combined TMR and FRVC version.

Using fault injection to improve the mitigation of a design

Besides using fault injection as a method to predict the failure rate, it can also be utilized to improve mitigation strategies during the development phase. Fault injection can for instance be used to map sensitive parts of a design. In cases were the available resources are limited, and therefore a full TMR implementation is difficult, the strategy could be to only mitigate the most sensitive parts of a design. Even though this was not the main purpose of the randomized fault injection tests of the shift register, the test results can still be used to illustrate this potential. In total 2600 functional failures were induced for the four different mitigation approaches. Mapping the location of the bit flips that induced these functional failures, plots like shown in figures 5.5(a) through 5.5(f) can be produced. For the different mitigation scenarios, these plots show how the location of bit flips causing functional failures are distributed among the 96 frames used for fault injection. To better understand the plots, the frame numbers have to be linked to the particular type of resources used, and if possible to the related logic in the design. This information is given in table 5.3.2 for the 96 frames used in the randomized fault injection test of the shift register design. The first 30 frames contains the GCLK, IOB and IOI frames. Both the TMR shift register and the majority voter is located in the 66 CLB frames numbered 31-96. To limit the widespread of the majority voter it was constrained to a small part of the first 22 CLB frames.The remaining part of the 22 frames are shared with the TMR shift register, while frames 53-96 contains only the TMR shift register. As already demonstrated, enabling TMR alone or in combination with FRVC, increases the average number of fault injections or detected bit flips per functional failure. To reach 2600 functional failures for each of the mitigation



(a) (b) (c) (d) (e) (f)

Figure 5.5: Plots a) through d) shows how the distribution of sensitive bits changes with the applied mitigation method. In e) and f) 2D scatter plots shows additional information on how the sensitive bits are distributed within each frame.

approaches, different numbers of bit flips are needed. The highest number was reached for the combined TMR and FRVC run. In order to better compare the results, the other runs were therefore scaled to the number of detected bit flips reached during the combined TMR and FRVC run. This means that the plots reflect the number functional failures expected if the four different mitigation approaches were exposed to the same fluence of fault injections, or particles in a radiation environment. When TMR mitigation is applied to the shift register, a reduction of functional failures in the corresponding frames is seen. Also, since no mitigation is applied to protect the effect of an SEU in the GCLK, IOB, and IOI frames, the contribution from these frames remains similar through all the four plots. Again, comparing figures 5.5(a) and 5.5(c) it can be seen that FRVC alone does not seem to improve the result. Enabling TMR alone give much better results, but it is when combined with FRVC the best results are observed. Nevertheless, due to reasons already discussed it does not fully remove functional failures in the shift register. Taking a closer look at the distribution in figure 5.5(d), frames 31-52 have a few more entries than frames 53-96. The reason for this difference is not evident from this type of plot. In figures 5.5(e) and 5.5(f) the sensitive bits are plotted in a 2D scatter plot with the frame number along the x-axis and the bit positions within the frame along the y-axis. When only FRVC is applied most of the sensitive bit locations are distributed within frame 30-96 which contains the shift register and majority voter. Applying both TMR and FRVC it is expected that the number of entries in these frames should be reduced compared to the first non-mitigated GCLK, IOI and IOB frames. This is also observed in figure 5.5(f) except for the two areas highlighted by the dashed circles. There seems to be two distinct areas within these frames that now stands out as more sensitive. Comparing the relative position within the frame along the y-axis to the location of the majority voter, it can be concluded that this concentration of sensitive bits most likely is connected to the majority voter.

Frame numbers	Resource type	Design usage
1-4	GCLK	-
5-8	IOB	-
9-30	IOI	-
31-96 31-52	CLB CLB	shift register Majority voter

Table 5.5: Correlation between frame number and the resource type and usage in the design. The majority voter is sharing the frames 30-51 with the shift register.

The main purpose of showing these plots was to illustrate how fault injection can be used to locate the sensitive parts of a design. When developing user designs for FPGAs that will operate in radiation exposed environments, this type of analysis can be used to improve the mitigation strategies of a design.

5.4 Discussion and Summary

A consequence of software controlled fault injection is the reduced injection speed compared to an implementation directly in the user design of the RCU support FPGA. This limitation is largely caused by the 60 ms overhead which is believed to be connected to accessing drivers and switching of bus modes by the DCS embedded computer. For continued development, effort could be put into optimizing the drivers and software in order to try to reduce this overhead. During the validation test, fault injection could not be carried out simultaneous to an FRVC cycle. The reason is that only one physical interface, the SelectMAP interface, is available to access the configuration memory space. This will slightly influence the distribution in the time it takes to detect a bit flip, and potentially a minor change in the life time of any subsequent functional failure in the user design.

Using the DCS bus as a shared interface for all operations limits the flexibility of the fault injection for the test design. For example, the communication of stimuli to and response from the shift register test design has to be paused during fault injection. A reduction in visibility of any time sensitive failure signatures caused by injected bit flips may be seen. The impact of this limitation is considerably reduced when fault injection will be used for the final RCU main FPGA firmware. The reason is because the DCS bus is not part of the main data path of the readout system. A dedicated data path is used for communicating data from the front end cards to the data acquisition system (DAQ). This data path is indicated by the black arrows in figure 3.5. Fault injection can then be carried out in parallel to normal operation of the RCU main FPGA. This will be used to test how configuration memory upsets may impact the readout of detector data.

The main purpose of adding configurable TMR was to show the potential of fault injection as a method to test mitigation techniques in the final firmware of the RCU main FPGA. It was therefore kept simple and the factor 4 improvement compared to no mitigation is only an indication of its effect. Effort was not put into increasing this factor by improving the TMR implementation by for instance also replicating clock domains. The quantitative results from fault injection tests will vary with the complexity of the design and implemented mitigation technique.

Multiple bit upsets have not been addressed in this chapter. Using fault injection, there is no limitation on the number of bits that can be flipped at the same time as long as they are located within the same frame. If multiple bit flips should take place between different frames, a short delay between the bit flips will be present. This is because the frame data input register in the Xilinx configuration state machine only can latch one frame of data into the configuration memory at a time. In [16] multiple bit upsets have been investigated for different versions of the Xilinx Virtex family. Only $1 - 3\%$ of the upsets caused by a 63.3 MeV proton beam are reported to be multiple bit upsets. Multiple bit upsets are therefore considered to be a higher order effect in the hadron radiation environment of the TPC detector.

Chapter 6

Monte Carlo based SEU simulations

Physics based simulations to investigate the SEU response of an SRAM device have been demonstrated in a number of cases [59][60][61][62]. These simulations utilize Monte Carlo transport codes to describe the different physical interactions between an incident particle and the device material. The Fluka Monte Carlo simulation package [25][26] and the Geant4 toolkit [63][64] are two examples of such particle transport codes. Another example is the IBM proprietary simulation platform called SEMM-2 [65]. While Fluka and Geant4 are general purpose tools that can be tuned by the user to meet a number of different applications, SEMM-2 is specifically developed for single event effect analysis of advanced CMOS technologies. These codes can be used to study radiation induced energy depositions from particles transported through complex geometrical structures.

As the critical charge decreases with newer technology and decreasing feature sizes, any small variation in particle energy or material structure may become essential. The importance of the overlaying metal interconnect layers have for instance been demonstrated in [66], where simulations showed increased energy deposition within the sensitive regions when the interconnect layers were included. A similar discussion is also presented in [67]. Both these studies show that Monte Carlo simulation can be used to study how different parameters such as particle type and energy, material composition and structural layout can contribute to the SEU rate of an SRAM device.

This chapter is motivated by applying Monte Carlo simulation as a method to study the SEU sensitivity of a device. It will describe the work which has been carried out in order to prepare a Monte Carlo simulation case study for the RCU main FPGA.

6.1 General methodology

Monte Carlo simulation of SEUs incorporates a number of important aspects ranging from the description of the target geometry, through particle transport, and finally post-processing of the results. A generalized overview of this flow is shown in figure 6.1. The first phase of the simulation flow is the preparation of the simulation request. This involves the description of the target geometry and the radiation source that will be used. The radiation source can for instance be a mono-energetic beam of a given particle type, or a more complex source representing a distribution of particles and energies like in the TPC radiation environment. When a source particle travels through the target geometry it will undergo different physical interactions with the material. An accurate description of the target geometry and material composition is therefore important in order to achieve the most correct results. Depending on the entry point of the source particle, it may pass different compositions of material on its path through the target geometry. This will lead to a dispersion in stopping power for the particles reaching the sensitive region. Consequently particles with equal intial energy may deposit different amounts of energy in a sensitive volume. In one case this may lead to an SEU and in another not.

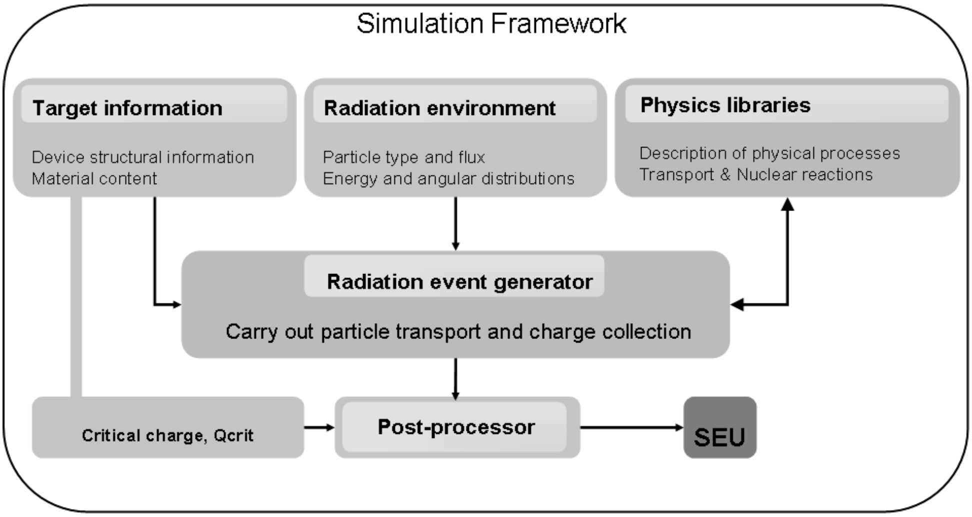

Simulation Framework
Target information
Device structural information
Material content
Radiation environment
Particle type and flux
Energy and angular distributions
Physics libraries
Description of physical processes
Transport & Nuclear reactions
Radiation event generator
Carry out particle transport and charge collection
Critical charge, Qcrit
Post-processor
SEU

Figure 6.1: Generalized flow of the SEU simulation methodology. The diagram is based on a similar flow chart found in [65].

The second phase is the actual simulation phase where source particles are transported through the target material. A number of libraries and databases are utilized to describe the different physical interactions between the source particle and the target material. This can for instance be libraries containing non-elastic and elastic interaction cross sections for a given particle type and energy in a given material. Monte Carlo sampling techniques are then applied to determine whether a non-elastic interaction will take place along the path of a source particle or not. In a similar way, possible reaction fragments can be collected from a reaction database for further transport.

The basic concept of SEU simulation is to define a sensitive region within the geometry and then detect the amount of energy deposited in this region. The sensitive region is often described as a rectangular parallelepiped (RPP) volume. Whether an SEU has occurred or not is decided in the post-processing phase by comparing the deposited energy to a predefined critical energy. As pointed out in [62], the RPP and critical energy approach is rather simplified as it neglects for instance charge collection efficiencies and characteristics of the generated current pulse and the circuit response. Nevertheless, to a first order it still captures the main effects. Others have suggested improvements to this method. In [68] a single sensitive region is made up of a number of sub-volumes. Each sub-volume is then assigned a collection efficiency factor depending on the sub-volumes' distance to the circuit's sensitive node. A similar type of approach is also reported in [69]. This allows to have more complex sensitive regions and to avoid a sharp cut-off at the edges of a single RPP volume.

6.1.1 SEU cross section from Monte Carlo simulations

As shown in section 4.0.1 the SEU cross section can be expressed as

$$\sigma_{SEU,bit}(E_P) = \frac{N_{SEU}}{\Phi\, N_B}, \tag{6.1}$$

where N_{SEU} is the number of detected SEUs in N_B number of bits or SRAM cells, and Φ is the fluence of incident particles with energy E_p. In this case the fluence is restricted to the the surface area A_{sim} of the target geometry so that

$$\Phi = \frac{I_0}{A_{sim}}, \tag{6.2}$$

where I_0 is the number of incident particles. An SEU is defined as an event where an energy, E_{dep}, larger than the critical energy, E_{crit}, has been deposited within a collection volume. The Collection Volume (CV) represents the sensitive part of an

SRAM cell where any charge or energy deposited will be collected by the nearby node.

During a simulation the energy deposited by each transported particle must be collected individually. The final result is a distribution of events that can be binned in a histogram according to the amount of energy deposited. Given a critical charge E_{crit}, the number of SEUs can then be computed by integrating over all bins $N(E_i > E_{crit})$, where $N(E_i)$ is the number of events having energy depositions within the range $E_i \leq E_{dep} < E_{i+1}$, and i is the bin number.

$$N_{SEU} = \int_{E_{crit}}^{\infty} N(E_i)\, dE \tag{6.3}$$

When normalized to the fluence Φ and number of bits N_B, equation 6.4 describes the probability that an incident particle will result in an SEU. This is the SEU cross section as a function of incident particle energy E_P,

$$\sigma_{SEU}(E_P) = \frac{\int_{E_{crit}}^{\infty} N(E_i)\, dE}{\Phi\, N_B}. \tag{6.4}$$

Another often used approach is to instead normalize equation 6.3 by the total number of energy deposition events as shown in equation 6.5,

$$F_C(E^{'}) = P(E_{dep} > E^{'}) = \frac{\int_{E^{'}}^{\infty} N(E_i)\, dE}{\int_{0}^{\infty} N(E_i)\, dE}, \tag{6.5}$$

where $P(E_{dep} > E^{'})$ describes the probability that an energy larger than a given energy $E^{'}$ will be deposited by a particle hitting the collection volume. This is known as the the complementary cumulative distribution function.

6.2 Resolving case study geometry

In order to carry out a meaningful Monte Carlo simulation the description of the target geometry has to be resolved. A picture of the Xilinx Virtex II Pro FPGA along with a schematic cross section is shown in figure 6.2. The device die is mounted in a 672 pin flip-chip BGA [1] package. In a flip-chip BGA the die is flipped over and placed face down. It is connected with solder bumps to the package substrate. This means that for the accelerated beam tests described in section 4, where the proton beam entered perpendicular to the copper lid, the beam first has to travel through the copper lid and full silicon substrate before reaching the layer of sensitive

[1] BGA: Ball Grid Array

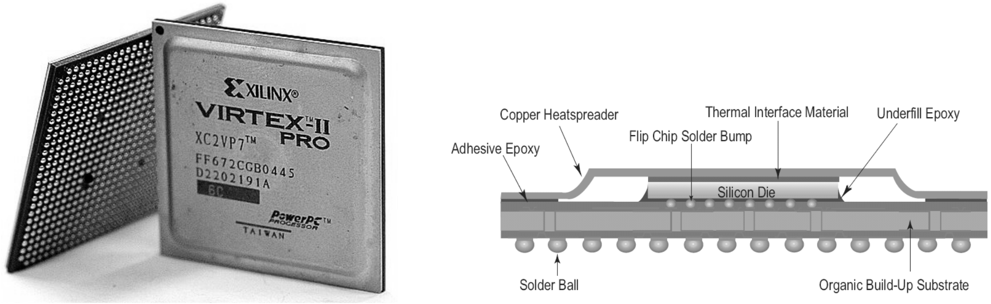


Figure 6.2: Picture of the Xilinx Virtex II-Pro FPGA to the left and a schematic cross section to the right [70].

transistors. This topology is shown in figure 2.4. Only after passing the sensitive region the beam will traverse the interconnect layers and into the package substrate and potentially a flip chip solder bump.

As the Xilinx Virtex-II Pro FPGA is a commercial device, only limited information about its internal structure is available. Except for that it is manufactured in a 0.13 μm and nine-layer copper process, the data sheet [37] reveals no other essential details such as the thicknes of the silicon substrate or the interconnect layers. Some additional information is found in the material data sheet [71] were the flip-chip solder bumps are reported to be an alloy of 63 % tin and 27 % lead. It can be important to know the material composition and the thickness of these different layers in order to have an accurate modeling of the beam particles through the target geometry. In addition the size of the collection volume representing the sensitive regions of an SRAM cell must also be defined.

6.2.1 Structural analysis

A structural analysis was carried out to determine some of the most important parameters needed for the simulation. The analysis was carried out using a Focused Ion Beam (FIB) instrument and the procedure is illustrated in figure 6.3. To prepare for the FIB analysis the device had to be cut open to expose the die. Most of the silicon substrate was polished away to allow the FIB to dig a trench in the interconnect layers with a high intensity gallium beam. The device was then tilted 45 degrees and images were produced using a low intensity beam. Based on the FIB analysis the thickness of the silicon substrate and the interconnect layers were determined. The thickness of the copper lid in addition to the length and width of the silicon die was simply measured using a digital ruler. Two examples of FIB

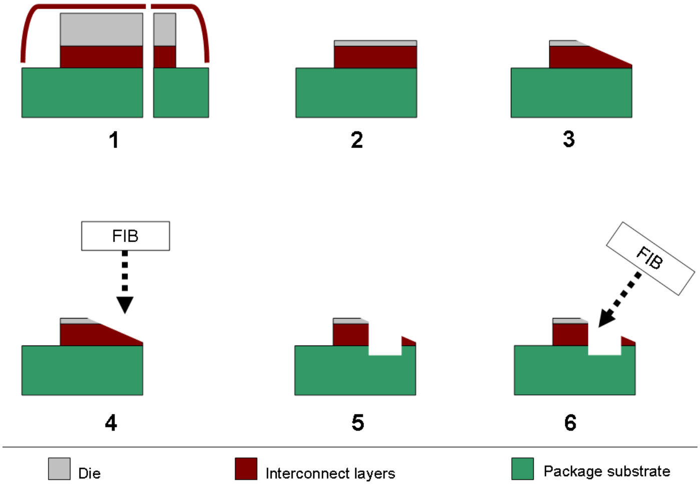


Figure 6.3: Schematic showing the the focused ion beam analyse procedure. The device is first cut in two parts before most of the silicon substrate is polished away(1 and 2). To expose the interconnect layers the last part of the polishing process is carried out in a shallow angle (3). The FIB is then used to dig a trench into the inteconnect layers before the device is titlted with respect to the beam in order to produce an image (4-6).

images are shown in figure 6.4 where the thickness of each individual layer is simply determined by visual measurement. Each layer is further assigned a metal fraction factor that gives the ratio of metal interconnect to dielectric for each layer. The FIB images in figure 6.4 represents only a tiny fraction of the full interconnect layers. Basing the metal fraction factor solely on these two images will obviously limit the accuracy. It can however be used as a nominal starting value for a variability study were the metal fraction factor will be varied in steps.

Collection volume parameters

The FIB analysis does not reveal any information about the actual dimensions of the SRAM cell or its sensitive nodes. Estimates of nominal values are therefore derived based on information from other sources. Similar to the metal fraction factor for the interconnect layers, the dimensions of the sensitive volume can then be varied to investigate how this will impact the result.

Values for the area of a standard SRAM cell is available for different UMC pro-

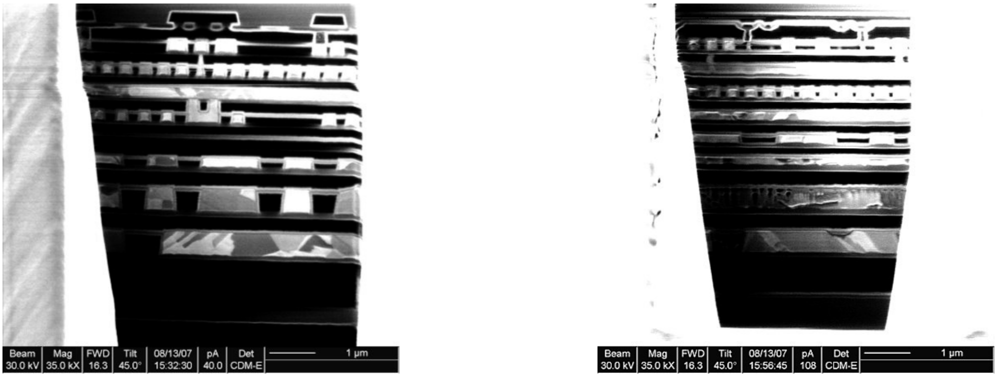


Figure 6.4: FIB images showing the structure of the interconnect layers at two different locations.

cess technologies [72], [73] and [74]. The SRAM cell area for their 0.15 micron, 0.13 micron and 90 nanometer process technology is 3.15 μm^2, 2.28 μm^2, 1.16 μm^2 respectively. However, in [17] it is reported that Xilinx has always designed their static latch configuration cells to different criteria than those used to design commercial SRAM cells. These cells are designed for robustness and not speed. Maximizing the capacitive load is reported to be one design criterion. This can for instance be done by increasing the width of the transistor. Thus, the UMC numbers should only be treated as guidelines. Moreover, as pointed out in section 2.2.2 the most sensitive region in an SRAM cell is the depleted drain region of an "off" state transistor. The area of the collection volume is therefore smaller than the full area of the SRAM cell. Also, an SRAM cell in normal operation has both a PMOS and an NMOS in an "off" state. Each SRAM cell therefore has two sensitive regions. Due to the higher mobility of electrons compared to holes, the charge collection efficency is considered to be higher for the NMOS transistor. This makes it more vulnerable to SEUs than the PMOS transistor [75]. Consequently the SEU cross section will be dominated by SEUs in the "off" state NMOS transistor. For simplicity only one collection volume is therefore assumed for the simulations carried out in this thesis work.

