ULB UNIVERSITÉ LIBRE DE BRUXELLES

Bruface Brussels Faculty of Engineering

VUB VRIJE UNIVERSITEIT BRUSSEL

Experimental study on SEU mitigation in a commercial-off-the-shelf SRAM FPGA for the LHC Phase-2 upgrade

Charles Detemmerman

Master thesis submitted under the supervision of
Prof. Frédéric Robert

The co-supervision of
Prof. Gilles De Lentdecker

Academic year
2018-2019

In order to be awarded the Master's Degree in
Electrical Engineering

Abstract

Future upgrades to the Large Hadron Collider will lead to an increase in the levels of background radiation experienced by the detector electronics. Within the muon detectors of the Compact Muon Solenoid (CMS) experiment, a series of new components will be installed, including the GE2/1 and ME0 detectors. Inside their data acquisition chain is the subject of this thesis, the OptoHybrid. It is a board centred around an SRAM-based Field-Programmable Gate Array (FPGA), a type of programmable electronics which is highly susceptible to radiation-induced soft errors called single event upsets (SEU). These are non-damaging but they will modify the internal configuration of the FPGA, and cause errors in its output and internal signals. The FPGA model used in the OptoHybrid is the 28nm Xilinx Artix-7, which is not radiation-proof by nature. It must therefore be qualified for use in the CMS environment.

This thesis studies the possible systems which can be implemented to mitigate the effects of SEUs on the OptoHybrid. The selected system is a combination of configuration memory scrubbing using an Error Correction Code built into the Xilinx FPGA, and Triple Modular Redundancy. There are two main limitations to this system, the first is that the scrubber cannot repair memory frames which contain more than one erroneous bit. The second is that Triple Modular Redundancy breaks if more than one of the triplicated module is compromised. This can occur when SEUs accumulate, or when a single SEU flips multiple bits in the configuration memory. Because of these limitations, a reliability model based on Markov chains is proposed to predict the mean time to failure of the system.

The second contribution of this thesis is a study into the impact of radiation on the FPGA through accelerated radiation testing at the UCLouvain's cyclotron facility. The main sources of SEUs in the CMS environment are neutrons. The UCLouvain's proton line is used as a neutron analogue as both have a similar SEU cross-section. The experimental protocols for two tests are presented, the first provides proton SEU cross-section data, and statistical data on the proportion of multiple cell upsets (MCU) for a range of energies present in CMS, and for a range of particle angles of incidence. The second studies the efficiency of the mitigation system, specifically it tests various TMR implementation, and measures parameters for the Markov chain model on the experimental firmware, which does not use modules from the OptoHybrid firmware.

Previous work on the OptoHybrid of the GE1/1 detector, which used a 40nm Xilinx Virtex-6 FPGA, found a linear relation between proton cross-section and energy in the 30 to 62 MeV range, and a drop at 20 MeV. In the Artix-7 test, the proton cross-section is found to be lower but uniform in the 28.9 to 62 MeV range. Furthermore, the MCU proportion is found to rise both with energy and angle of incidence, as does the subset proportion of non-repairable MCUs.

The mitigation system tests found that increasing the logical granularity of TMR yields significant improvement in error rates, while simulations with the Markov chain model indicate that the rate of non-repairable errors has a major impact on reliability. These results are only valid for the experimental firmware, therefore methods are proposed for measuring the Markov model parameters on the OptoHybrid firmware in the future.

Keywords: CMS, SRAM, FPGA, SEU mitigation, SEU cross-section, reliability modelling.

Student: Charles Detemmerman
Degree: Master of Science in Electrical Engineering
Academic year: 2018-2019
Title: Experimental study on SEU mitigation in a commercial-off-the-shelf SRAM FPGA for the LHC Phase-2 upgrade

Acknowledgements

I would like to extend my thanks to my two supervisors, Prof. Frédéric Robert and Prof. Gilles De Lentdecker for their help and guidance throughout this master thesis. I must also express my deepest gratitude to Laurent Pétré for his involvement. The time and energy he has spent helping me during the course of this year has been truly invaluable.

To all my friends at the ULB and the VUB, I will say that you have made my time at university one of the most joyous experience of my life, it would not have been the same without you.