Basic scalable CMOS layout rules can be applied as a reference to estimate a nominal size for the sensitive area. Considering this area to include the drain and channel region, a first suggestion for the minimum dimensions is approximately 0.4 μm by 0.4 μm. These dimensions can be compared the values used in [76] where a collection volume area of 0.4 by 0.4 μm^2 was used for a 0.13 μm SRAM process. It is however assumed that Xilinx does not follow the aggressive scaling of configuration cell dimensions since stability and not speed is prioritized. As previously mentioned, increasing the width of the transistor could be a possible method to increase the critical charge and thereby by make the transistor more robust. For the simulations presented in this work, a sensitive area of 0.4 by 0.8 μm^2 will therefore be used as a inital case and increased in steps in order to see how this impacts the SEU cross sections.

The study in [76] also gives an estimate of 1μm for the worst case depth of the collection volume. For comparison a different study [69] used a charge collection depth of 0.48 μm for a 0.25 μm process. The depth of the sensitive region is largely determined by the depth of the depleted drain region, which again is dependent on biasing conditions and doping profiles. An accurate determination of the depth is therefore difficult when this information is unavailable. The depth of the sensitive volume will therefore also be varied similar to the case of the sensitive area.

For the Xilinx Virtex-II Pro the circuit simulations suggest an average value for the critical charge of 12 fC [24]. It should be kept in mind that this is an average value. In reality the critical charge is a dynamic value which can show a wide distribution without a sharp cut-off. For example, in [17] Xilinx reports that the critcal charge varies by as much as a factor 12:1 due to the loading on the memory cells.

6.2.2 Generic geometry input description

A generic geometry description is made based on the information retrieved from the structural analysis and collection volume related discussions. This description will be used as the basic input to the simulation. A schematic of the geometry is shown in figure 6.5 where the main dimensions are labeled according to the information in appendix E. The reference point R_C is used to locate the geometry in the coordinate system of the simulation tool. It is defined as the origin (X=0,Y=0,Z=0) and its xy-plane represents the interface between the surface of the Sensitive Region (SR) and the interconnect layers (icl). Depending on the simulation platform used, small adjustments can be made. In particular the details of the interconnect layers may be differently described due to special implementation solutions.

The collection volume, representing the sensitive "off" state NMOS transistor, is enclosed in the sensitive region as shown in figure 6.6. Its area is given by the dimensions *xlcv* and *ylcv* while the depth is described by *zlcv*.

Even though local density variations may exist, it is assumed that the SRAM cells are evenly distributed over the area of the silicon die. This assumption is valid as long as only single event upsets are studied. Each SRAM cell can then be treated as an isolated case. In case of multiple bit upsets the inter-cell distance becomes

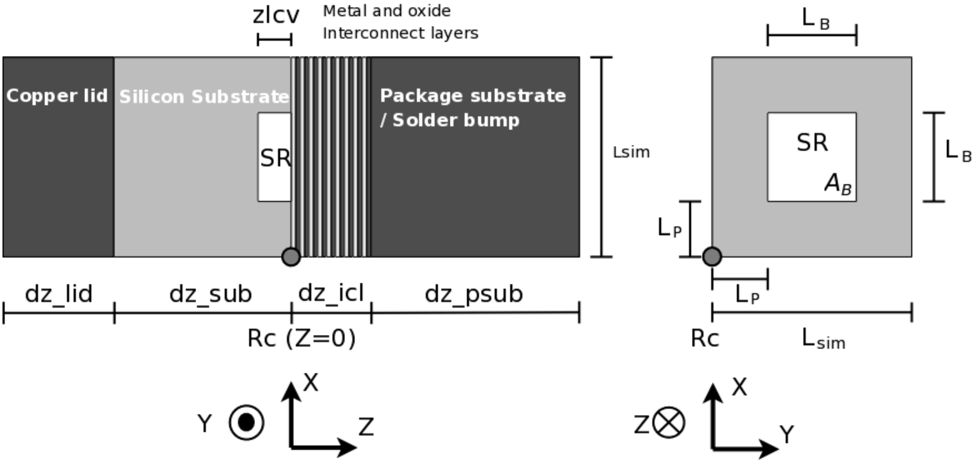


Figure 6.5: Schematic showing the generic geometry description for the simulation. See figure 6.6 for a closer description of the sensitive region. Schematic is not to scale.

important and the previous assumption is no longer valid.

6.3 Preparation and setup of simulation tools

Preparations are made to run a case study simulation using the Fluka simulation package [25], [26] in addition to an adapted version of the IBM soft-error Monte Carlo model SEMM-2 [65]. Fluka is a general purpose tool for calculations of particle transport and interactions with matter. It spans a number of applications like proton and electron accelerator shielding applications, target design, calorimetry, activation, dosimetry, detector design, Accelerator Driven Systems, cosmic rays, neutrino physics and radiotherapy. SEMM-2 is the most recent version of the IBM soft-error simulation project and is optimized for analysis of single event effects in advanced CMOS technologies. For more in depth information about the tools the reader is referred to [25], [26], [77], and [65], including references therein.

6.3.1 Fluka specifics

The setup of a Fluka simulation is based on the standard Fluka input options. In the following section a brief overview of the setup for this case study is given. A more comprehensive description on how to use the standard Fluka input options

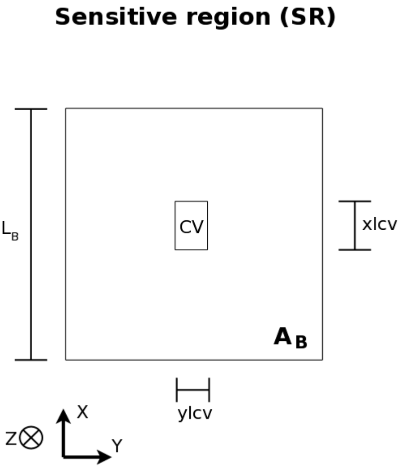


Figure 6.6: A schematic showing the basic structure of the a sensitive region containing one SRAM cell (one bit, $N_B = 1$). Each SRAM cell has two collection volumes representing the depleted drain region of the two “off” state transistors. Schematic is not to scale.

can be found in the Fluka online manual [77].

The input file and general settings

A simulation request in Fluka is based on the input of a text file. The file contains a number of individual input cards describing different features of the simulation. To run a basic simulation, a minimum amount of information is needed related to the

- radiation source
- geometry
- scoring

Additional input cards can be used to optimize settings like physics, transport of particles, and biasing. Fluka has a special input card called *DEFAULT* which can be used to collectively optimize a number of settings to a certain problem. Thus with a minimum effort it is possible to carry out a simulation with appropriate conditions. Nevertheless, some additional features have to be enabled for this case study. In detector environments the scale of transport is in the metre range. Thus, the short range of heavy fragment ions can be considered to be point deposition of energy. By default, production (evaporation) of heavy fragments from non-elastic interactions is not activated. Likewise the transport of heavy fragment ions is disabled or approximated. The result is a reduction of computation time. Approximation means that the particle is not transported but ranged out to rest in an approximate way. Its kinetic energy is deposited uniformly over the residual range within a region. If the range crosses a boundary to the next region a new residual range is calculated in the new region [77]. To gain a more accurate result for residual production and ion transport these features should be enabled in Fluka. This is done using the special input cards *EVENTYPE* and *PHYSICS*. The energy cut-off for transport of particles should also be lowered to an appropriate level. The simplest method is to use the *PRECISIOn* setting of the *DEFAULT* input card. This collectively sets the transport cut-off to 100 keV for all particles except neutrons which goes down to thermal energies. A 100 keV α-particle has a range of 0.7 μm and can maximally deposit 100 keV or 4.5 fC within that range. Similar for a 100 keV silicon ion the range is approximately 0.1 μm. Thus the default transport threshold of 100 keV is sufficient for application with critical charge above 10 fC.

Geometry description

The geometry input is based on the generic description from section 6.2.2. Figure 6.7 shows a cross section of the geometry in the in the zx-plane. It consists of a 450 μm copper layer, 850 μm silicon layer, 10 μm of interconnect layers and finally a 100 μm layer of either package substrate or solder bump material. The package substrate layer is shorter than for the real device. This is done deliberately to reduce the distance the primary source particles have to be transported. A length of 100 μm should be more than sufficient to include the effect of possible fragments produced in non-elastic collisions. In figure 6.8 the region of interconnects has been zoomed in. As previously mentioned in section 6.2.1, an exact description of these layers is not available. Each layer has therefore been assigned a fraction of metal based on the metal fraction factor determined from the FIB images of the structural analysis.

Fluka uses combinatorial geometry description to build the geometry. For structures containing many regions, this can result in relatively complex descriptions. The resolution of the interconnect layers is kept in multiples of the thickness of the actual layers. With 18 individual layers (oxide + metal) and a sub-micron resolution, a large number of regions is needed to describe the interconnect layers. This has been avoided by using the voxel based geometry description which is available in Fluka. In the Fluka online manual [77] voxels are referred to as tiny parallelepipeds forming a 3-dimensional grid. Each voxel is equally sized and is assigned a material. Two materials are needed to describe the interconnect layers, copper to represent the metal wires, and SiO_2 to represent the dielectric(oxide). A numerical 3-dimensional matrix is constructed where each voxel is assigned a material based on an integer number. The result is a geometrical structure described without the use of combinatorial geometry. This approach is especially useful when the full layout information of the interconnect layers is unknown. The metal can easily be distributed among the voxels in a random fashion based on the metal fraction factor for each layer. Due to the repetitive pattern of logic blocks and wire interconnects in an FPGA, it is assumed that the fraction of metal for each interconnect layer is relatively evenly distributed. Thus, using a random distribution of material is expected to give a reasonable representation of the interconnect layout.

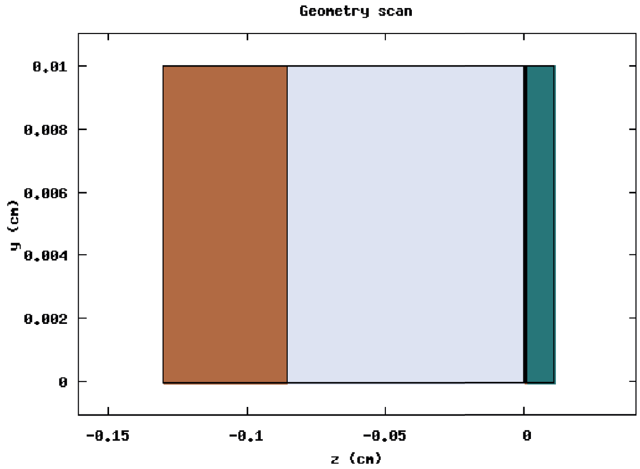


Figure 6.7: Target geometry as described for the Fluka simulation. Full structure with the Copper lid (brown), silicon substrate (grey), thin layer of interconnect (not visible) and the solder bump region(blue).

Scoring and collection volume

To score event by event energy deposition in Fluka the *EVENTBIN* scoring card can be used. Similar to the closely related *USRBIN* card it scores the requested quantity in a regular spatial structure (binning) independent from the geometry [77].

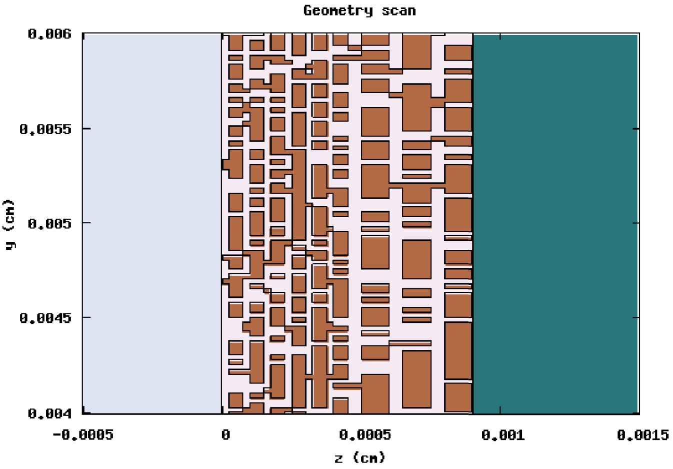


Figure 6.8: Close up of the layers of interconnects with randomly distributed Copper according to the predefined metal fraction factor for each layer. The sensitive region is located in a thin layer at the very end of the silicon substrate close to the layer of metal interconnects.

However, in contrast to the *EVENTBIN* card, the *USRBIN* card scores the total accumulated value for a full simulation and can therefore not be used for event by event scoring. An event corresponds to the transport and interactions of one primary beam particle. This includes, if produced, transport and interactions of any secondary fragments related to the primary particle. Thus, the number of events in a simulation is identical to the number of primary beam particles.

A great advantage of the *EVENTBIN* scoring is its independence of the geometry. This means that for energy deposition scoring there is no need to implement the geometrical description of collection volumes. The bin volume dimension can simply be sized in the same resolution as the collection volume size. In this fashion the energy deposited in one single bin represents the energy deposited in one collection volume. As explained in appendix E.2, the simulation time can be optimized by increasing the number of collection volumes. This can be done by increasing the number of bins in the *EVENTBIN* scoring structure. Further, if the binning structure is made smaller than the size of a collection volume, a configuration of several bins can make up the total collection volume. In fact, by running one simulation and combining the bins in different configurations, this allows to do post simulation variability studies of the collection volume dimensions. In addition it enables the possibility to apply charge collection efficiency factors to different parts of a collection volume as suggested in [68] and [69]. If the collection volumes were implemented as geometrical structures, a new simulation would have to be carried out if the dimensions of the collection volume were to be changed.

The *EVENTBIN* card can be combined with the *AUXSCORE* card in order to filter particles of interest. This can for example be used to study the contribution in deposited energy from individual particle types.

Limitations of the ***EVENTBIN*** scoring scheme

The use of *EVENTBIN* comes with some limitations. During simulation an entry is written to the log file for every event. This takes place even though no hits were detected in any of the scoring bins. This is a significant disadvantage. Due to the low reaction cross section, a large number of events is needed to create a statistically significant number of energy depositions which is larger than the critical energy. As a result the log files can reach sizes in the order of gigabytes.

Another drawback of the *EVENTBIN* scoring is the increased CPU time. A few simulations were carried out to measure the average CPU time used to follow a primary particle. After a simulation this information is available in the standard Fluka output file. Activating the *EVENTBIN* scoring for a grid of 235 x 235 x 10 bins increased the simulation time by a factor 15 compared to not using EVENTBIN scoring. A simulation was also run with half the number of bins to show that the CPU time clearly is dependent on the number of bins scored. These runs are based on the geometry as it is described in section 6.3.1 and using settings as described previously in this section. When *EVENTBIN* scoring is activated, an average CPU time of 12 ms per primary particle gives a total simulation time of approximately 14 days for 10^8 primary particles. Running without the *EVENTBIN* scoring this could be reduced to less than 1 day. The CPU time is of course also dependent on the specifications of the computer running the job. Nevertheless, the result shows that the relative difference can be significant. This result also contradicts the previous statement that increasing the number of collection volumes by increasing the number of bins would decrease the simulation time. In reality this is a trade-off situation were a decrease in simulation time will be achieved as long as the number of collection volumes is kept relatively low.

Two step simulation method

Some effort can be put into increasing the effectivity when using *EVENTBIN* scoring. A two step simulation method is therefore proposed. The first step involves a simulation to determine all non-elastic interaction points. This step does not use *EVENTBIN* scoring and is further optimized by turning off irrelevant physics and particle transport settings. The result is reduced simulation time. A special user modified scoring routine (mgdraw.f [77]) is used to detect the interaction points. This scoring routine produces a log file containing the coordinates of the interaction point and information about the fragments produced in each interaction. In the second step another special user routine (source.f [77]) is used to only load the fragments of interest for further transport. *EVENTBIN* scoring is now activated to score the deposited energy. Because the number of non-elastic interactions is much lower than the number of primaries needed to create them, the number of events to be transported is equally lower. This significantly reduces the size of the log file. Also, the second step is faster due to the lower number of particles to be transported.

6.3.2 Modifications of the SEMM2 model

Some of the simulation methodologies of SEMM-2 have been adopted to generate a separate SEU simulation module specific to the needs of this case study. The module is labeled SEMM2.vBergen. A main reason for developing this module is that SEMM-2 invokes a lot more details than can be treated in the practical situation where design information such as the geometry description is minimal.

Similar to Fluka, a SEMM2.vBergen simulation is requested based on the input of a text file containing information on the simulation setup. The three main blocks of data that needs to be specified are

- Geometry description
- collection volume description
- radiation source

where in particular the description of the geometry differs from the standard version of SEMM-2. As SEMM-2 is already optimized for simulation of SEUs, no additional tuning of parameters connected to physics or transport of particles are needed.

Geometry description

In the standard version of SEMM-2 the geometry description is extracted directly from the device layout produced by a custom IC design tool. An algorithm patented by IBM [78][79] can automatically convert this layout information into a large number of 3-dimensional and rectangular pixels. Each pixel is characterized by its location and the material it contains. In its simplest form the material is dielectric with a certain fraction of metal. This method makes it possible to give a complete and detailed description of the interconnection layers. Because the circuit layout information is not available for this case study, the part of this method related to extraction of layout information can not be applied. Instead the geometry description based on the generic geometry presented in section 6.2.2 is used. It is then utilized as a special case of the standard SEMM-2 where each layer is represented by a single pixel. This allows to take advantage of the metal fraction factor that can be assigned to each pixel. The content of each layer can then be characterized by two material types and the fraction of the first material to the second material. In SEMM-2 each pixel is also associated with what is called a granularity sampling length. This defines the distance a particle is transported before a new sampling is carried out to determine the material content of the pixel. A special case is when a layer contains only one type of material like for the silicon substrate. The metal fraction factor is then set to 1 and the granularity sampling length is of no importance. However to save computational time it can be set to a value larger than the longest path through the layer.

When applied to the interconnect layers the metal fraction factor and granularity sampling length becomes powerful. Combined the result is that a particle being transported through this region effectively sees a number of randomly distributed pixels of either metal or dielectric. This can be compared to the voxel approach used to describe the interconnect layers for Fluka in section 6.3.1. An important difference is however that for the voxel based approach, the pixel distribution is defined before a simulation is carried out. The advantage of SEMM-2 is that the granularity sampling length approach makes this distribution dynamic during the simulation. Particles entering the same region at a similar location having an identical direction will therefore not see exactly the same material[2]. In case these particles crosses a collection volume on their path, the result is a distribution in the energy deposited within this volume. Consequently, one particle may cause an SEU and another not. The complex feature of the interconnect layers can therefore still be captured even though only one collection volume is implemented for the simulation. To

[2] Average over a large number of particles the material budget will however reflect the assigned metal fraction factor.

achieve the same effect for the fixed voxel based approach, a number of collection volumes must be used and spread over a larger area. As explained in section 6.3.1 implementing several collection volumes for the Fluka simulations is actually an advantage as it increases the detection efficiency, but only to a certain point. This type of optimization is not necessary in SEMM-2 due to how the radiation source is described.

Radiation source

SEMM-2 has three main options of particle sources available for simulation.

- α-particle source
- Mono-energetic hadron beam (proton, neutron, pion, heavy ion)
- Cosmic ray sources

These options cover the main areas of interest such as α-particles from contamination of the package material or leaded solder bumps, hadron beams for simulation and comparison to experimental results, and cosmic ray sources to investigate the impact of terrestrial cosmic ray neutrons. The cosmic ray source can in principle be a mix of particles with different energy distributions. It can therefore be used to describe complex radiation fields like for instance the TPC radiation environment. For the preliminary version of SEMM2.vBergen only the mono-energetic hadron beam option is implemented. A great advantage of this option is that the user can specify the number of non-elastic interactions instead of the number of source particles to transport. That is, all the source particles simulated will experience a non-elastic reaction. This saves a significant amount of CPU time, and combined with the granularity sampling length it greatly reduces need for other optimization techniques.

6.3.3 Status

At present it has been demonstrated that SEMM2.vBergen successfully can interpret the specialized input file. The first preliminary simulations runs also show promising results when compared to the corresponding Fluka results. However, some additonal work and testing is needed before a full production run can be carried out. Finally it was therefore not possible to present any results within the time frame given by this thesis period.

6.4 Fluka simulation results

The primary objective of the preliminary simulation runs was to validate the setup and methodology by comparing the simulation results to the mone-energetic experimental results discussed in chapter 4. At the same time this comparison was also used to try and determine a possible size of the sensitive volume through a variability study.

Due to the uncertainties connected to the geometry description of the metal interconnect layers, a number of simulations were carried out varying these input. The simulations were carried out according to the two step method discussed in section 6.3.1. Only particles of charge equal to or higher than 2 were transported during the second step. In addition to calculating the total cross section, *EVENTBIN* scoring cards were implemented to study the contribution from α-particles and the group of particles with charge higher than 2. The latter group is referred to as heavy fragments (HF) in the following.

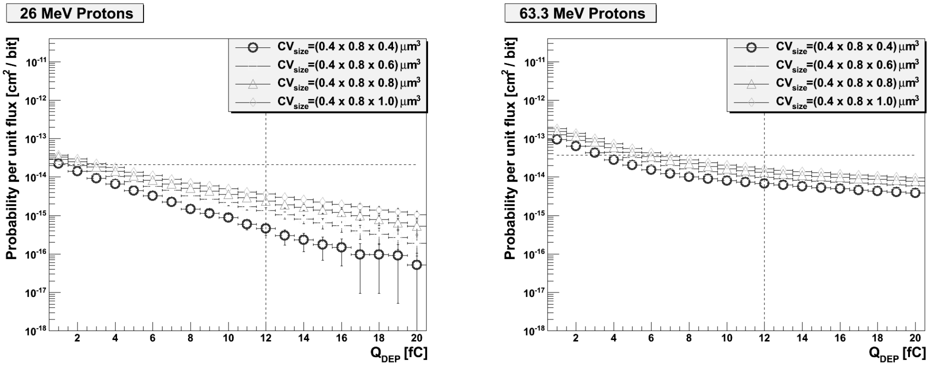
6.4.1 Collection Volume variability study

By far the greatest uncertainty in the simulations is related to the dimensions of the collection volume. A variability study was therefore carried out to investigate if the dimension of the collection volume could be estimated based on comparison with the experimental results. The different dimensions used are listed in table 6.1. For each configuration of the area the depth was varied from 0.4 μm to 1.0 μm in steps of 0.2 μm. Two simulations were carried out using proton beam energies of 26 MeV and 63.3 MeV entering the target geometry from the copper lid side (from the left side in figure 6.8). The metal interconnect layers were described using the voxel based approach as shown in figure 6.8. To compare the results to the experimental SEU cross section, the results in figures 6.9 through 6.12 are plotted as the probability of charge deposition in a sensitive region per primary beam particle. This is also

length [μm]	width [μm]	depths [μm]
0.4	0.8	0.4, 0.6, 0.8, 1.0
0.4	1.2	0.4, 0.6, 0.8, 1.0
0.6	1.2	0.4, 0.6, 0.8, 1.0
0.6	1.8	0.4, 0.6, 0.8, 1.0

Table 6.1: Collection volume dimensions applied to the post-simulation variability analysis. For each area configuration four different depths of the collection volume were studied.

a conveniant way of presenting the data since the critical charge is dynamic value. The charge deposition axis is limited to the region of interest being around the expected average critical charge of the Xilinx Virtex-II Pro FPGA. A reference line corresponding to the experimental SEU cross section is added in each respective case of the energy. Similarly a line indicating the average critical charge is also added. If these lines and the simulated probability curve all intersect at the same point, this may be used to suggest a possible size of the collection volume. A first observation is that the the calculated cross section increases with decreasing critical charge. This is expected as a lower critical charge means that less charge is needed to induce an SEU. Similarly, if the size of the collection volume increases, the cross section also increases. This is due to the increased path a particle can travel through the volume which again leads to more charge being deposited.

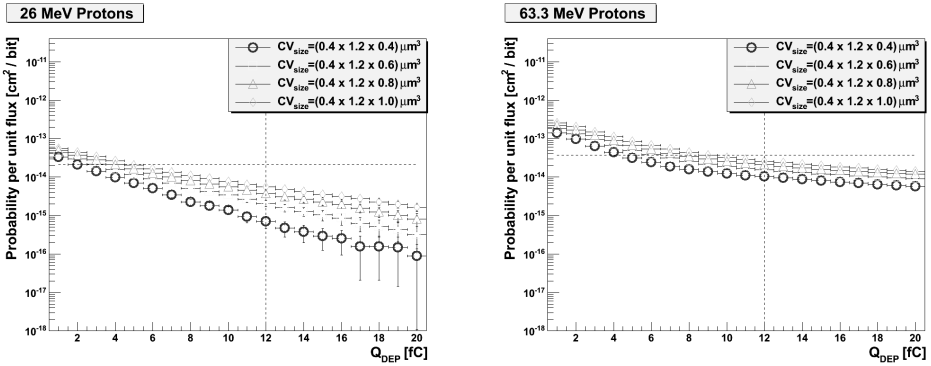


(a)

(b)

Figure 6.9: Results for the depth variability study for a collection volume area of (0.4 x 0.4 μm^2) and primay beam energies of 26 MeV (a) and 63.3 MeV (b).

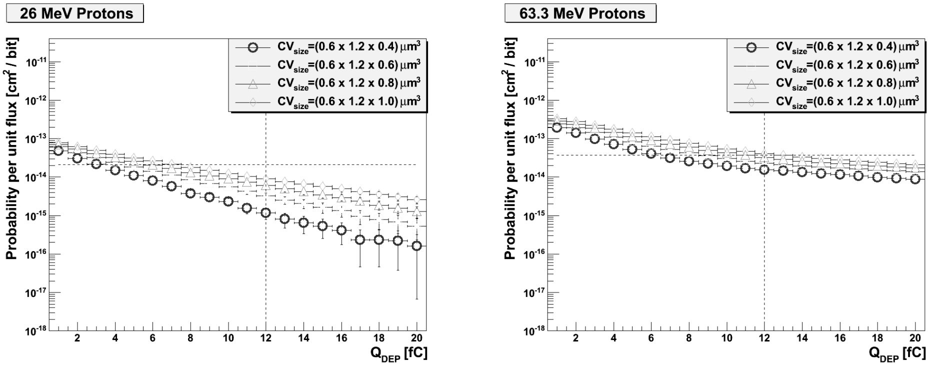
Comparing the various plots, it is not possible to find a perfect match where both the 26 MeV and 63.3 MeV curves crosses the targeted intersection point for the same dimensions of the collection volume. While the 26 MeV case generally underestimate the cross section, the 63.3 MeV case suggests a few possible configurations of the collection volume. In figure 4.12 it is shown how the energy of a 29 MeV proton beam is attenuated as it first travels through air and then the different materials of the device. In the nearby location of the sensitive region, the energy of the beam has been reduced to about 15 MeV. For protons in silicon this is at the energy threshold of producing non-elastic interactions. Small variation in the beam energy or in the simulation geometry may therefore impact the simulated cross section result. In fact a seperate simulation was carried out where the incident



(a)

(b)

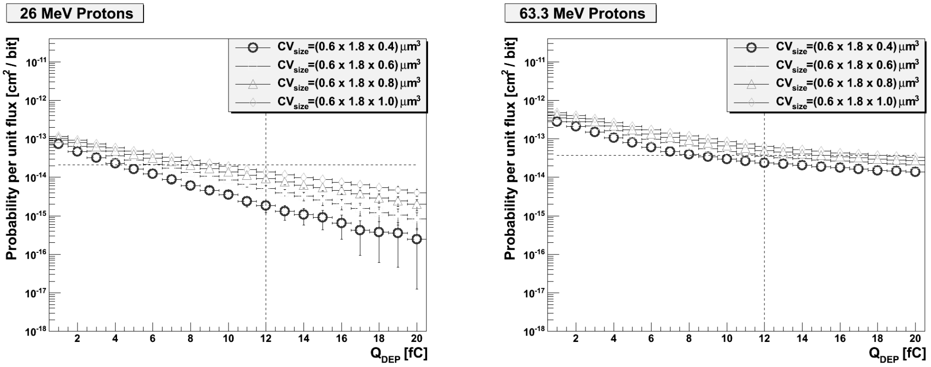
Figure 6.10: Results for the depth variability study for a collection volume area of (0.4 x 1.2 μm^2) and primay beam energies of 26 MeV (a) and 63.3 MeV (b).



(a)

(b)

Figure 6.11: Results for the depth variability study for a collection volume area of (0.6 x 1.2 μm^2) and primay beam energies of 26 MeV (a) and 63.3 MeV (b).



(a)

(b)

Figure 6.12: Results for the depth variability study for a collection volume area of (0.6 x 1.8 μm^2) and primay beam energies of 26 MeV (a) and 63.3 MeV (b).

energy was slightly increased to 30 MeV. The results are compared in tables 6.2 and 6.3 where a factor 1.7-3 increase in the SEU cross section is seen from 26 MeV to 30 MeV. Matching the experimental and simulated cross sections at these low energies consequently proves more difficult than at higher energies, above where the cross section curve reaches its plateau. Therefore, a firm conclusion on the size of the collection volume can not be drawn based on these simulation results. Still, the data clearly suggest that the dimensions are larger than the nominal suggested starting point of 0.4 by 0.8 by 0.4 μm^3. Furthermore, the simulation results show that even with limited geometrical information, it is possible to reproduced the experimental SEU cross sections within reasonable expectations. As pointed out in [80], when dealing with a dynamic values such as the critical charge and SEU measurements, results are acceptable within an unusually large range of error. For example, errors of 25%, 75% and even a factor 2 is reported in [80] and references therein. Another example where 35% is considered an accceptable error is reported in [68]. This is in line with differences between experimental and simulated results obtained in this thesis work[3].

6.4.2 Contribution from α-particles and heavy fragments

In section 4.2.1 figure 4.12 it is shown how the energy of the proton beam is attenuated to approximately 15 MeV when reaching the region of interest within the

[3] Note that the referenced errors should not be mistaken for the errors given in tables 6.2 and 6.3 which are purely statistical.

Simulated SEU cross sections for CV area of $0.6 \cdot 1.8\,\mu m^2$ and Qdep $\geq$ 12fC			
Depth [μm]	σ_{sim} (63.3 MeV)	σ_{sim} (30 MeV)	σ_{sim} (26 MeV)
0.4	$2.3 \cdot 10^{-14} \pm 10\%$	$5.5 \cdot 10^{-15} \pm 18\%$	$1.8 \cdot 10^{-15} \pm 34\%$
0.6	$3.5 \cdot 10^{-14} \pm 8\%$	$1.1 \cdot 10^{-14} \pm 14\%$	$5.1 \cdot 10^{-15} \pm 19\%$
0.8	$5.0 \cdot 10^{-14} \pm 6\%$	$1.7 \cdot 10^{-14} \pm 10\%$	$9.2 \cdot 10^{-15} \pm 14\%$
1.0	$6.4 \cdot 10^{-14} \pm 5\%$	$2.4 \cdot 10^{-14} \pm 9\%$	$1.4 \cdot 10^{-14} \pm 12\%$

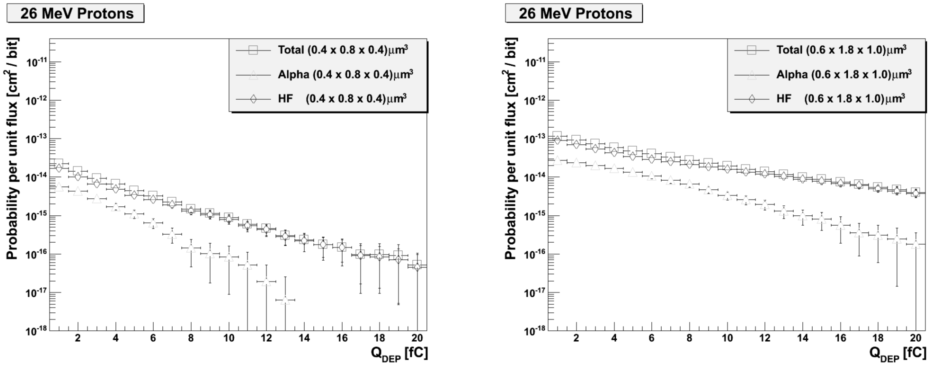
Table 6.2: Simulated SEU cross section for incident proton beams of 63.3 MeV, 30 MeV and 26 MeV for a sensitive area of $0.6 \cdot 1.8\,\mu m^2$ and for different collection depths. σ_{exp}(63.3 MeV) $= 3.7 \cdot 10^{-14}$ cm^{-2} [16], σ_{exp}(26 MeV) $= 2.1 \cdot 10^{-14}$ cm^{-2}. Uncertainties are statistical only.

Simulated SEU cross sections for CV area of $0.6 \cdot 1.2\,\mu m^2$ and Qdep $\geq$ 12fC			
Depth [μm]	σ_{sim} (63.3 MeV)	σ_{sim} (30 MeV)	σ_{sim} (26 MeV)
0.4	$1.6 \cdot 10^{-14} \pm 10\%$	$3.6 \cdot 10^{-15} \pm 19\%$	$1.2 \cdot 10^{-15} \pm 33\%$
0.6	$2.3 \cdot 10^{-14} \pm 8\%$	$6.9 \cdot 10^{-15} \pm 13\%$	$3.2 \cdot 10^{-15} \pm 19\%$
0.8	$3.2 \cdot 10^{-14} \pm 7\%$	$1.1 \cdot 10^{-14} \pm 11\%$	$6.0 \cdot 10^{-15} \pm 14\%$
1.0	$3.7 \cdot 10^{-14} \pm 6\%$	$1.5 \cdot 10^{-14} \pm 10\%$	$8.9 \cdot 10^{-15} \pm 13\%$

Table 6.3: Simulated SEU cross section for incident proton beams of 63.3 MeV, 30 MeV and 26 MeV for a sensitive area of $0.6 \cdot 1.2\,\mu m^2$ and for different collection depths. σ_{exp}(63.3 MeV) $= 3.7 \cdot 10^{-14}$ cm^{-2} [16], σ_{exp}(26 MeV) $= 2.1 \cdot 10^{-14}$ cm^{-2}. Uncertainties are statistical only.

irradiated device. This energy level is at the threshold of α-particle production. In figure 6.15 it is clearly seen that it is the heavier fragments (HF) and not the α-particles that contributes to the total cross section at a primary proton energy of 26 MeV. At 63.3 MeV the contribution from α-particles becomes significant for lower values of the critical charge. Analysing the collision data file produced after the first step in the simulation, figure 6.15(a) and 6.15(b) shows the charge distribution of particles produced in non-elastic reactions. The distribution is sampled within a 100 μm region in the silicon substrate just before the location of the sensitive collection volumes. In this region the average energy of the protons is 16 MeV for a primary beam energy of 26 MeV, and 59 MeV for a primary beam energy of 63.3 MeV. It can be seen that for the lowest energy the production of α–particles is negligible compared to the silicon recoil. This explains the low contribution to the total cross section at 26 MeV.

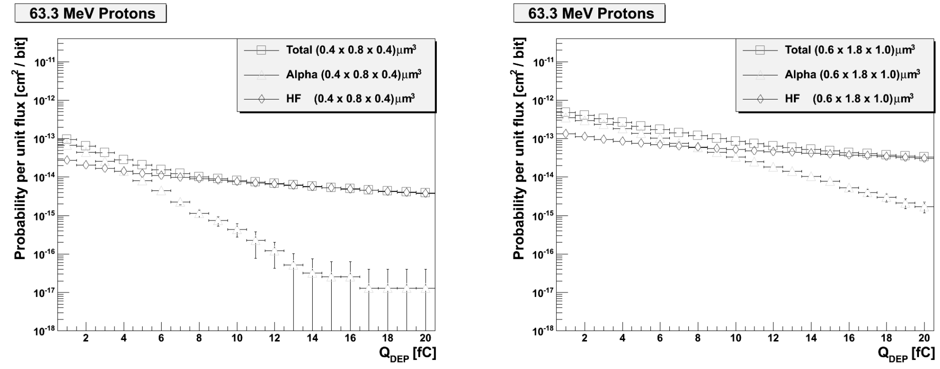
At a primary beam energy of 63.3 MeV the average energy of the produced α-particles is approximately 6 MeV, corresponding to a dE/dx of 126 keV/μm or 5.6 fC/μm. The longest paths a particle can travel through the volume sizes used in figures 6.14(a) and 6.14(b) are close to 1 μm and 2 μm. Given that not all particles will take the longest path through a collection volume, these numbers can explain why the contribution of the α-particles starts to become significant below 4 fC and 7 fC for the respective volume sizes. With an average critcal charge of 12 fC, α-particles may only play a minor role for the Xilinx Virtex II Pro.



(a)

(b)

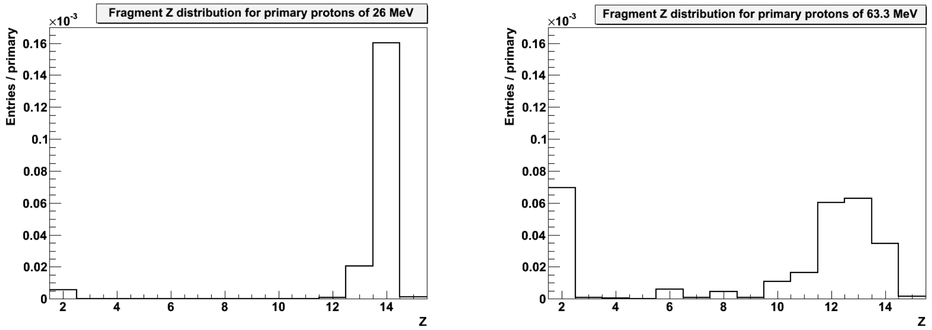
Figure 6.13: Contribution to total cross section at 26 MeV from secondary α-particles and heavy fragments (Z>2) produced in the non-elastic reactions. Simulations where carried out for one small (a) and one large (b) volume size.



(a)

(b)

Figure 6.14: Contribution to total cross section at 63.3 MeV from secondary α-particles and heavy fragments (Z>2) produced in the non-elastic interactions. Simulations where carried out for one small (a) and one large (b) volume size.



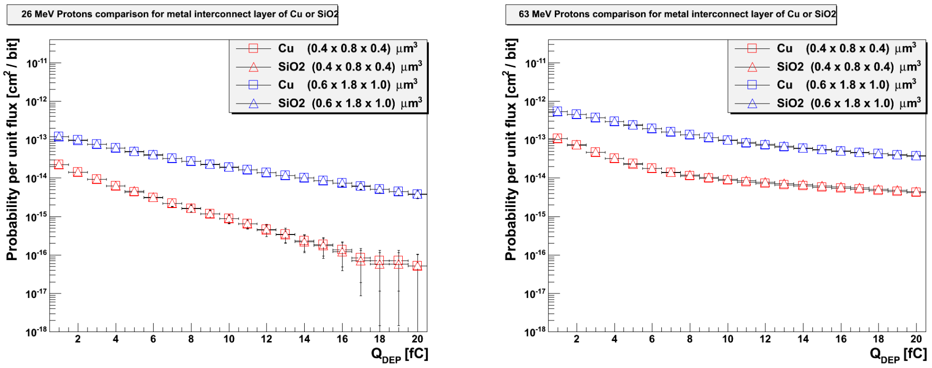
(a)

(b)

Figure 6.15: Charge distribution of the fragments produced in non-elastic interactions in the silicon substrate.(a) Primary proton beam of 26 MeV. (b) Primary proton beam of 63.3 MeV. Protons, neutrons and photons are not included in the plots.

6.4.3 Role of metal interconnect layers

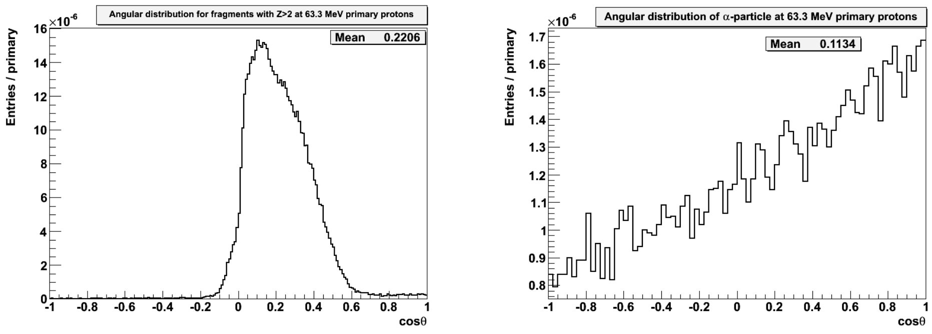
A source of uncertainty in the simulation setup is the description of the metal interconnect layers. However, the simulation results presented in figure 6.16 shows that an accurate description is of less importance for this specific and preliminary study. For each proton beam energy simulations were carried out where the layer representing the metal interconnects either were fully filled with copper or fully filled with SiO_2. The results show no significant difference in the probability curve for neither the smallest nor the largest volume size used. Due to reaction kinematics the majority of the fragments are forward peaked with respect to the beam direction as can be seen in figure 6.17. Thus when the beam first passes the layer of sensitive collection volume before the metal interconnect layers, as were the case for the flip-chip package and irradiation test setup in chapter 4, most fragments produced in the metal interconnect layers will not deposit charge in the collection volumes. However in a scenario where the beam enters from the other direction, or the device is placed in an isotropic radiation environment, it should be foreseen that the metal interconnect layers or other non-symmetric geometry may play a more significant role.



(a)

(b)

Figure 6.16: No significant difference is seen between the simulations when the layer of metal interconnects is filled with either copper or SiO_2. This is true for both a small and large collection volume and for primary proton beam of 26 MeV (a) or 63.3 MeV (b).



(a)

(b)

Figure 6.17: Fluka simulations of the angular distribution with respect to beam axis of fragments produced in non-elastice interaction throughout the target geometry. (a) All fragments with Z > 2. (b) α-particles.

6.5 Summary

A preliminary simulation case study was carried out and compared to experimental data. Applying variability analysis the study also aimed at determining the dimensions of the sensitive volume of the target device. As discussed in 6.4.1 this proved difficult due to the uncertainties connected with using proton energies close to the production threshold of non-elastic interactions. Small variations in the primary beam energy or in the device geometry may therefore significantly change the result. Finding a match at low energies demands a much higher accuracy in the experimental setup and geometry description than what was achievable for this case study. Consequently the size of the collection volume should be determined by comparing simulation results and experimental results at two or more energies well above the non-elastic interaction threshold.

The simulation results further shows that even with limited geometrical information, experimental SEU cross sections can be acceptably reproduced. Monte Carlo simulation can therefore be used as a tool to study how the SEU cross section is affected by different beam energies, particle types, and geometrical modifications. Further work that can provide valuable information for the RCU main FPGA, is to study how the SEU cross section will be affected by a mixed radiation field compared to using a mono-energetic beam. Other possible studies can be to study the impact of beam orientation. This can provide information on how fragments produced in the metal interconnect layers may contribute to the SEU cross section. Moreover, this may give valuable input on how future irradiation tests should be carried out.

Finally, the needed effort to finalize the SEMM2.vBergen for a production run is of great interest as this will provide an additional source of comparison.

Chapter 7

Conclusion and outlook

The work presented in this thesis was carried out in order to investigate the use of an SRAM based FPGA in the ALICE TPC detector radiation environment. Due to the nature of this radiation environment, single event upsets are expected to be the main source of radiation induced failures. Causing an FPGA configuration memory element to change its stored value, single event upsets can consequently lead to malfunction of the user design operating on the FPGA. As the RCU main FPGA is in charge of reading out detector data, a single event upset can cause this readout chain to break down. The result can be temporary loss of data. In [11] it has even been pointed out that in its utmost consequence, a single event upset has the potential to abort the ongoing run. The focus of this thesis has therefore been to investigate the single event upset induced failure probability for the RCU main FPGA in the TPC radiation environment.

Irradiation tests show that the single event upsets probability is low when considering each FPGA as an isolated case. However, due to the large number of FPGAs utilized in the front-end electronics, the overall probability becomes significant. Experiencing 4-8 single event upset induced functional failures in the RCU main FPGA during an ALICE run is therefore a realistic scenario. An important part of this thesis work has therefore been to develop, implement and test a state-of-the art solution to correct single event upsets in the configuration memory of the RCU main FPGA. This is a solution that can be run in the background without interrupting the normal operation of the FPGA. As it will prevent accumulation of single event upsets in the configuration memory in addition to reducing their life time, it is part of an overall mitigation strategy to reduce the probability of functional failures in the FPGA. Irradiation tests have nevertheless shown that for this solution to have a significant effect, it must be combined with additional mitigation on the level of the FPGA user design.

In an SRAM based FPGA the user defined functionality is stored in an array of

SRAM memory cells where only a small number of these memory cells are utilized for any given design. Consequently the ratio between single event upsets and measurable functional failures in the user design is greater than 1. This ratio, usually determined by irradiation tests, can be applied to scale the single event upset rate to obtain the functional failure rate of a design. However, this ratio is strongly coupled to the specifics of the implemented design. Predictions of the functional failure rate for the RCU main FPGA can therefore only be based on tests of the final user design. Due to the complexity of the front-end electronics readout chain, irradiation tests may prove technically difficult. Fault injection has therefore been presented as an alternative solution. Purely based on software, it has been implemented in the existing front-end electronics without the need for hardware modifications. It has been validated by comparable irradiation test results, and further testing has shown that it can be used to enforce a more targeted mitigation strategy during a development phase. The suggested continuation of this work is to carry out fault injection on the final RCU main FPGA design. This will establish the design specific ratio of sensitive bits to functional failures, which again can be used to more accurately predict the expected number of functional failures during operation.

Finally physics based Monte Carlo simulations have been applied to a case study of the RCU main FPGA. Within the given uncertainties, experimental results of the single event upset cross section were reasonably reproduced. Explanations for observed discrepancies were suggested based on further analysis of the simulation results. Potential applications of physics based Monte Carlo simulations are many: studying the critical charge and SEU cross section of a memory cell, contribution to SEU cross sections from different types of reactions, total dose deposition studies, and structural and material layout studies in order to reduce charge deposition in sensitive regions. For experiments like ALICE, an interesting possibility is to simulate the response of a device when exposed to a mixed radiation field. This can be very difficult to achieve in accelerated beam tests where usually mono-energetic beams of one particle type are used. Further work should therefore be carried out to: increase the accuracy in the geometry description of the RCU main FPGA, obtain additional experimental data for comparison in order to better determine the dimensions of the sensitive volume, and increase the efficiency of the presented simulation methodology.

The work and analysis carried out in this thesis demonstrate a viable technique to use programmable devices in radiation exposed areas such as high energy physics experiments. The system is designed to tolerate a certain level of SEUs by instead developing solutions to mitigate the effect of these SEUs. This method allows to introduce the flexibility offered by FPGAs in radiation exposed electronics of running experiments, contrary to traditional ASIC designs. In future upgrades of LHC and new experiments like CBM[1] and ILC[2], the increased complexity and demand for rapid development will make the use of FPGAs even more attractive. However, with potentially harsher radiation environments the need for extensive radiation tolerance studies will also increase. This will make tools like fault injection analysis and physics based Monte Carlo simulations even more important in the design of new detector electronics.

[1] Compressed Baryonic Matter

[2] International Linear Collider

Appendix A

AliRoot simulations of the TPC radiation environment

A.1 Previous work

Tables A.1 and A.2 presents the summarized results from the simulations of the radiation environment in [29]. A 1 mm thick silicon disc was used to score the fluences at each side of the TPC. The silicon disc was divided in 4 concentric circular scoring regions with increasing radial distances from the beam line. Scoring region 1 is the innermost while scoring region 4 is the outermost.

Layers	1	2	3	4
Neutron	$3.42 \cdot 10^{-5}$	$2.57 \cdot 10^{-5}$	$2.13 \cdot 10^{-5}$	$1.85 \cdot 10^{-5}$
Neutron $\mathrm{E_{kin}} > 10$ MeV	$2.61 \cdot 10^{-6}$	$1.60 \cdot 10^{-6}$	$1.05 \cdot 10^{-6}$	$7.49 \cdot 10^{-7}$
Proton	$1.03 \cdot 10^{-7}$	$6.04 \cdot 10^{-8}$	$3.93 \cdot 10^{-8}$	$4.01 \cdot 10^{-8}$
Proton $\mathrm{E_{kin}} > 10$ MeV	$9.96 \cdot 10^{-8}$	$5.89 \cdot 10^{-8}$	$3.79 \cdot 10^{-8}$	$3.88 \cdot 10^{-8}$
$\mathrm{Pion}^{\pm}$	$2.92 \cdot 10^{-7}$	$4.37 \cdot 10^{-7}$	$3.71 \cdot 10^{-7}$	$2.23 \cdot 10^{-7}$
$\mathrm{Pion}^{\pm}$ $\mathrm{E_{kin}} > 10$ MeV	$2.91 \cdot 10^{-7}$	$4.36 \cdot 10^{-7}$	$3.71 \cdot 10^{-7}$	$2.23 \cdot 10^{-7}$
Sum $\mathrm{E_{kin}} > 10$ MeV	$3.00 \cdot 10^{-6}$	$2.10 \cdot 10^{-6}$	$1.46 \cdot 10^{-6}$	$1.01 \cdot 10^{-6}$

Table A.1: Particle fluences (particles/(cm^2 primary)) for a minimum-bias Pb-Pb run (absorber side). Summarized from table C.4 in [29].

Layers	1	2	3	4
Neutron	$1.27 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$	$1.27 \cdot 10^{-5}$	$1.27 \cdot 10^{-5}$
Neutron $E_{kin} > 10$ MeV	$8.70 \cdot 10^{-7}$	$5.80 \cdot 10^{-7}$	$4.47 \cdot 10^{-7}$	$3.56 \cdot 10^{-7}$
Proton	$1.52 \cdot 10^{-7}$	$7.22 \cdot 10^{-8}$	$6.34 \cdot 10^{-8}$	$3.61 \cdot 10^{-8}$
Proton $E_{kin} > 10$ MeV	$1.50 \cdot 10^{-7}$	$7.10 \cdot 10^{-8}$	$6.18 \cdot 10^{-8}$	$3.54 \cdot 10^{-8}$
Pion$^{\pm}$	$8.94 \cdot 10^{-7}$	$5.13 \cdot 10^{-7}$	$3.65 \cdot 10^{-7}$	$2.42 \cdot 10^{-7}$
Pion$^{\pm}$ $E_{kin} > 10$ MeV	$8.93 \cdot 10^{-7}$	$5.11 \cdot 10^{-7}$	$3.64 \cdot 10^{-7}$	$2.42 \cdot 10^{-7}$
Sum $E_{kin} > 10$ MeV	$1.91 \cdot 10^{-6}$	$1.16 \cdot 10^{-6}$	$8.73 \cdot 10^{-7}$	$6.33 \cdot 10^{-7}$

Table A.2: Particle fluences (particles/(cm^2 primary)) for a minimum-bias Pb-Pb run (non-absorber side). Summarized from table C.5 in [29].

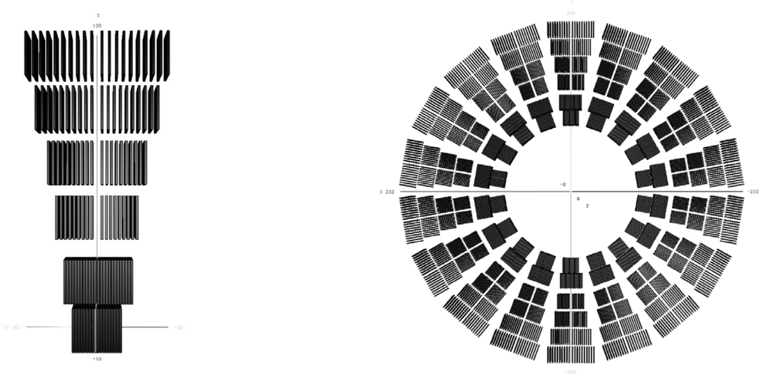
A.2 Geometry description

A.2.1 Description of front-end cards

The basic structure of the new geometry is the Front End Card (FEC). All together one chamber/sector contains 121 FECs, one side of the TPC contains $18 \cdot 121 = 2178$ FECs, and the full TPC contains 4356 FECs. In the AliRoot implementation the FEC structure is composed by a thin copper housing surrounding the actual FEC PCB. First the copper housing is created and filled with a volume of air leaving only a 0.5 mm thin copper wall. The FEC PCB is then placed inside the volume of air. The dimensions of these volumes are listed in table A.3. Figure A.1(a) shows the FECs implemented for one sector. This description of FECs is then translated and rotated into the 18 sectors on each side of the TPC. The final result for one side is shown in figure A.1(b).

Volume/Material	X [cm]	Y [cm]	Z [cm]
Copper	19	1	17
Air	18.9	0.9	16.9
PCB	18.8	0.1	16.8

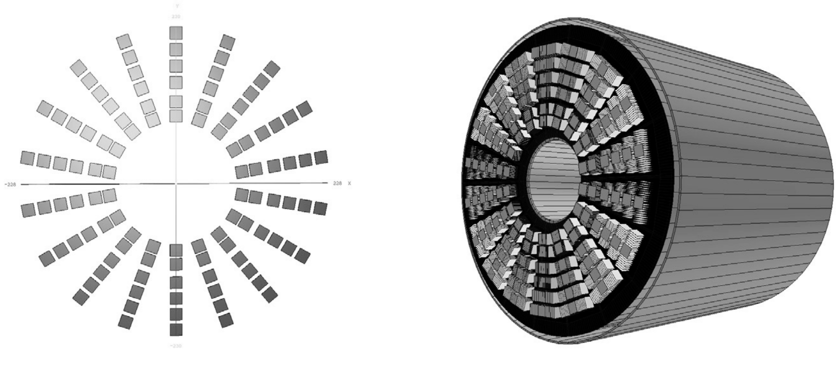
Table A.3: Dimensions of the volumes and material composing the FEC structure.



(a)

(b)

Figure A.1: (a) Front end cards for one full chamber with 6 readout partitions. (b) Front end cards translated and rotated around the full TPC end plate.



(a)

(b)

Figure A.2: (a) Geometry description showing six rings of in total 108 RCUs for one side of the TPC. (b) The final implementation for the FECs and the RCUs shown for one side of the TPC.

A.2.2 The RCU scoring region

The purpose of the simulation is to determine the radiation field at the location of the RCU main FPGA. In principle a scoring area equal to the surface area of the RCU main FPGA should be used. To gain more statistics an area of approximately the size of the RCU motherboard is used instead. That is, a rectangular volume of $18x18x0.1cm^3$ is implemented to describe the RCU. A thickness of 1 mm in the beam direction (Z) is used to resemble the thickness of the silicon die. There are in all 216 readout partitions in the TPC and thus 216 RCU motherboards. Figure A.2(a) shows how the RCUs are distributed in the location of the readout partitions for one side of the TPC. This makes up six rings where each ring contains 18 RCUs. Each ring of 18 RCUs corresponds to a scoring region for the simulations presented in this thesis. In this way particle fluences can be studied at six radial distances from the centre and outwards. Figure A.2(b) shows the final implementation of the front end cards (yellow) and the RCU volumes (green) for one side of the TPC.

A.2.3 The C++ code of the geometry description

Listing A.1: Geometry description of front-end cards and RCU

```
//-----------------------------------------------------------------
// Front end electronics
// Added by KR HiB/UiB Norway
//-----------------------------------------------------------------

//Disk that will contain all the Front end cards
TGeoTube *disk = new TGeoTube(90,250.,8.6);
TGeoVolume *vDisk = new TGeoVolume("TPC_disk",disk,m1);



//some geometric measures for the FEC
Int_t fecA[6] = {9,12,9,10,10,10};  //number of FECs on the A branch
Int_t fecB[6] = {9,13,9,10,10,10};  //number of FECs on the B branch
Int_t noPart = 6; //no of partitions

//y distance between the FEC for each partitions
Double_t dytr_tab[6] = {1.2,1.2,1.9,2.1,2.7,3.2};
//x distance between each partions dxdr_tab[6] is a dummy value(not used)
Double_t dxtr_tab[6] = {2,9,5,5,5,0};
//distance between branch A and B for each partition
Double_t dy_between_brancAB[6] = {0.2,0.5,3.5,3.5,3.5,3.5};
Double_t RCURadius[6];
RCURadius[0] = 0;
Int_t R = 100; //radius at which FEE starts
Double_t dPhi = 20.*TMath::DegToRad();
for(int i = 1 ; i<6;i++){
  RCURadius[i] = RCURadius[i-1] + dxtr_tab[i-1] + 9.5*2;
}

//get medium
TGeoMedium *mCu=gGeoManager->GetMedium("TPC_Cu");
//added in AliTPC.cxx based on FMD
TGeoMedium *mPCB=gGeoManager->GetMedium("TPC_PCBFEC");

TGeoVolumeAssembly *contFEC = new TGeoVolumeAssembly("TPC_FECASS");

Double_t dxtr = 0;
Double_t dytr = 0;
//FEC construction
//MakeBox is using half lengths
TGeoVolume *CuHousing=gGeoManager->MakeBox("TPC_CUHOUSING",mCu,9.5,0.5,8.5);
TGeoVolume *FECAir = gGeoManager->MakeBox("TPC_FECAIR",m1,9.45,0.45,8.45);
TGeoVolume *FEC =     gGeoManager->MakeBox("TPC_FEC",mPCB,9.4,0.05,8.4);
//position the air and FEC inside the copper volume/housing
TGeoTranslation *trFEC  = new TGeoTranslation(0.,0.2,0.);
TGeoTranslation *trAir  = new TGeoTranslation(0.,0.,0.);
FECAir->AddNode(FEC,1,trFEC);
CuHousing->AddNode(FECAir,1,trAir);
int noContMod = 0;


//add all FEC for one chamber to the contFEC assembly
for(Int_t k = 0 ; k<noPart; k++){  //loop over all partitions
  dytr = (dy_between_brancAB[k]/2);
  for(Int_t i = 0; i < fecA[k]; i++)  //loop over no. FECs in branch A
  {
    noContMod++;
    //position FEC
    //(dxtr-xs,dytr-ys,0.);
    TGeoTranslation *tr  = new TGeoTranslation(dxtr,dytr+0.5,0.);
    contFEC->AddNode(CuHousing,noContMod,tr);
    dytr += dytr_tab[k];

  }
  dytr = -(dy_between_brancAB[k]/2);
  for(Int_t i = 0; i < fecB[k]; i++)  //loop over no. FECs in branch B
  {
    noContMod++;
```

```
      //position FEC
      //(dxtr-xs, dytr-ys, 0.);
      TGeoTranslation *tr  = new TGeoTranslation(dxtr, dytr - 0.5, 0.);
      contFEC->AddNode(CuHousing, noContMod, tr);
      dytr -= dytr_tab[k];

    }
    //add hight of FEC + distance between partitions
    dxtr += 19 + dxtr_tab[k];
  }

  //translate the chamber assembly to the 18 TPC sectors and add
  //it to the disk that will be placed in the TPC volume
  for(Int_t i =0;i<18;i++){
    Double_t phi = (dPhi*i)+ openingAngle ;
    TGeoRotation *r = new TGeoRotation();
    r->RotateZ((20*i)+10);
    Double_t dy = R*TMath::Sin(phi);
    Double_t dx = R*TMath::Cos(phi);
    vDisk->AddNode(contFEC, i, new TGeoCombiTrans(dx, dy, 0, r));
  }

  //Construct all the RCU volumes used as scoring regions
  char RCUName[1024];

  // - side (muon side)
  TGeoVolumeAssembly *tpcFEEC = new TGeoVolumeAssembly("TPC_FEE_C");
  Int_t cntFEEC=0;
  for(Int_t i =0;i<18;i++){
    Double_t phi = (dPhi*i)+ openingAngle ;
    TGeoRotation *r = new TGeoRotation();
    r->RotateZ((20*i)+10);
    //cntFEEC++;
    //  tpcFEEC->AddNode(sFECv, cntFEEC, new TGeoCombiTrans(0,0,0,r));
    for(Int_t k=0; k<6;k++){
      Double_t dy = (RCURadius[k]+R)*TMath::Sin(phi);
      Double_t dx = (RCURadius[k]+R)*TMath::Cos(phi);

      sprintf(RCUName,"RCU_C_%d",k);
      TGeoBBox *box = new TGeoBBox(9,9,0.05,0);
      TGeoVolume *RCU  = new TGeoVolume(RCUName, box, m9);
      RCU->SetLineColor(3);
      RCU->SetFillColor(2);
      cntFEEC++;
      tpcFEEC->AddNode(RCU, cntFEEC, new TGeoCombiTrans(dx, dy, 0, r));//
    }
  }
```

```
//+ side (non muon side)
TGeoVolumeAssembly *tpcFEEA = new TGeoVolumeAssembly(”TPC_FEE_A”);
Int_t cntFEEA=0;
for(Int_t i =0;i<18;i++){
  Double_t phi = (dPhi*i)+ openingAngle ;
  TGeoRotation *r = new TGeoRotation();
  r->RotateZ((20*i)+10);
  // cntFEEA++;
  //  tpcFEEA->AddNode(sFECv,cntFEEA,new TGeoCombiTrans(0,0,0,r));
  for(Int_t k=0; k<6;k++){
    Double_t dy = (RCURadius[k]+R)*TMath::Sin(phi);
    Double_t dx = (RCURadius[k]+R)*TMath::Cos(phi);

    sprintf(RCUName,”RCU_A%d”,k);
    TGeoBBox *box = new TGeoBBox(9,9,0.05,0);
    TGeoVolume *RCU  = new TGeoVolume(RCUName,box,m9);
    RCU->SetLineColor(3);
    RCU->SetFillColor(2);
    cntFEEA++;
    tpcFEEA->AddNode(RCU,cntFEEA,new TGeoCombiTrans(dx,dy,0,r));
  }
}

//add one disk of FEC to each side of the TPC
v1->AddNode(vDisk,5,new TGeoTranslation(0.,0.,-270));
v1->AddNode(vDisk,6,new TGeoTranslation(0.,0.,270));
//add the RCU scoring regions to the TPC
v1->AddNode(tpcFEEC,3,new TGeoTranslation(0.,0.,-279));
v1->AddNode(tpcFEEA,4,new TGeoTranslation(0.,0.,279));
//end construction of FEC——————————————
```

A.3 Visual check using energy scoring

Cartesian energy scoring was used to visualize the implemented front end electronics. The results are compared to a run without the front-end electroncis. For each run (with and without FEC) 40000 primary tracks where transported in the TPC. Table A.4 gives the details of the cartesian scoring setup. A Fluka backend graphical user interface called flukaGUI was used to produce the plots showing the energy scoring in figures A.3 through A.5. The energy cut-off was set to 10 MeV for all particles except neutrons which includes all energies. The sole purpose of these plots was to validate that the geometry description of the front-end electronics was implemented and accounted for in the simulations.

TPC Side	X-bins	Y-bins	Z-bins	X_{min}	X_{max}	Y_{min}	Y_{max}	Z_{min}	Z_{max}
Absorber (C)	300	300	60	-300	300	-300	300	-280	-250
Non-absorber (A)	300	300	60	-300	300	-300	300	250	280

Table A.4: Cartesian scoring regions for absorber and non-absorber side.

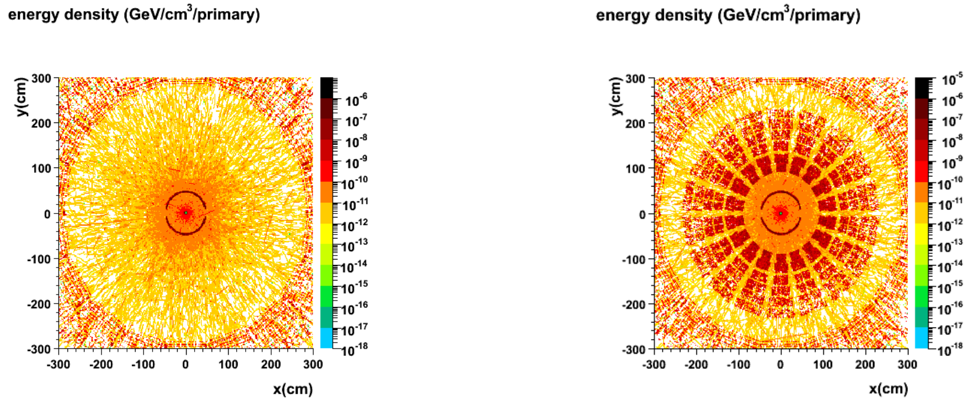


Figure A.3: Energy deposition scoring in the position $278 > Z > 262$ non-absorber side. Left is without FEC and right is with FEC.

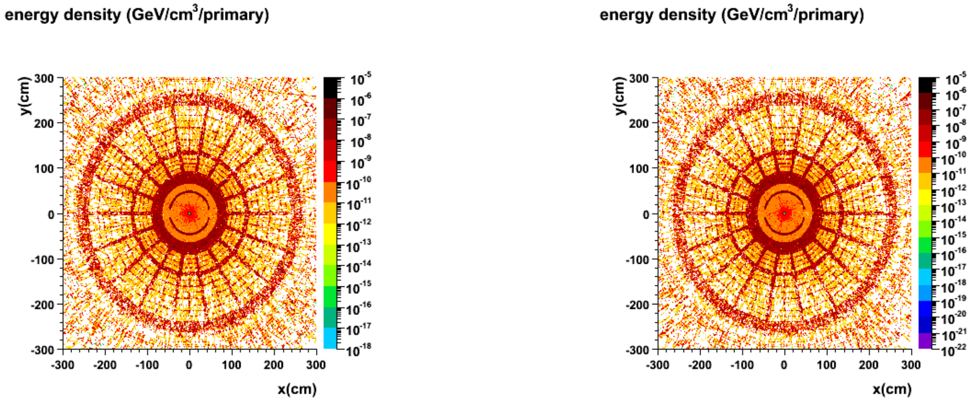


Figure A.4: Energy deposition scoring in the position $260 > Z > 252$ non-absorber side. This shows the structure of the readout chamber at the end cap of the TPC. Left is without FEC and right is with FEC.

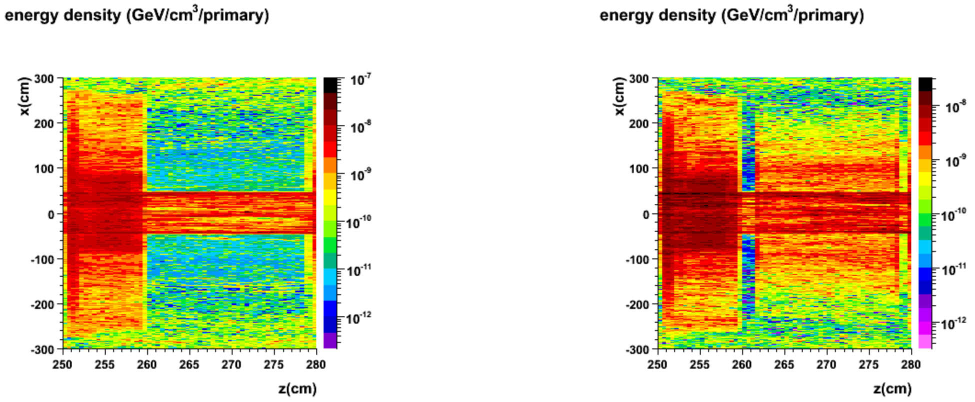
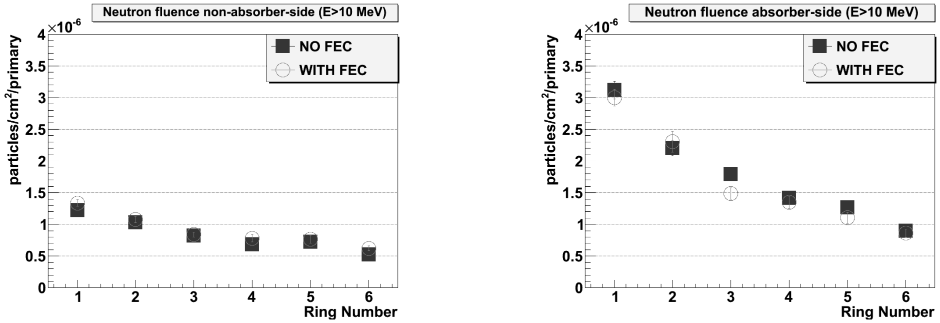


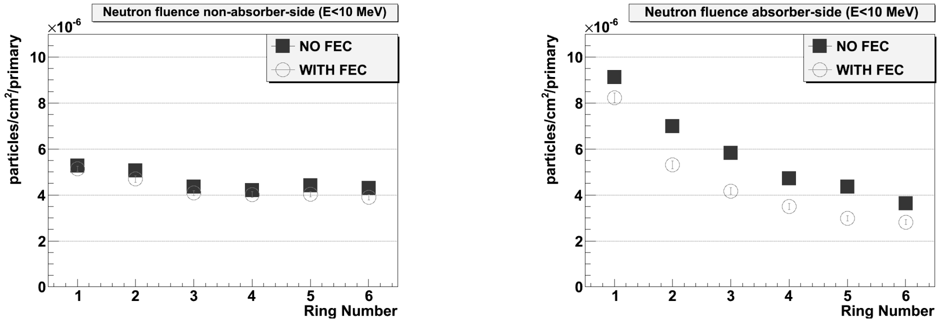
Figure A.5: Energy deposition scoring in the position $280 > Z > 250$ non-absorber side. Left is without FEC and right is with FEC.

A.4 Fluence results for the 6 scoring regions



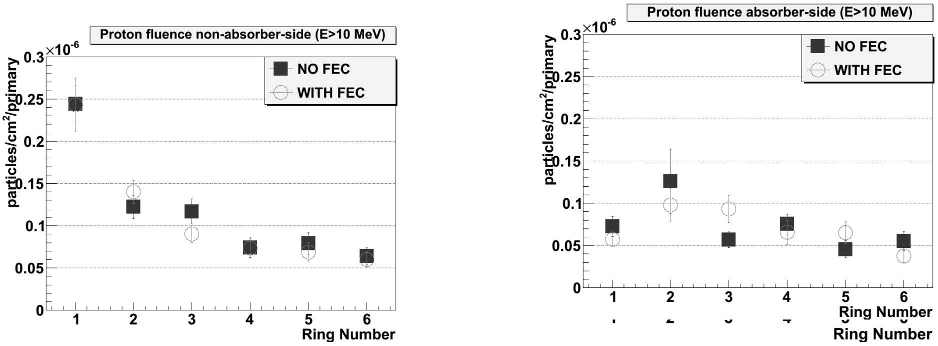
(a) (b)

Figure A.6: Fluence of neutrons ($E_{kin} > 10MeV$) for the non-absorber(a) and absorber(b) side as a function of radial distance from the beam line.



(a) (b)

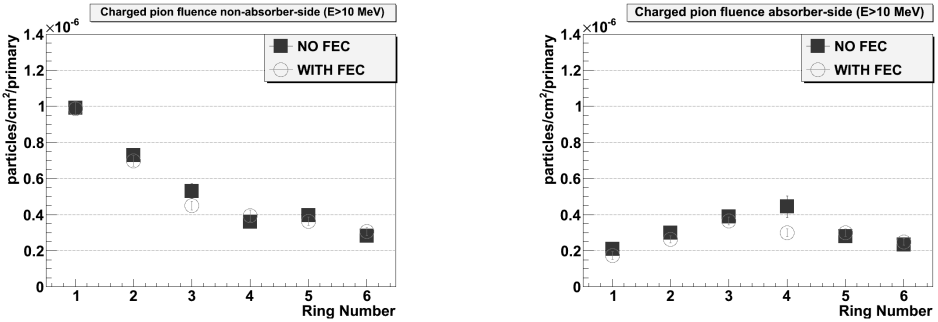
Figure A.7: Fluence of neutrons ($E_{kin} < 10MeV$) for the non-absorber(a) and absorber(b) side as a function of radial distance from the beam line.



(a)

(b)

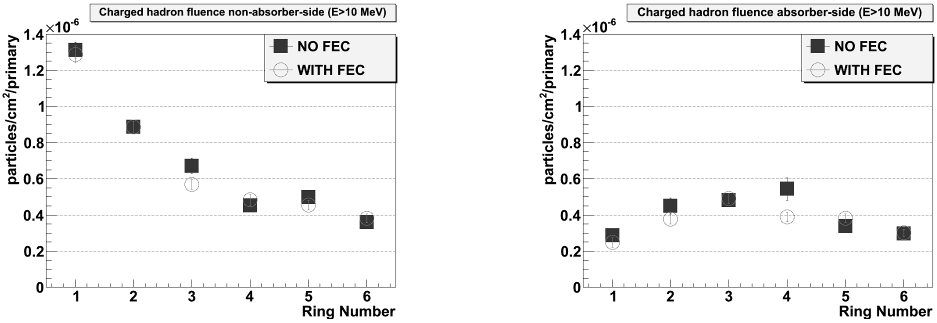
Figure A.8: Fluence of protons ($E_{kin} > 10MeV$) for the non-absorber(a) and absorber(b) side as a function of radial distance from the beam line.



(a)

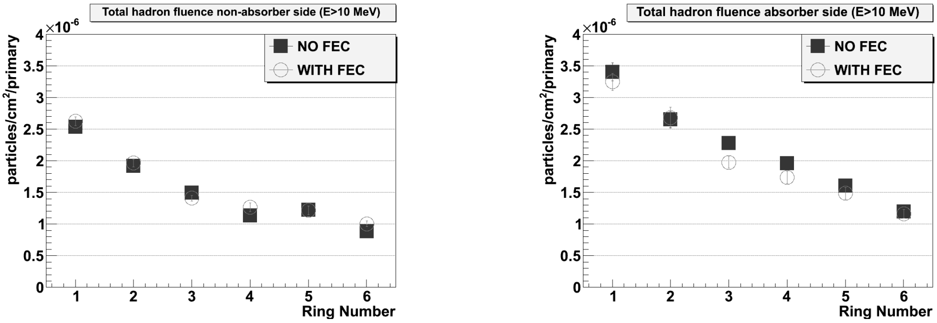
(b)

Figure A.9: Fluence of charged pions ($E_{kin} > 10MeV$) for the non-absorber(a) and absorber(b) side as a function of radial distance from the beam line.



(a) (b)

Figure A.10: Fluence of charged hadrons ($E_{kin} > 10MeV$) for the non-absorber(a) and absorber(b) side as a function of radial distance from the beam line.



(a) (b)

Figure A.11: Total fluence of energetic hadrons ($E_{kin} > 10MeV$) for the non-absorber(a) and absorber(b) side as a function of radial distance from the beam line. (Neutrons + charged hadrons)

Fluence [$particles/cm^2/primary$]	Scoring region absorber side (WITH FEC)		
	1	2	3
Protons	6.22e-08 ± 13.9%	9.95e-08 ± 19.6%	9.52e-08 ± 16.7%
Protons $E_{\text{kin}} > 10$ MeV	5.75e-08 ± 15.0%	9.81e-08 ± 19.9%	9.33e-08 ± 17.1%
Neutrons	1.12e-05 ± 2.2%	7.61e-06 ± 3.1%	5.66e-06 ± 2.9%
Neutrons $E_{\text{kin}} > 10$ MeV	3.00e-06 ± 4.4%	2.30e-06 ± 7.1%	1.48e-06 ± 6.0%
Charged Pions	1.73e-07 ± 13.3%	2.68e-07 ± 6.9%	3.65e-07 ± 6.0%
Charged Pions $E_{\text{kin}} > 10$ MeV	1.73e-07 ± 13.3%	2.63e-07 ± 7.0%	3.65e-07 ± 6.0%
Charged Hadrons	2.54e-07 ± 10.6%	3.84e-07 ± 7.0%	4.93e-07 ± 5.5%
Charged Hadrons $E_{\text{kin}} > 10$ MeV	2.49e-07 ± 10.8%	3.78e-07 ± 7.1%	4.91e-07 ± 5.6%

Table A.5: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 1 - Ring 3) on the absorber side. Simulations were run with the geometry of the front-end cards implemented. Energy cut set to 0.1MeV for all particles.

Fluence [$particles/cm^2/primary$]	Scoring region absorber side (WITH FEC)		
	4	5	6
Protons	6.87e-08 ± 20.8%	6.69e-08 ± 19.4%	3.94e-08 ± 19.3%
Protons $E_{\text{kin}} > 10$ MeV	6.57e-08 ± 21.7%	6.53e-08 ± 19.8%	3.76e-08 ± 20.2%
Neutrons	4.84e-06 ± 3.4%	4.09e-06 ± 3.6%	3.68e-06 ± 3.4%
Neutrons $E_{\text{kin}} > 10$ MeV	1.35e-06 ± 7.2%	1.10e-06 ± 8.0%	8.59e-07 ± 7.4%
Charged Pions	3.01e-07 ± 7.1%	3.01e-07 ± 6.8%	2.49e-07 ± 7.5%
Charged Pions $E_{\text{kin}} > 10$ MeV	3.00e-07 ± 7.1%	3.01e-07 ± 6.8%	2.49e-07 ± 7.5%
Charged Hadrons	3.93e-07 ± 6.6%	3.84e-07 ± 6.4%	3.03e-07 ± 6.7%
Charged Hadrons $E_{\text{kin}} > 10$ MeV	3.89e-07 ± 6.7%	3.82e-07 ± 6.4%	3.01e-07 ± 6.7%

Table A.6: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 4 - Ring 6) on the absorber side. Simulations were run with the geometry of the front-end cards implemented. Energy cut set to 0.1MeV for all particles.

Fluence [$particles/cm^2/primary$]	Scoring region non-absorber side (WITH FEC)		
	1	2	3
Protons	2.48e-07 ± 12.6%	1.44e-07 ± 9.6%	9.40e-08 ± 11.1%
Protons $E_{\text{kin}} > 10$ MeV	2.43e-07 ± 12.8%	1.40e-07 ± 9.8%	9.04e-08 ± 11.5%
Neutrons	6.47e-06 ± 2.0%	5.77e-06 ± 2.7%	4.93e-06 ± 2.3%
Neutrons $E_{\text{kin}} > 10$ MeV	1.34e-06 ± 4.0%	1.08e-06 ± 4.5%	8.39e-07 ± 4.7%
Charged Pions	9.87e-07 ± 3.4%	6.99e-07 ± 4.2%	4.51e-07 ± 5.5%
Charged Pions $E_{\text{kin}} > 10$ MeV	9.86e-07 ± 3.4%	6.99e-07 ± 4.2%	4.50e-07 ± 5.5%
Charged Hadrons	1.29e-06 ± 3.6%	8.90e-07 ± 3.6%	5.74e-07 ± 4.7%
Charged Hadrons $E_{\text{kin}} > 10$ MeV	1.29e-06 ± 3.6%	8.86e-07 ± 3.7%	5.69e-07 ± 4.8%

Table A.7: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 1 - Ring 3) on the non-absorber side. Simulations were run with the geometry of the front-end cards implemented. Energy cut set to 0.1MeV for all particles.

Fluence [$particles/cm^2/primary$]	Scoring region non-absorber side (WITH FEC)		
	4	5	6
Protons	7.41e-08 ± 15.0%	6.95e-08 ± 15.4%	6.45e-08 ± 15.2%
Protons $E_{\text{kin}} > 10$ MeV	7.35e-08 ± 15.1%	6.91e-08 ± 15.5%	5.99e-08 ± 16.3%
Neutrons	4.80e-06 ± 2.8%	4.80e-06 ± 2.8%	4.52e-06 ± 2.6%
Neutrons $E_{\text{kin}} > 10$ MeV	7.83e-07 ± 7.7%	7.63e-07 ± 8.2%	6.22e-07 ± 5.8%
Charged Pions	3.96e-07 ± 7.9%	3.63e-07 ± 6.4%	3.06e-07 ± 7.0%
Charged Pions $E_{\text{kin}} > 10$ MeV	3.94e-07 ± 8.0%	3.63e-07± 6.4%	3.06e-07 ± 7.0%
Charged Hadrons	4.85e-07 ± 6.8%	4.55e-07 ± 5.7%	3.85e-07 ± 6.3%
Charged Hadrons $E_{\text{kin}} > 10$ MeV	4.83e-07 ± 6.9%	4.54e-07 ± 5.7%	3.81e-07 ± 6.4%

Table A.8: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 4 - Ring 6) on the non-absorber side. Simulations were run with the geometry of the front-end cards implemented. Energy cut set to 0.1MeV for all particles.

Fluence [$particles/cm^2/primary$]	Scoring region absorber side (NO FEC)		
	1	2	3
Protons	7.60e-08 ± 15.6%	1.31e-07 ± 29.2%	6.05e-08 ± 15.6%
Protons $E_{\text{kin}} > 10$ MeV	7.24e-08 ± 16.3%	1.26e-07 ± 30.2%	5.70e-08 ± 16.5%
Neutrons	1.22e-05 ± 2.0%	9.20e-06 ± 2.4%	7.62e-06 ± 3.1%
Neutrons $E_{\text{kin}} > 10$ MeV	3.12e-06 ± 4.5%	2.20e-06 ± 5.4%	1.79e-06 ± 5.4%
Charged Pions	2.10e-07 ± 11.0%	3.02e-07 ± 7.2%	3.93e-07 ± 6.8%
Charged Pions $E_{\text{kin}} > 10$ MeV	2.09e-07 ± 11.0%	3.00e-07 ± 7.3%	3.90e-07 ± 6.9%
Charged Hadrons	2.92e-07 ± 9.0%	4.55e-07 ± 9.6%	4.87e-07 ± 6.0%
Charged Hadrons $E_{\text{kin}} > 10$ MeV	2.87e-07 ± 9.2%	4.49e-07 ± 9.7%	4.81e-07 ± 6.0%

Table A.9: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 1 - Ring 3) on the absorber side. Simulations were run without the geometry of the front-end cards. Energy cut set to 0.1 MeV for all particles.

Fluence [$particles/cm^2/primary$]	Scoring region absorber side (NO FEC)		
	4	5	6
Protons	7.99e-08 ± 15.0%	4.62e-08 ± 22.1%	5.70e-08 ± 20.2%
Protons $E_{\text{kin}} > 10$ MeV	7.54e-08 ± 15.9%	4.57e-08 ± 22.4%	5.54e-08 ± 20.8%
Neutrons	6.13e-06 ± 3.2%	5.61e-06 ± 3.2%	4.53e-06 ± 3.7%
Neutrons $E_{\text{kin}} > 10$ MeV	1.42e-06 ± 6.4%	1.26e-06 ± 7.5%	8.96e-07 ± 8.0%
Charged Pions	4.44e-07 ± 13.5%	2.80e-07 ± 7.6%	2.39e-07 ± 7.8%
Charged Pions $E_{\text{kin}} > 10$ MeV	4.44e-07 ± 13.5%	2.80e-07 ± 7.6%	2.35e-07 ± 7.9%
Charged Hadrons	5.48e-07 ± 11.2%	3.39e-07 ± 7.1%	3.03e-07 ± 7.2%
Charged Hadrons $E_{\text{kin}} > 10$ MeV	5.44e-07 ± 11.3%	3.38e-07 ± 7.1%	2.97e-07 ± 7.3%

Table A.10: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 4 - Ring 6) on the absorber side. Simulations were run without the geometry of the front-end cards. Energy cut set to 0.1MeV for all particles.

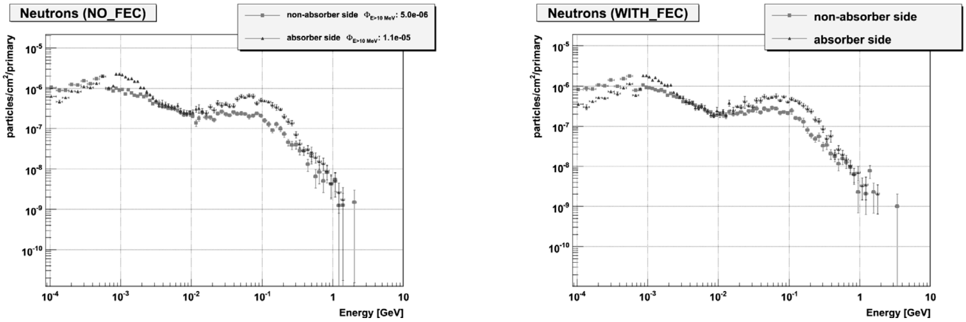
Fluence [$particles/cm^2/primary$]	Scoring region non-absorber side (NO FEC)		
	1	2	3
Protons	2.50e-07 ± 8.6%	1.25e-07 ± 11.1%	1.18e-07 ± 12.5%
Protons $\mathrm{E_{kin}} > 10$ MeV	2.44e-07 ± 8.8%	1.22e-07 ± 11.3%	1.17e-07 ± 12.7%
Neutrons	6.49e-06 ± 2.3%	6.09e-06 ± 2.5%	5.18e-06 ± 2.6%
Neutrons $\mathrm{E_{kin}} > 10$ MeV	1.22e-06 ± 4.2%	1.03e-06 ± 5.1%	8.24e-07 ± 4.6%
Charged Pions	9.95e-07 ± 3.6%	7.32e-07 ± 4.4%	5.34e-07 ± 7.5%
Charged Pions $\mathrm{E_{kin}} > 10$ MeV	9.91e-07 ± 3.6%	7.30e-07 ± 4.4%	5.31e-07 ± 7.6%
Charged Hadrons	1.32e-06 ± 3.3%	8.91e-07 ± 4.0%	6.76e-07 ± 6.4%
Charged Hadrons $\mathrm{E_{kin}} > 10$ MeV	1.31e-06 ± 3.3%	8.86e-07 ± 4.1%	6.72e-07 ± 6.5%

Table A.11: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 1 - Ring 3) on the non-absorber side. Simulations were run without the geometry of the front-end cards. Energy cut set to 0.1 MeV for all particles.

Fluence [$particles/cm^2/primary$]	Scoring region non-absorber side (NO FEC)		
	4	5	6
Protons	7.74e-08 ± 15.7%	7.96e-08 ± 16.2%	6.56e-08 ± 15.6%
Protons $\mathrm{E_{kin}} > 10$ MeV	7.42e-08 ± 16.3%	7.90e-08 ± 16.4%	6.42e-08 ± 15.9%
Neutrons	4.88e-06 ± 2.7%	5.13e-06 ± 2.9%	4.82e-06 ± 2.5%
Neutrons $\mathrm{E_{kin}} > 10$ MeV	6.82e-07 ± 6.1%	7.22e-07 ± 6.9%	5.21e-07 ± 6.5%
Charged Pions	3.62e-07 ± 7.1%	3.98e-07 ± 7.1%	2.83e-07 ± 7.8%
Charged Pions $\mathrm{E_{kin}} > 10$ MeV	3.60e-07 ± 7.1%	3.97e-07 ± 7.1%	2.83e-07 ± 7.8%
Charged Hadrons	4.57e-07 ± 6.4%	5.00e-07 ± 6.2%	3.62e-07 ± 6.8%
Charged Hadrons $\mathrm{E_{kin}} > 10$ MeV	4.52e-07 ± 6.5%	4.98e-07 ± 6.3%	3.60e-07 ± 6.8%

Table A.12: Particle fluences (particles/cm^2/primary) for scoring regions (Ring 4 - Ring 6) on the non-absorber side. Simulations were run without the geometry of the front-end cards. Energy cut set to 0.1MeV for all particles.

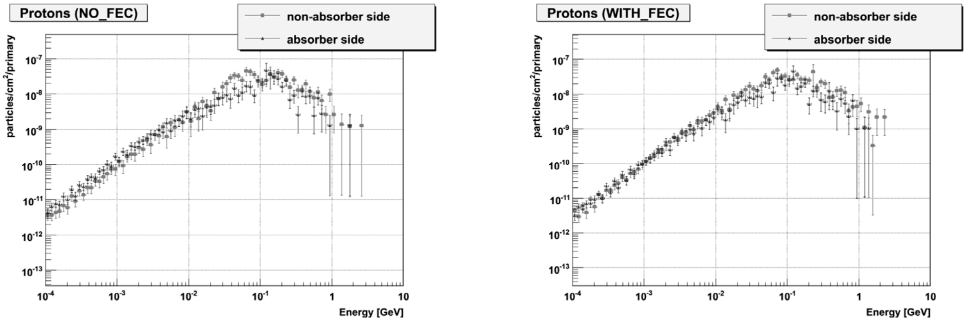
A.5 Fluence as a function of energy



(a)

(b)

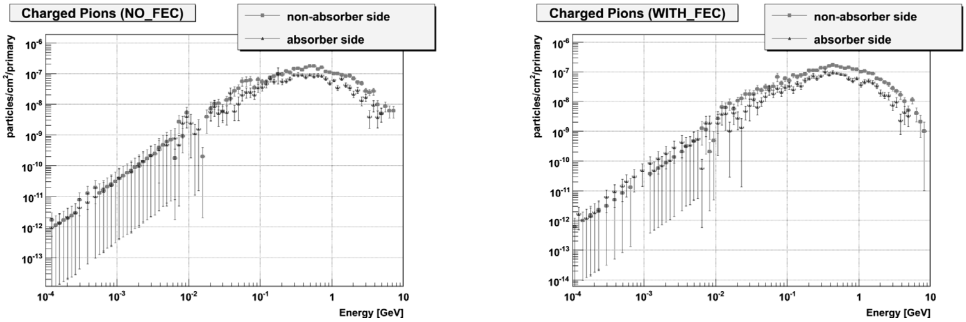
Figure A.12: Fluence of neutrons as a function of energy summed over all scoring regions without front-end cards (a), and with front-end cards included(b).



(a)

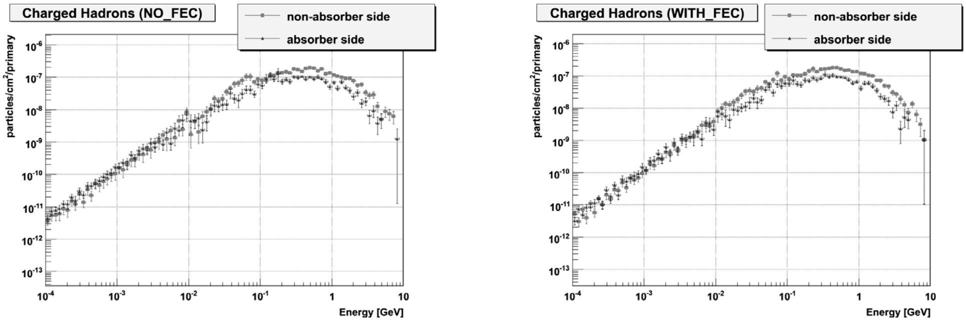
(b)

Figure A.13: Fluence of protons as a function of energy summed over all scoring regions without front-end cards (a), and with front-end cards included(b).



(a) (b)

Figure A.14: Fluence of charged pions as a function of energy summed over all scoring regions without front-end cards (a), and with front-end cards included(b).



(a) (b)

Figure A.15: Fluence of charged hadrons as a function of energy summed over all scoring regions without front-end cards (a), and with front-end cards included(b).

Appendix B

Flow diagram of the FRVC procedure

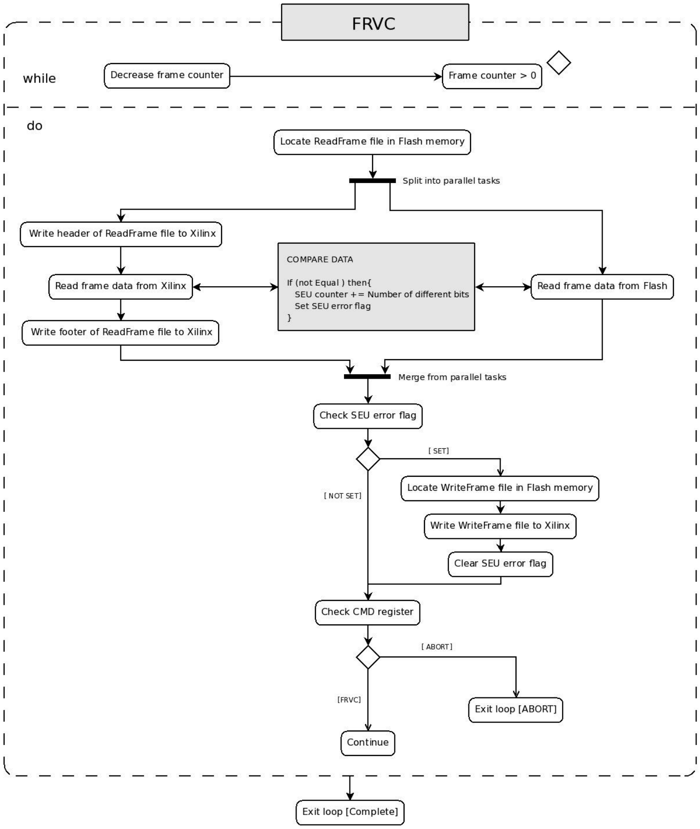

FRVC
while
Decrease frame counter
Frame counter > 0
do
Locate ReadFrame file in Flash memory
Split into parallel tasks
Write header of ReadFrame file to Xilinx
Read frame data from Xilinx
Write footer of ReadFrame file to Xilinx
COMPARE DATA
If (not Equal) then{
SEU counter += Number of different bits
Set SEU error flag
}
Read frame data from Flash
Merge from parallel tasks
Check SEU error flag
[SET]
[NOT SET]
Locate WriteFrame file in Flash memory
Write WriteFrame file to Xilinx
Clear SEU error flag
Check CMD register
[ABORT]
[FRVC]
Exit loop [ABORT]
Continue
Exit loop [Complete]

Figure B.1: Flow diagram of the FRVC procedure.

Appendix C

Irradiation test results

C.1 SEU cross section results

The Thin Film Breakdown Counter [45][46] used for flux measurements during the irradiation tests was produced by A.N.Smirnov, V.G. Khlopin Radium Institute, $2^{n}d$ Murinskiy Prospect 28, St.Petersburg 194021, Russia. The sensitivity at 25-26 MeV is $1.5 \cdot 10^{-8}$ within an accuracy of ±13%.

The results of the individual irradiation test runs at OCL are listed in table C.1 and C.2. During irradiation of the FPGAs, a scintillator was used as a relative fluence monitor. Before irradiation this scintillator was calibrated with the thin film break down counters.

id	Scintillator	Time [s]	Scint/TFBC	Flux [p/s$\cdot cm^2$]	SEU	$\sigma_{SEU}[cm^2/bit]$
0	110397	30	10271	$2.39 \cdot 10^7$	51	$2.24 \cdot 10^{-14}$
1	368215	100	10271	$2.39 \cdot 10^7$	140	$1.85 \cdot 10^{-14}$
2	368215	100	10271	$2.39 \cdot 10^7$	119	$1.57 \cdot 10^{-14}$
3	110467	100	10255	$0.72 \cdot 10^7$	56	$2.46 \cdot 10^{-14}$
4	220000	200	10255	$0.71 \cdot 10^7$	83	$1.83 \cdot 10^{-14}$
5	226710	200	10255	$0.74 \cdot 10^7$	94	$2.01 \cdot 10^{-14}$
6	218813	200	10255	$0.71 \cdot 10^7$	88	$1.95 \cdot 10^{-14}$
7	222475	200	10255	$0.72 \cdot 10^7$	123	$2.68 \cdot 10^{-14}$
8	217181	200	10255	$0.70 \cdot 10^7$	115	$2.57 \cdot 10^{-14}$
9	220247	200	10255	$0.71 \cdot 10^7$	93	$2.05 \cdot 10^{-14}$
10	146952	200	12806	$0.38 \cdot 10^7$	52	$2.14 \cdot 10^{-14}$
11	648392	1000	12806	$0.34 \cdot 10^7$	202	$1.88 \cdot 10^{-14}$
12	119018	201	12806	$0.31 \cdot 10^7$	42	$2.14 \cdot 10^{-14}$
13	646382	1000	12806	$0.34 \cdot 10^7$	219	$2.05 \cdot 10^{-14}$
14	697485	500	12806	$0.73 \cdot 10^7$	268	$2.32 \cdot 10^{-14}$
15	1020163	500	12806	$1.06 \cdot 10^7$	341	$2.02 \cdot 10^{-14}$
16	1003517	500	12806	$1.04 \cdot 10^7$	293	$1.77 \cdot 10^{-14}$
17	1021488	500	12806	$1.06 \cdot 10^7$	364	$2.16 \cdot 10^{-14}$
18	1028797	500	12806	$1.07 \cdot 10^7$	366	$2.15 \cdot 10^{-14}$
19	920346	500	12806	$0.96 \cdot 10^7$	374	$2.46 \cdot 10^{-14}$
20	1124478	500	12806	$1.17 \cdot 10^7$	327	$1.76 \cdot 10^{-14}$
21	1115103	500	12806	$1.16 \cdot 10^7$	287	$1.56 \cdot 10^{-14}$
22	403201	212	12806	$0.99 \cdot 10^7$	113	$1.70 \cdot 10^{-14}$
23	927941	500	12806	$0.97 \cdot 10^7$	289	$1.88 \cdot 10^{-14}$
24	1298297	300	12806	$2.25 \cdot 10^7$	469	$2.19 \cdot 10^{-14}$
25	1362773	200	12806	$3.55 \cdot 10^7$	486	$2.16 \cdot 10^{-14}$

Table C.1: Results from the individual irradiation test runs for OCL period 1.

id	Scintillator	Time [s]	Scint/TFBC	Flux [p/s$\cdot cm^2$]	SEU	$\sigma_{SEU}[cm^2/bit]$
26	448950	300	15461	$0.65 \cdot 10^7$	149	$2.42 \cdot 10^{-14}$
27	426466	306	15461	$0.60 \cdot 10^7$	149	$2.55 \cdot 10^{-14}$
28	433058	300	15461	$0.62 \cdot 10^7$	118	$1.99 \cdot 10^{-14}$
29	806820	400	15461	$0.87 \cdot 10^7$	236	$2.14 \cdot 10^{-14}$
30	802343	400	15461	$0.86 \cdot 10^7$	230	$2.09 \cdot 10^{-14}$
31	1073869	400	15461	$1.16 \cdot 10^7$	349	$2.37 \cdot 10^{-14}$
32	1102102	400	15461	$1.19 \cdot 10^7$	338	$2.24 \cdot 10^{-14}$
33	763212	400	15461	$0.82 \cdot 10^7$	226	$2.16 \cdot 10^{-14}$
34	578768	300	15461	$0.83 \cdot 10^7$	175	$2.21 \cdot 10^{-14}$
35	1049987	500	15461	$0.90 \cdot 10^7$	305	$2.12 \cdot 10^{-14}$
36	1110742	500	15461	$0.96 \cdot 10^7$	338	$2.22 \cdot 10^{-14}$
37	728643	400	15461	$0.78 \cdot 10^7$	238	$2.39 \cdot 10^{-14}$
38	930099	500	15461	$0.80 \cdot 10^7$	267	$2.10 \cdot 10^{-14}$
39	434071	300	14730	$0.65 \cdot 10^7$	152	$2.44 \cdot 10^{-14}$
40	452361	300	14730	$0.68 \cdot 10^7$	123	$1.89 \cdot 10^{-14}$
41	454589	300	14730	$0.69 \cdot 10^7$	130	$1.99 \cdot 10^{-14}$
42	459400	300	14730	$0.69 \cdot 10^7$	136	$2.06 \cdot 10^{-14}$
43	446024	300	14730	$0.67 \cdot 10^7$	142	$2.22 \cdot 10^{-14}$
44	440664	300	14730	$0.66 \cdot 10^7$	123	$1.94 \cdot 10^{-14}$
45	444148	300	14730	$0.67 \cdot 10^7$	146	$2.29 \cdot 10^{-14}$
46	430222	300	14730	$0.65 \cdot 10^7$	136	$2.20 \cdot 10^{-14}$
47	415376	300	14730	$0.63 \cdot 10^7$	113	$1.89 \cdot 10^{-14}$
48	407933	300	14730	$0.61 \cdot 10^7$	115	$1.96 \cdot 10^{-14}$
49	404209	300	14730	$0.61 \cdot 10^7$	129	$2.22 \cdot 10^{-14}$
50	1307839	600	14730	$0.99 \cdot 10^7$	434	$2.31 \cdot 10^{-14}$
51	1229882	600	14730	$0.93 \cdot 10^7$	356	$2.01 \cdot 10^{-14}$
52	3116010	600	14730	$2.35 \cdot 10^7$	945	$2.11 \cdot 10^{-14}$
53	3669538	600	14730	$2.77 \cdot 10^7$	1101	$2.09 \cdot 10^{-14}$
54	2361790	600	14145	$1.86 \cdot 10^7$	710	$2.01 \cdot 10^{-14}$
55	2166914	600	14145	$1.70 \cdot 10^7$	636	$1.96 \cdot 10^{-14}$
56	3901386	600	14145	$3.06 \cdot 10^7$	1190	$2.04 \cdot 10^{-14}$
57	5208940	600	14145	$4.09 \cdot 10^7$	1464	$1.88 \cdot 10^{-14}$
58	4353841	600	14145	$3.42 \cdot 10^7$	1255	$1.93 \cdot 10^{-14}$
59	4597769	600	14145	$3.61 \cdot 10^7$	1379	$2.00 \cdot 10^{-14}$
60	4366041	600	14145	$3.43 \cdot 10^7$	1263	$1.93 \cdot 10^{-14}$

Table C.2: Results from the individual irradiation test runs for OCL period 2.

C.1.1 Total dose calculation

The die of Xilinx Virtex-II Pro is approximately 1 cm^2. For the active semiconductor region of transistor a depth dx=2 μm is assumed. This makes a volume of $2{\cdot}10^{-4}$cm^2. The mass of this volume is

$$M = \rho \cdot V_{si} = 2.32g/cm^3 \cdot 2 \cdot 10^{-4} cm^2 = 4.64 \cdot 10^{-7} kg. \tag{C.1}$$

The absorbed dose in a material is measured in the unit gray (Gy) or Rad where

$$1Gy = \frac{1joule}{kg} = 100Rad. \tag{C.2}$$

The stopping power (dE/dx) of a 15 MeV proton in silicon is 5.9 keV/μm [23]. The conversion factor between eV and joule is

$$C = 1eV = 1.602\dot{1}0^{-19}J \tag{C.3}$$

During the irradiation test the FPGA was exposed to a total fluence of a total of $\Phi_p = 4 \cdot 10^{10}$ protons. The total absorbed dose can then be calculated:

$$Dose_{absorbed} = \frac{\frac{dE}{dx} \cdot dx \cdot N_p \cdot 1.602 \cdot 10^{-19}}{M} \approx 162Gy \tag{C.4}$$

Appendix D

Class diagram of fault injection software

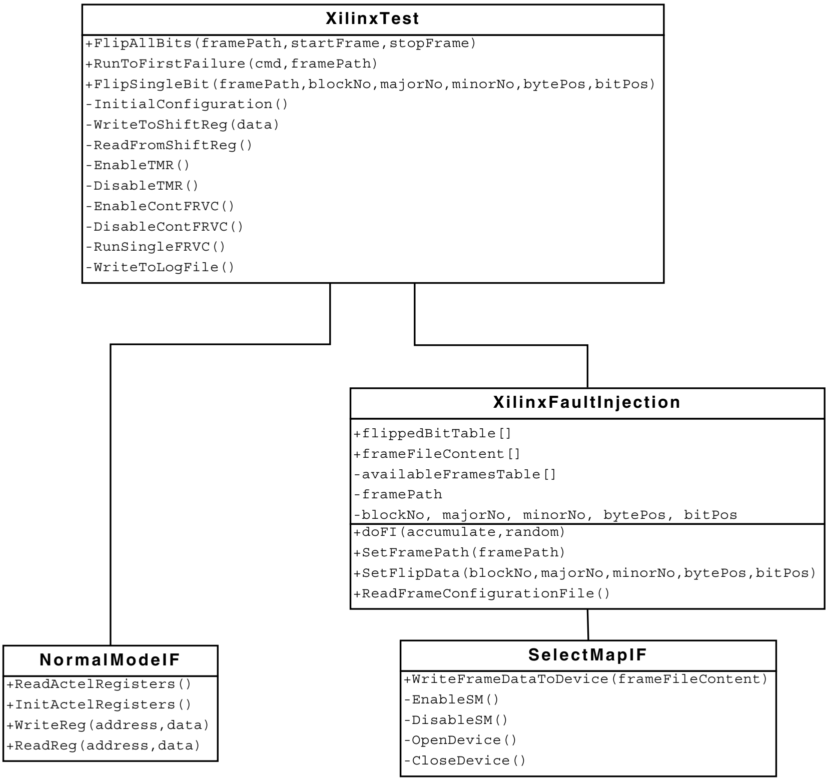


Figure D.1: Class diagram showing the classes of the fault injection software.

Appendix E

SEU Monte Carlo simulation

E.1 FPGA geometry analysis

A structural analysis was carried out for both a Xilinx Virtex-II Pro 7 and a Xilinx Virtex-II Pro 4. These are based on the similar technology process but contain different amounts of the logical resources. The main parameters measured are listed in table E.1. A FIB image example of a solder bump is shown in figure E.1.

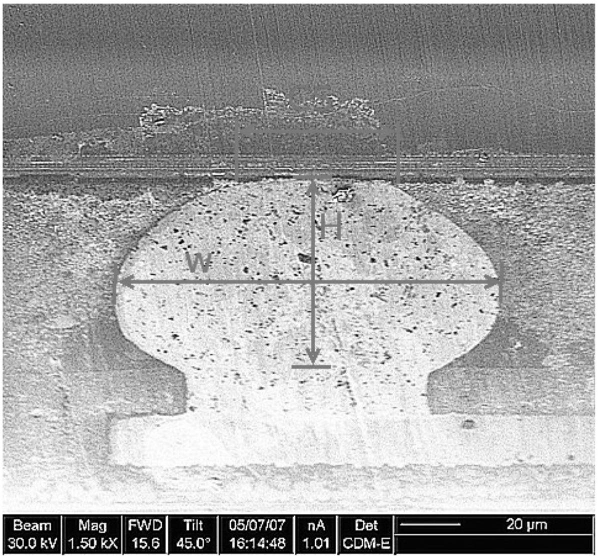


Figure E.1: FIB image showing the dimensions of the flip chip solder ball

The thickness of the interconnect layers are estimated from the FIB images

Parameter	Measured value	
	XC2VP4	XC2VP7
Number of frames	884	1320
Number of bits (N_T)	2998528	4477440
Die geometry (A_{die})	7 mm x 9 mm	8.9 mm x 11.1 mm
Average calculated area per bit (A_{bit})	21 μm^2	22 μm^2
Lid thickness	450 μm	450 μm
Die thickness	854 μm	852
Interconnect thickness	11 μm	9 μm
Package substrate	771 μm	NA
Solder bump width (W)	100 μm	NA
Solder bump height (H)	70 μm	NA
Solder bump contact surface (CS)	40 μm	NA
Center to center of solder bumps	250 μm	NA

Table E.1: Geometry parameters of the Xilinx Virtex-II Pro. The die length and width in addition to the copper lid thickness is measured using a digital ruler. The values for the die, interconnect and package substrate thickness is estimates based on visual measurements from FIB images. Package substrate is measured from bottom of solder bump to top of solder ball. The solder bump contact surface is the diameter of the surface where there is full contact between the interconnect layers and the solder ball. NA: Not measured.

in figure 6.4. Table E.2 lists a set of nominal values used as input data for the simulation setups. The metal fraction is based on the average ratio of visual metal to the visual oxide. As the images only represents a very small part of the full chip, it is recommended to carry out a variability study to investigate the importance of different metal fraction values.

Layer	Layer id	Thickness dz [μm]	Material 1	Material 2	fm1
Oxide 1	ox1	0.25	Cu	SiO_2	0.1
Metal 1	m1	0.5	Cu	SiO_2	0.7
Oxide 2	ox2	0.25	Cu	SiO_2	0.1
Metal 2	m2	0.5	Cu	SiO_2	0.7
Oxide 3	ox3	0.25	Cu	SiO_2	0.1
Metal 3	m3	0.5	Cu	SiO_2	0.7
Oxide 4	ox4	0.25	Cu	SiO_2	0.1
Metal 4	m4	0.5	Cu	SiO_2	0.7
Oxide 5	ox5	0.25	Cu	SiO_2	0.1
Metal 5	m5	0.5	Cu	SiO_2	0.7
Oxide 6	ox6	0.25	Cu	SiO_2	0.1
Metal 6	m6	0.5	Cu	SiO_2	0.7
Oxide 7	ox7	0.5	Cu	SiO_2	0.1
Metal 7	m7	1.0	Cu	SiO_2	0.7
Oxide 8	ox8	0.5	Cu	SiO_2	0.1
Metal 8	m8	1.0	Cu	SiO_2	0.7
Oxide 9	ox9	0.5	Cu	SiO_2	0.1
Metal 9	m9	1.0	Cu	SiO_2	0.7

Table E.2: Table of nominal values for the metal interconnect layers. Each layer is associated with a thickness dz, two materials, and a value $fm1$ giving the fraction of material 1 to material 2 in this layer.

E.2 Determining an optimal simulation target area

The area A_{sim} of the simulation target is calculated based on the total number of configuration bits, the area of the silicon die, and the number of bits N_B that will be used for the simulation. For simplicity it is assumed that the configuration bits of the FPGA are evenly distributed over the full area of the chip. Thus, on average each bit occupies an area

$$A_{bit} = \frac{A_{die}}{N_T} = L_{bit}^2 \tag{E.1}$$

where N_T is the number of configuration bits for the Xilinx Virtex-II Pro, A_{die} is the area of the silicon die, and L_{bit} is the length of the sides in the square defining the area of the bit. If more than one bit will be used for the simulation, the area is scaled by N_B, where N_B is the number of bits.

$$A_B = N_B A_{bit} = N_B \, \frac{A_{die}}{N_T} = N_B \, L_{bit}^2 \tag{E.2}$$

Considering that the range of possible fragments from a non-elastic interaction can be longer than the L_{bit}, the area of the target should be extended with an extra padding length L_p in each direction. The length of each side of the simulation target is then

$$L_{sim} = 2\,L_p + \sqrt{N_B}\,L_{bit} \tag{E.3}$$

and the total area of the simulation target is

$$A_{sim} = L_{sim}^2 = 4\,L_p^2 + 4\,L_p\sqrt{\frac{N_B\,A_{die}}{N_T}} + \frac{N_B\,A_{die}}{N_T} \tag{E.4}$$

In equation E.4 the two first terms are the area of the padding and the last is the area occupied by the bits. This is illustrated in figure E.2 were

$$A_C = L_p^2 \tag{E.5}$$

and

$$A_S = L_p\sqrt{\frac{N_B\,A_{die}}{N_T}} \tag{E.6}$$

For the TPC radiation environment the main concern are the fragments produced in non-elastic interactions. Due to the low interaction cross section a large initial fluence of primary source particles is needed in order to gain a significant number of SEUs. For accelerated beam tests this simply is a matter of increasing the beam flux as long as the upset rate is kept at a reasonable and detectable level. This is however not a favorable method in Monte Carlo simulations as it can lead to increased simulation time. Keeping the fluence as low as possible is therefore of interest. Combining equations 6.1 and 6.2 the detection efficiency is defined as the number of SEUs per primary source particle

$$\frac{N_{SEU}}{I_0} = \frac{\sigma_{SEU,bit}(E)\,N_B}{A_{sim}} \tag{E.7}$$

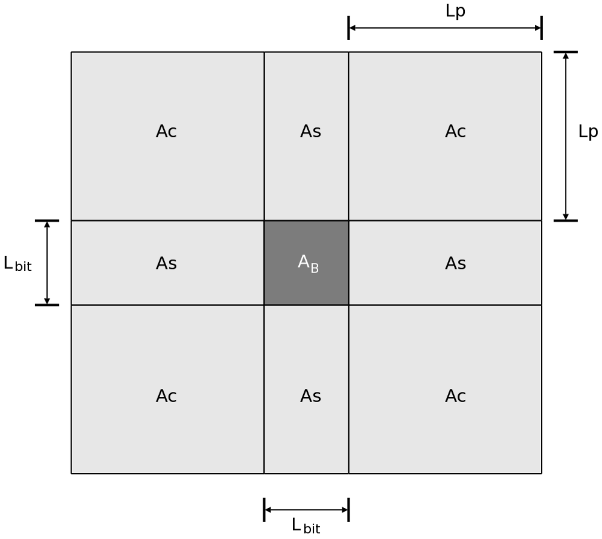


Figure E.2: Simulation target area A_{sim}

$$D_{eff} \equiv \frac{N_{SEU}}{I_0} \propto \frac{N_B}{A_{sim}} \tag{E.8}$$

In figure E.2 the relative detection efficiency is plotted as a function of increasing number of used simulation bits. The relative detection efficiency is here defined as

$$D_R(N_B) = \frac{D_{eff}(N_B)}{D_{eff}(N_T)} \tag{E.9}$$

It can be seen that the relative increase in the detection efficiency is highest in the beginning and slowly saturates after a certain number of N_B. The reason is linked to the increase in the area A_B compared to the padding area. The change in this relative area is highest until

$$\begin{aligned} A_B &= 4\ A_c \\ \Rightarrow N_B &= \frac{4\,L_P^2\,N_T}{A_{die}} \end{aligned} \tag{E.10}$$

Applying the values for the Xilinx Virtex-II Pro from table E.1 in addition to a padding length of $L_P = 25\ \mu$m results in an optimal number simulation bits of $N_B = 113$. The padding length is based on the average range of α-particles produced in non-elastic interactions in the energy range of interest in TPC environment.

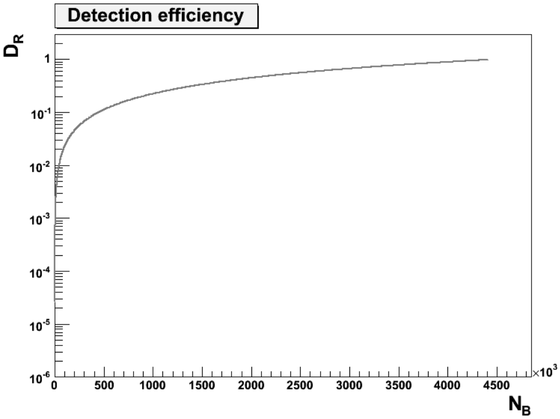


Figure E.3: Relative detection efficiency as a function of the used simulation bits. $D_{max} = D_{eff}(N_B = N_T)$

Rearranging equation E.7, the number of incident particles needed for a simulation can now be calculated when a wanted number of SEUs is specified. As a starting point table E.3 lists the simulation parameters when $N_{SEU} = 100$ using the SEU cross section measured for 29 MeV protons.

Parameter	Value
L_P	25 μm
N_B	113
A_{sim}	$9.97 \cdot 10^{-5}$ cm^2
L_{sim}	100 μm
A_B	$2.49 \cdot 10^{-5}$ cm^2
L_B	50 μm
N_{SEU}	100
$\sigma_{SEU,bit}(E = 29\ MeV)$	$2.1 \cdot 10^{-14}$ cm^2/bit
I_0	$2.38 \cdot 10^{10}$ cm^{-2}

Table E.3: Parameters calculated as starting point values for the Fluka simulation setup.

E.3 SEMM2.vBergen setup specifics

The geometry description is based on the generic input from section 6.2.2 and figure 6.5. The dimension of the simulation structure is defined by xl, yl and the total thickness of all layers in the stack. Each layer is characterized by its location coordinate L_C relative to a reference point, its spatial dimension, and its assigned material mixture. In SEMM2.vBergen the reference point is defined as the origin of the coordinate system with $Y_O = 0$, $X_O = 0$ and $Z_O = 0$.

The xy-plane at Z_O coincides with the top surface of the collection volumes. The layer coordinate L_C is fixed to the $X_C = 0$ and $Y_C = 0$ corner of a layer and relative to the system reference point defined by X_C. Currently each layer is limited to be a mixture of two materials defined by their material id, $id1$ and $id2$. The material fraction factor fm defines the fraction of material of $id1$ to the material of $id2$.

E.3.1 Tables of simulation parameters

This section presents tables of parameters used in the SEMM2.vBergen input file. An example of the input file can be found in section E.3.2. It is divided into three main sections where the first contains the description of the metal interconnect layers. The corresponding parameters are listed in table E.4. Each layer is associated with a layer name and material identification as given in tables E.5 and E.6 respectively. It should be noted that the layer name is not parsed when reading the input file and is only included for documentation purposes.

The preliminary version of SEMM2.vBergen supports the implementation of 1 to 10 collection volumes. Table E.7 lists the parameters that describes the size and location of the collection volume(s). Similar to each layer, the collection volume is defined in space by its own location coordinate (xccv,yccv,zccv). Finally the beam property parameters are listed in table E.8.

Parameter	Description
xo	x-coordinate of geometry reference point
yo	y-coordinate of geometry reference point
zo	z-coordinate of geometry reference point
xl	layer dimension in x-direction [μm]
yl	layer dimension in y-direction [μm]
n	number of layers in target
name	Name of layer
id1	id of metal layer
id2	id of dielectric
fm	Material fraction of idm ($0 \leq \text{fm} \leq 1$) (material fraction of id2 is 1-fm)
dz	layer dimension in z-direction [μm]
z	z-coordinate for the layer reference corner relative to zo [μm]
xscmc	Granularity sampling length [μm]

Table E.4: Parameters used to describe the properties of the metal interconnect layers in the geometry input file

Layer name	Description
lid	Chip lid
sub	Die substrate
ox1 - oxN	Interconnect dielectric layers 1 through N
m1 - mN	Interconnect metal layers 1 through N
ps	Substrate of chip package

Table E.5: Table of names used for the interconnect layers.

id	Number	type
si	1	Silicon
oxide	2	Silicon dioxide
Al-27	3	Aluminum
Cu-64	4	Copper
W184	5	Tungsten

Table E.6: Material ID and corresponding number

ParameterDescription	
ncv	Number of collection volumes
xccv	x-coordinate of collection volume reference point relative to the target reference point Xc
yccv	y-coordinate of collection volume reference point relative to the target reference point Yc
zccv	z-coordinate of collection volume reference point relative to the target reference point Zc.
xlcv	length of collection volume in x-direction
ylcv	length of collection volume in y-direction

Table E.7: Collection volume parameters.

Parameter	Description
idprj	id/type of projectile
ekprj	Kinetic energy of projectile
irseed	Random number seed
noint	Number of inelastic reactions requested within the target
dirprj	Starting point and direction of projectile [±Z]

Table E.8: Beam property parameters.

E.3.2 Input file example

For this case study an input file describing in total 8 collection volumes of different sizes has been prepared.

```
C***********************************************************************
C**   File Content:  3 blocks of Input parameters for SEMM2-BERGEN     *
C*               block #1: Geometry description        *
C*               block #1: collection volume description           *
C*               block #1: Beam property description               *
C***********************************************************************
C***********************************************************************
C**  Block #1: GEOMETRY DESCRIPTION OF TARGET                          *
c*   PARAMETER LIST DESCRIPTION:                                       *
c*   1st line:
c*   xo    - x-coordinate of geometry reference point [default 0]
c*   yo    - y-coordinate of geometry reference point [default 0]
c*   zo    - z-coordinate of geometry reference point [default 0]
c*   xl    - target dimensionin x-direction [um]
c*   yl    - target dimensionin y-direction [um]
c*   n     - number of layers in geometry description
c*
c*   2nd line to n+1 line:
c*   name  - name of layer
c*   id1   - Id of material 1 (typically the metal layer)
c*   id2   - Id of dielectric layer (typically the dielectric layer)
c*   fm    - Material fraction of id1 (0<fm<1)
c*   dz    - Thickness of layer in z-direction [um]
c    z     - z-coordinate for the reference corner relative to xc
c*   xscmc - length scale for the monte carlo sampling [um]
c*
c*----------------------------------------------------------------------
c* BEGIN PARAMETER LIST
c* The geometry description is based on visual analysis of FIB images
c* The targer for the case study is an FPGA. The experimental SEU cross
c* section is based on the following numbers:
c* Total number of SRAM bits in chip: 4519680
c* Number of SRAM bits checked for upsets: 3174912
c* Dimensions of chip 0.89cm x 1.11cm = 0.9879e8 um^2
c* Each SRAM bit occupies then 0.9879e8/4519680 = 21.85 um^2
```

```
c* The total length and width of the device including
c* 25um padding is 60 um
c*
c*                          ------> Z
C*              450um       850um         9um       100um
C*            |-------|-----------------|--------|-----------|
C*            +-------+---------------+-+--------+-----------+ -
C*            |       |               |*|        |           | |
C*  63.3 MeV  |       |               |*| Inter  | Pack.Sub  | |
c*    p-->    |Cu Lid |Si substracte  |*| connect| /Solder   | | 60um
C*            |       |               |*| layers | bum       | |
c*            |       |               |*|        |           | |
c*            +-------+---------------+-+--------+-----------+ -
c*                                    |-|
c*                                   ~1um
c*
c* For the preliminary run the pack.sub is set to oxide
c*
c* xo yo zo xl yl n
c* name idm idd fm dz xcsmc
0 0 0 60 60 21
lid  cu oxide 1.00 450  -1300 1.0
sub  si oxide 1.00 850  -850 1.0
ox1  cu oxide 0.10 0.25 0.00 0.25
m1   cu oxide 0.70 0.50 0.25 0.5
ox2  cu oxide 0.10 0.25 0.75 0.25
m2   cu oxide 0.70 0.50 1.00 0.5
ox3  cu oxide 0.10 0.25 1.50 0.25
m3   cu oxide 0.70 0.50 1.75 0.5
ox4  cu oxide 0.10 0.25 2.25 0.25
m4   cu oxide 0.70 0.50 2.50 0.5
ox5  cu oxide 0.10 0.25 3.00 0.25
m5   cu oxide 0.70 0.50 3.25 0.5
ox6  cu oxide 0.10 0.25 3.75 0.25
m6   cu oxide 0.70 0.50 4.00 0.5
ox7  cu oxide 0.10 0.50 4.50 0.5
m7   cu oxide 0.70 1.00 5.00 1.0
ox8  cu oxide 0.10 0.50 6.00 0.5
m8   cu oxide 0.70 1.00 6.50 1.0
```

```
ox9  cu oxide 0.10 0.50 7.50 0.5
m9   cu oxide 0.70 1.00 8.00 1.0
ps   oxide oxide 0.63 100 9.00 1.0
c* END PARAMETER LIST
c*******************************************************************
c*******************************************************************
c*   Block #2: COLLECTION VOLUME DESCRIPTION
c*
c*   PARAMETER LIST
c*
c*   1st line:
c*   ncv   - number of collection volumes
c*
c*   2nd line to ncv+1 line
c*   xccv - x-coordinate of the reference point of the collection volume
c*           relative to target reference point zc
c*   yccv - y-coordinate of the reference point of the collection volume
c*           relative to target reference point zc
c*   zccv - z-coordinate of the reference point of the collection volume
c*           relative to target reference point zc
c*   xlcv - dimension of collection volume in x-direction
c*   ylcv - dimension of collection volume in y-direction
c*-------------------------------------------------------------------
c* BEGIN PARAMETER LIST
c*
c*    p = 25 um
c*
c*    For the preliminary run 8 collection volumes of different sizes
C*    are implemented
C*    V1 (xlcv,ylcv,zccv) = (0.4 , 0.8 , 0.8)
C*    V2 (xlcv,ylcv,zccv) = (0.4 , 0.8 , 1.0)
C*    V3 (xlcv,ylcv,zccv) = (0.4 , 1.2 , 0.8)
C*    V4 (xlcv,ylcv,zccv) = (0.4 , 1.2 , 1.0)
C*    V5 (xlcv,ylcv,zccv) = (0.6 , 1.2 , 0.8)
C*    V6 (xlcv,ylcv,zccv) = (0.6 , 1.2 , 1.0)
C*    V7 (xlcv,ylcv,zccv) = (0.6 , 1.8 , 0.8)
C*    V8 (xlcv,ylcv,zccv) = (0.6 , 1.8 , 1.0)
c*
c*
```

```
c*          P                                        P
c*       |------|----------------|----------------|--------|
c*
c*
c*                     ylcv
c*                      |-|
c*    -  +-------------------------------------------------+  60 um
c*    |  |      .                .                .        |
c*  P |  |      .                .                .        |
c*    |  |      .                .                .        |
c*    -  |.................................................|  35 um
c*    |  |      .                .                .        |
c*    |  |      .       +-+      .       +-+      .        |
c*    |  |      .       |7|      .       |8|      .        |
c*    |  |      .       +-+      .       +-+      .        |-xccv4
c*    |  |      .                .                .        |
c*    |  |.................................................| 32.5 um
c*    |  |      .                .                .        |
c*    |  |      .       +-+      .       +-+      .        |
c*    |  |      .       |5|      .       |6|      .        |
c*    |  |      .       +-+      .       +-+      .        |-xccv3
c*    |  |      .                .                .        |
c*    -  |.................................................| 30 um
c*    |  |      .                .                .        |
c*    |  |      .       +-+      .       +-+      .        |
c*    |  |      .       |3|      .       |4|      .        |
c*    |  |      .       +-+      .       +-+      .        |-xccv2
c*    |  |      .                .                .        |
c*    |  |.................................................| 27.5 um
c*    |  |      .                .                .        |
c*    |  |      .       +-+      .       +-+      .        |
c*    |  |      .       |1|      .       |2|      .        |
c*    |  |      .       +-+      .       +-+      .        |-xccv1
c*    |  |      .                .                .        |
c*    -  |.................................................| 25um
c*    |  |      .                .                .        |
c*  p |  |      .                .                .        |
c*    |  |      .                .                .        |
c*    -  +-------------------------------------------------+
c*              2       |        3       |        3        6
c*              5       y        0       y        5        0
c*              u       c        u       c        u        u
c*              m       c        m       c        m        m
c*                      v                v
c*                      1                2
c* ncv
c* xccv yccv zccv xlcv ylcv
8
26.05 27.10 -0.8 0.4 0.8
26.05 32.10 -1.0 0.4 0.8
28.55 26.90 -0.8 0.4 1.2
28.55 31.90 -1.0 0.4 1.2
30.95 26.90 -0.8 0.6 1.2
30.95 31.90 -1.0 0.6 1.2
33.45 26.60 -0.8 0.6 1.8
33.45 31.60 -1.0 0.6 1.8
c***********************************************************************
c***********************************************************************
c*  BLOCK #3: BEAM PROPERTY DESCRIPTION
c*
c*  PARAMETER LIST:
c*
c*  idprj   - id of projectile
c*  ekprj   - kinetic energy of projectile [MeV]
c*  irseed  - random number seed
c*  noint   - number of nuclear interactions within the target
c*  dirprj  - direction of projectile (+1 = from positive Z direction,
c*          - -1 = from negative z direction
c*----------------------------------------------------------------------
c* BEGIN PARAMETER LIST
c* Case study of a 30 MeV proton beam
c*
c* idprj ekprj irseed noint dirprj
p 63.3 123456 100000 -1
c***********************************************************************
```

Appendix F

List of Publications

F.1 As main contributor

- Røed,K. et al. “A Fault Injection Solution for an FPGA in Charge og Data Readout for a Large Tracking Detector”, 8^{th} European Workshop on Radiation Effects on Components and Systems, Sept. 2008, Jyväskylä, Finland

- Røed,K. et al. “Case Study of a Solution for Active Partial Reconfiguration of a Xilinx Virtex-II Pro”, Proceeding of FPGAworld Conference Proceedings 2006, Page(s) 30-34, ISSN 1404-3041 ISRN MDH-MRTC-204/2006-1-SE I

- Røed,K. et al. “Irradiation tests of the complete ALICE TPC Front-End Electronics chain”, Proceedings of the 11^{th} Workshop on electronics for LHC and future experiments, Sept. 2005, Heidelberg, Germany, Page(s): 165-169, ISBN 9290832622

F.2 As collaborator

- Alme, J. Røed,K. et al. “Radiation-Tolerant, SRAM-FPGA Based Trigger and Readout Electronics for the ALICE Experiment”, IEEE transactions on Nuclear Science Feb. 2008, Volume 56, Issue 1, Part 1, Page(s): 76-83, Digital Object Identifier: 10.1109/TNS.2007.910677

- Richter, M. Røed,K. et al. “The control system for the front-end electronics of the ALICE time projection chamber”, IEEE Transactions on Nuclear Science June 2006, Volume 53, Part 1, Page(s): 980-985, Digital Object Identifier: 10.1109/TNS.2006.874726

- Fehlker, D. Røed,K. et al. “Software environment for controlling and reconfiguration of Xilinx Virtex FPGAs”, Proceedings for the Topical Workshop for Particle Physics Sept. 2007, Prague, Czech Republic, URL: http://www.particle.cz/conferences/twepp07/

- Richter, M. Røed,K. et al. “A distributed, Heterogeneous Control System for the ALICE TPC electronics”, Proceedings of the International Conference on Parallel Processing Workshops June 2005, Page(s): 265-272, Digital Object Identifier 10,1109/ICPPW.2005.7

- Tröger, G. Røed,K. et al. “FPGAs - Reconfiguration for Radiation Tolerance”, GSI Scientific Report 2005, Instrumentation Methods 27, Page(s):288, URL: http://www.gsi.de/informationen/wti/library/scientificreport2005/index.html

- Tröger, G. Røed,K. et al. “FPGA Dynamix Reconfiguration in ALICE and beyond”, Proceedings for the 11th Workshop on electronics for LHC and future experiments, Sept. 2005, Heidelberg, Germany, Page(s): 119-122, ISBN 9290832622

- Gutierrez, C. G. Røed,K. et al. “The ALICE TPC Readout Control Unit”, Proceedings of the 2005 IEEE Nuclear Science Symposium and Medical Imaging Conference, Puerto Rico, USA, Oct. 2005, Page(s): 575-579, Volume 1, Digital Identifier 10.1109/NSSMIC.2005.1596317

Bibliography

[1] Xilinx, Inc. *Correcting Single-Event Upsets in Virtex-II Platform FPGA Configuration Memory*, xapp779 v1.1 edition, Feb. 2007.

[2] The ALICE Collaboration, K. Aamodt, and et al. ALICE Experiment at the CERN LHC. *IOP Journal of Instrumentation*, JINST 3 S08002, 2008.

[3] The ALICE Collaboration: F Carminati, P Foka, P Giubellino, A Morsch, G Paic, J-P Revol, K Safarík, Y Schutz, and U A Wiedemann (editors). ALICE: Physics Performance Report, Volume I. *Journal of Physics G: Nuclear and Particle Physics*, 30(11):1517–1763, 2004.

[4] C. G. Gutiérrez. *Readout and control system for the ALICE TPC electronics.* PhD thesis, Escuela Técnica Suprior de Ingenieros Industrriales y de Telecomunicación University of Cantabria, Spain, 2007.

[5] L. Musa, J. Baechler, N. Bialas, R. Bramm, R. Campagnolo, C. Engster, F. Formenti, U. Bonnes, R. Esteve Bosch, U. Frankenfeld, P. Glassel, C. Gonzales, H.-A. Gustafsson, A. Jimenez, A. Junique, J. Lien, V. Lindenstruth, B. Mota, P. Braun-Munzinger, H. Oeschler, L. Osterman, R. Renfordt, G. Ruschmann, D. Röhrich, H.-R. Schmidt, J. Stachel, A.-K. Soltveit, and K. Ullaland. The ALICE TPC front end electronics. *Nuclear Science Symposium Conference Record, 2003 IEEE*, 5:3647–3651 Vol.5, Oct. 2003.

[6] C. G. Gutiérrez, R. Campagnolo, A. Junique, L. Musa, J. Alme, J. Lien, B. Pommersche, M. Richter, K. Røed, D. Röhrich, K. Ullaland, and T. Alt. The ALICE TPC readout control unit. *Nuclear Science Symposium Conference Record, 2005 IEEE*, 1:575–579, Oct. 2005.

[7] M. Richter, J. Alme, T. Alt, S. Bablok, R. Campagnolo, U. Frankenfeld, C.G. Gutierrez, R. Keidel, Ch. Kofler, T. Krawutschke, D. Larsen, V. Lindenstruth, B. Mota, L. Musa, K. Røed, D. Röhrich, M.R. Stockmeier, and H. Tilsner. The control system for the front-end electronics of the ALICE time projection chamber. *Nuclear Science, IEEE Transactions on*, 53(3):980–985, June 2006.

[8] Actel Corporation. http://www.actel.com/products/solutions/ser/default.aspx.

[9] Actel Corporation. *APA750 and A54SX32A LANSCE Neutron Test Report*, white paper edition, Dec. 2003.

[10] Xilinx, Inc. *Correcting Single-Event Upsets Through Virtex Partial Configuration*, xapp216 v1.0 edition, June 2000.

[11] Johan Alme. *Firmware Development and Integration for ALICE TPC and PHOS Front-end Electroncis*. PhD thesis, Universitetet i Bergen, Bergen, Norway, 2008.

[12] Heather Quinn, Paul Graham, Keith Morgan, Jim Krone, Michael Caffrey, and Michael J. Wirthlin. An introduction to radiation-induced failure modes and related mitigation methods for xilinx sram fpgas. *Proceedings of the 2008 International Conference on Engineering of Reconfigurable Systems & Algorithms, ERSA 2008, Las Vegas, Nevada, USA, July 14-17, 2008*, pages 139–145, 2008.

[13] JEDEC STANDARD: Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices. Technical report, JEDEC Solid State Technology Association, Arlington, VA 22201-3834, Revision of JESD89, Aug. 2001.

[14] R.C. Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *Device and Materials Reliability, IEEE Transactions on*, 5(3):305–316, Sept. 2005.

[15] Xilinx, Inc. *Single-Event Upset Mitigation Selection Guide*, xapp987 v1.0 edition, March. 2008.

[16] H. Quinn, P. Graham, J. Krone, M. Caffrey, and S. Rezgui. Radiation-induced multi-bit upsets in sram-based fpgas. *Nuclear Science, IEEE Transactions on*, 52(6):2455–2461, Dec. 2005.

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

[18] G. Bruguier and J.-M. Palau. Single particle-induced latchup. *Nuclear Science, IEEE Transactions on*, 43(2):522–532, Apr 1996.

[19] C.M. Hsieh, P.C. Murley, and R.R. O'Brien. A field-funneling effect on the collection of alpha-particle-generated carriers in silicon devices. *Electron Device Letters, IEEE*, 2(4):103–105, Apr 1981.

[20] J.J Fabula. The NSEU response of static latch based FPGAs. Presented at the Military and Aerospace Programmable Logic Devices (MAPLD) conference., Apr 2003.

[21] H.H.K Tang. Nuclear physics of cosmic ray interaction with semiconductor materials: Particle-induced soft errors from a physicists perspective. *IBM Journal of Research and Development*, 40(1):2162–2167, 1996.

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

[23] J.F. Ziegler. Srim-2006.02. http://www.srim.org.

[24] Austin Leasea. Xilinx. Personal communication, 2007.

[25] G. Battistoni, S. Muraro, P.R. Sala, F. Cerutti, A. Ferrari, S. Roesler, A. Fassó, and J. Ranft. The FLUKA code: Description and benchmarking. *Procedings of the Hadronic Shower Simulation Workshop 2006, Fermilab 6-8 September 2006, M. Albrow, R.Raja eds., AIP Conference Proceddings 896, 31-49, (2007).*

[26] A Ferrari, Paola R Sala, A Fassó, and Johannes Ranft. *FLUKA: A multi-particle transport code.* CERN-2005-10(2005), INFN/TC_05/11, SLAC-R-773.

[27] B. B. Back et al. The significance of the fragmentation region in ultrarelativistic heavy ion collisions. *Phys. Rev. Lett.*, 91:052303, 2003.

[28] A. H. Wuosmaa. dnch/d[eta] distributions from phobos. *Nuclear Physics A*, 698(1-4):88 – 93, 2002.

[29] Georgios Karolos Tsiledakis. *Scale Dependence of Mean Transverse Momentum Fluctations at Top SPS Energy measured by the CERES experiment and studies of gas properties for the ALICE experiment.* PhD thesis, Technische Universitat Darmstadt, Darmstadt, 2006.

[30] Alife: A geometry editor and parser for fluka. Technical Report ALICE-INT-1998-29. CERN-ALICE-INT-1998-29, CERN, Geneva, 1998.

[31] AliRoot. http://aliceinfo.cern.ch/Offline/.

[32] Virtual Monte Carlo. http://root.cern.ch/drupal/content/how-use-virtual-monte-carlo.

[33] M.B. Chadwick, P. Obložinský, M. Herman, et al. ENDF/B-VII.0: Next generation evaluated nuclear data library for nuclear science and technology. *Nuclear Data Sheets*, 107(12):2931–3118, December 2006.

[34] C.J. Gelderloos, R.J. Peterson, M.E. Nelson, and J.F. Ziegler. Pion-induced soft upsets in 16 mbit dram chips. *Nuclear Science, IEEE Transactions on*, 44(6):2237–2242, Dec 1997.

[35] S. Duzellier, D. Falguere, M. Tverskoy, E. Ivanov, R. Dufayel, and M.-C. Calvet. Seu induced by pions in memories from different generations. *Nuclear Science, IEEE Transactions on*, 48(6):1960–1965, Dec 2001.

[36] Xilinx, Inc. *Virtex-II Pro and Virtex-II Pro X FPGA User Guide*, ug012 v4.2 edition, Nov. 2007.

[37] Xilinx, Inc. *Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet*, ds083 v4.7 edition, Nov. 5 2007.

[38] Xilinx, Inc. *Single-Event Upset Mitigation for Xilinx FPGA Block Memories*, xapp962 v1.1 edition, March. 2008.

[39] Actel Corporation. *Actel ProASIC^{plus} Flash Family FPGAs datasheet*, v5.7 edition, Sept. 2008.

[40] MXIC Macronix International Co.,Ltd. *MX29LV640B T/B 64M-BIT Single Voltage 3V Only Flash Memory Datasheet*, 2007.

[41] Altera. *Excalibur Devices, Hardware Reference Manual*, v3.1 edition, Nov. 2002.

[42] Dominik Fehlker. Development and commissioning of a software environment for controlling and re-configuration of Xilinx Virtex FPGAs. Diploma thesis, Faculty of Mathematics/Physics/Computer Science, Hochschule Mittweida (FH) University of Applied Sciences, Germany, 2007.

[43] Gerd Tröger. PhD thesis, University of Heidelberg. To be published.

[44] B. Povh, C. Rith, S. Scholz, and F. Zetsche. *Particles and nuclei: An Introduction to the Physical Concepts*, volume 2nd Edition. Springer, 1999.

[45] V.P. Eismont, A.V. Prokofiev, and A.N. Smirnov. THIN-FILM BREAKDOWN COUNTERS AND THEIR APPLICATIONS(REVIEW). *Radiation Measurements*, 25(1-4):1151–156, 1995.

[46] A.V. Prokofiev, A.N. Smirnov, and P-U Renberg. A Monitor of Intermediate-Energy Neutrons Based on Thin Film Breakdown Counters. Technical report, The Svedberg Laboratory and Department of Radiation Science, Uppsala, Sweden, 1999.

[47] Ketil, Røed. Irradiation tests of ALTERA SRAM-based FPGAs. Master's thesis, University of Bergen, Norway, 2004.

[48] Bjørn Halvor, Straume. Strålingstester og utvikling av bestrålingsprosedyrer for ALICE TPC-elektronikken. Master's thesis, University of Bergen, Norway, 2006.

[49] Fernanda Lima Karstensmidt. *SEE Mitigation Strategies for Digital Circuit Design Applicable to ASIC and FPGAs.* Nuclear and Space Radiation Effects Conference, Short Course Notebook, July 2007.

[50] Philippe Adell and Greg Allan. Assessing and Mitigating Radiation Effects in Xilinx FPGAs. Technical Report JPL publication 08-9 2/08, Jet Propulsion Laboratory California Institute of Technology, Pasedena, California, 2008.

[51] E. Fuller, P. Blain, M. Caffrey, and C. Carmichael. Radiation Test Results of the Virtex FPGA and ZBT SRAM for Space Based Reconfigurable Computing. *Proc. MAPLD*, Sept. 1999.

[52] J. Voas. Fault injection for the masses. *Computer*, 30(12):129–130, Dec 1997.

[53] T.A. Delong, B.W. Johnson, and III Profeta, J.A. A fault injection technique for vhdl behavioral-level models. *Design & Test of Computers, IEEE*, 13(4):24–33, Winter 1996.

[54] G.M. Swift, S. Rezgui, J. George, C. Carmichael, M. Napier, J. Maksymowicz, J. Moore, A. Lesea, R. Koga, and T.F. Wrobel. Dynamic testing of Xilinx Virtex-II field programmable gate array (FPGA) input/output blocks (IOBs). *Nuclear Science, IEEE Transactions on*, 51(6):3469–3474, Dec. 2004.

[55] S. Rezgui, G.M. Swift, and A. Lesea. Characterization of upset-induced degradation of error-mitigated high-speed i/o's using fault injection on sram based fpgas. *Nuclear Science, IEEE Transactions on*, 53(4):2076–2083, Aug. 2006.

[56] F. Lima Karstensmidt, L. Carro, and R. Reis. *Fault-Tolerance Techniques for SRAM-based FPGAs*, volume 32. Springer, 2006.

[57] M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, S. Pastore, and G.R. Sechi. Evaluation of Single Event Upset Mitigation Schemes for SRAM based FPGAs using the FLIPPER Fault Injection Platform. *Defect and Fault-Tolerance in VLSI Systems, 2007. DFT '07. 22nd IEEE International Symposium on*, pages 105–113, Sept. 2007.

[58] Matthias Richter. Private communication, 2008. University of Bergen, Norway.

[59] *Modeling Alpha and Neutron Induced Soft Errors in Static Random Access Memories*, 30 2007-June 1 2007.

[60] *Predicting neutron induced soft error rates: Evaluation of accelerated ground based test methods*, 27 2008-May 1 2008.

[61] J. Baggio, V. Ferlet-Cavrois, D. Lambert, P. Paillet, F. Wrobel, K. Hirose, H. Saito, and E.W. Blackmore. Neutron and proton-induced single event upsets in advanced commercial fully depleted soi srams. *Nuclear Science, IEEE Transactions on*, 52(6):2319–2325, Dec. 2005.

[62] M. Huhtinen and F. Faccio. Computational method to estimate single event upset rates in an accelerator environment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 450(1):155 – 172, 2000.

[63] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J. J. Gmez Cadenas, I. Gonzlez, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F. W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampn, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M. G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer,

S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J. P. Wellisch, T. Wenaus, D. C. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche. G4–a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250 – 303, 2003.

[64] J. Allison, K. Amako, J. Apostolakis, H. Araujo, P.A. Dubois, M. Asai, G. Barrand, R. Capra, S. Chauvie, R. Chytracek, G.A.P. Cirrone, G. Cooperman, G. Cosmo, G. Cuttone, G.G. Daquino, M. Donszelmann, M. Dressel, G. Folger, F. Foppiano, J. Generowicz, V. Grichine, S. Guatelli, P. Gumplinger, A. Heikkinen, I. Hrivnacova, A. Howard, S. Incerti, V. Ivanchenko, T. Johnson, F. Jones, T. Koi, R. Kokoulin, M. Kossov, H. Kurashige, V. Lara, S. Larsson, F. Lei, O. Link, F. Longo, M. Maire, A. Mantero, B. Mascialino, I. McLaren, P.M. Lorenzo, K. Minamimoto, K. Murakami, P. Nieminen, L. Pandola, S. Parlati, L. Peralta, J. Perl, A. Pfeiffer, M.G. Pia, A. Ribon, P. Rodrigues, G. Russo, S. Sadilov, G. Santin, T. Sasaki, D. Smith, N. Starkov, S. Tanaka, E. Tcherniaev, B. Tome, A. Trindade, P. Truscott, L. Urban, M. Verderi, A. Walkden, J.P. Wellisch, D.C. Williams, D. Wright, and H. Yoshida. Geant4 developments and applications. *Nuclear Science, IEEE Transactions on*, 53(1):270–278, Feb. 2006.

[65] H. H.K. Tang. SEMM-2: a new generation of single-event-effect modeling tools. *IBM J. Res. Dev.*, 52(3):233–244, 2008.

[66] A.S. Kobayashi, D.R. Ball, K.M. Warren, R.A. Reed, N. Haddad, M.H. Mendenhall, R.D. Schrimpf, and R.A. Weller. The effect of metallization layers on single event susceptibility. *Nuclear Science, IEEE Transactions on*, 52(6):2189–2193, Dec. 2005.

[67] H.H.K. Tang, C.E. Murray, G. Fiorenza, K.P. Rodbell, and M.S. Gordon. Importance of BEOL Modeling in Single Event Effect Analysis. *Nuclear Science, IEEE Transactions on*, 54(6):2162–2167, Dec. 2007.

[68] K. M. Warren, B. D. Sierawski, R. A. Weller, R. A. Reed, M. H. Mendenhall, J. A. Pellish, R. D. Schrimpf, L. W. Massengill, M. E. Porter, and J. D. Wilkinson. Predicting thermal neutron-induced soft errors in static memories using tcad and physics-based monte carlo simulation tools. *Electron Device Letters, IEEE*, 28(2):180–182, Feb. 2007.

[69] K.M. Warren, R.A. Weller, B.D. Sierawski, R.A. Reed, M.H. Mendenhall, R.D. Schrimpf, L.W. Massengill, M.E. Porter, J.D. Wilkinson, K.A. LaBel, and J.H. Adams. Application of radsafe to model the single event upset response of a 0.25 m cmos sram. *Nuclear Science, IEEE Transactions on*, 54(4):898–903, Aug. 2007.

[70] Xilinx, Inc. *Device Package User Guide*, ug112 v2.0 edition, May 31 2006.

[71] Xilinx, Inc. *Material Declaration Data Sheet FF672*, pk134 v1.3 edition, Jan. 17 2007.

[72] UMC Process Technology. http://www.umc.com/english/process/index.asp.

[73] UMC 90 Nanometer SoC Process Technology. http://www.umc.com/english/pdf/90nm_DM.pdf.

[74] UMC 0.13 Micron SoC Process Technology. http://www.umc.com/english/pdf/0.13DM.pdf.

[75] T. Heijmen. Analytical semi-empirical model for SER sensitivity of deep-submicron CMOS circuits. In *On-Line Testing Symposium, 2005. IOLTS 2005, 11th IEEE International*, pages 3–8, 2005.

[76] A. D. Tipton, J. A. Pellish, R. A. Reed, R. D. Schrimpf, R. A. Weller, M. H. Mendenhall, B. Sierawski, A. K. Sutton, R. M. Diestelhorst, G. Espinel, J. D. Cressler, P. W. Marshall, and G. Vizkelethy. Multiple-bit upset in 130 nm cmos technology. *Nuclear Science, IEEE Transactions on*, 53(6):3259–3264, Dec. 2006.

[77] FLUKA Team 2000 – 2008. The Fluka Online Manual. http://www.fluka.org/fluka.php?id=man_onl.

[78] Giovanni Filippi Jr., Ronald G.and Fiorenza, Xiao Hu Liu, Conal Eugene Murray, Gregory Allen Northrop, Thomas M. Shaw, Richard Andre Wachnik, and Mary Yvonne Lanzerotti Wisniewski. Method of extracting properties of back end of line (beol) chip architecture. http://www.freepatentsonline.com/7260810.html, August 2007. U.S. Patent No. 7260810.

[79] Giovanni Fiorenza, Conal Eugene Murray, Kenneth P. Rodbell, and Henry Tang. Methdod of determining stopping powers of design structures with respect to a traveling particle. http://www.freepatentsonline.com/7386817.html, June 2008. U.S. Patent No. 7386817.

[80] J. Keane, AJ KleinOsowski, E. Cannon, F. Gebara, and C.H. Kim. Method for Qcrit Measurement in Bulk CMOS Using a Switched Capacitor Circuit. In *NASA Symposium on VLSI Design*, 2007.