To my parents, thank you for your continuous support, love, encouragement, and everything else. To my brothers, you are certainly not the worst people I have ever met.

I thank you all.

It is an experience common to all men
to find that, on any special occasion,
such as the production of a magical effect for the first time in public,
everything that can go wrong will go wrong.

Whether we must attribute this to the malignity of matter
or to the total depravity of inanimate things,
whether the exciting cause is hurry, worry, or what not,
the fact remains.

Nevil Maskelyne

Contents

Abstract iii

Acknowledgements v

Introduction 1

1 Context 3
- 1.1 Large Hadron Collider 3
- 1.2 Compact Muon Solenoid 3
  - 1.2.1 Overview of the Muon detectors 6
  - 1.2.2 Trigger system 6
- 1.3 GEM detectors 8
  - 1.3.1 Data acquisition system 8

2 Radiation effects in FPGAs 15
- 2.1 Basics of FPGAs 15
- 2.2 Radiation characteristics 15
- 2.3 Accumulated effects 18
  - 2.3.1 Displacement effect 18
  - 2.3.2 Charge accumulation effect 18
  - 2.3.3 Total Ionizing Dose 18
- 2.4 Single Event Effects 18
  - 2.4.1 Origin 18
  - 2.4.2 Categorization 19
- 2.5 CMS Environment 22
- 2.6 Existing studies 24
  - 2.6.1 Scaling trends 27

3 SEU mitigation techniques 29
- 3.1 Radiation hardening solutions 29
- 3.2 Scrubbing architectures 31
  - 3.2.1 Blind scrubbing 31
  - 3.2.2 Readback scrubbing 31
  - 3.2.3 Hybrid schemes 32
  - 3.2.4 External solutions 32
  - 3.2.5 Xilinx Soft Error Mitigation solution 32
- 3.3 Modular redundancy 34
  - 3.3.1 N-modular redundancy 34
  - 3.3.2 Granularity 35
  - 3.3.3 Signal triplication and TMR logical areas 35
  - 3.3.4 Sequential logic 36
  - 3.3.5 Implementation 36
- 3.4 Designing a mitigation solution 39
- 3.5 System reliability 39
  - 3.5.1 Limitations of the mitigation solution 39
  - 3.5.2 Notions of reliability theory for electronic systems 40
  - 3.5.3 Simple reliability model 41

3.5.4 Markov chain model 41

4 Experiments 45

4.1 Beam testing campaign 45

4.2 SEU cross-section test 45

4.2.1 Motivation 45

4.2.2 Principle 46

4.2.3 Experimental architecture 46

4.2.4 Parameters 48

4.3 Mitigation system test 48

4.3.1 Motivation 48

4.3.2 Principle 49

4.3.3 Experimental architecture 49

4.3.4 Parameters 51

5 Data analysis 53

5.1 SEU data 53

5.1.1 MCU extraction 53

5.1.2 SEU cross-sections and MCU statistical data 55

5.1.3 SEU location probability density function 55

5.1.4 Proposed error injection scheme 55

5.1.5 Conclusions 55

5.2 TMR data 58

5.2.1 Data culling 58

5.2.2 Results by sector type 58

5.2.3 Markov chain model simulations 59

5.2.4 Conclusions 62

5.3 Currents and voltages 63

5.4 TID 63

Conclusion 65

Glossary 67

Bibliography 69

A Device specifications 73

B Experimental dataset 75

C TMR Module - VHDL Example 79

D Markov chain model solution 81

Introduction

The Large Hadron Collider (LHC), developed in Switzerland by the European Organization for Nuclear Research (CERN) is, at almost 27 km in circumference, the most powerful particle collider ever built, and the largest machine in the world. A massive feat of physics and engineering, it currently involves thousands of researchers, technicians, engineers, and administrators across Europe and the world. In 2012, after four years in operation, the LHC has provided conclusive evidence for the existence of the Brout-Englert-Higgs boson, earning a Nobel Prize to François Englert and Peter Higgs. The first proton-proton collisions were achieved in 2010 at a total collisional energy of 7 TeV, steadily increased up to the present world record at 13 TeV.

Upgrades are now underway for Phase-2 of the LHC roadmap, during which the number of particles interactions in the detectors will increase by a factor of 10. Among the LHC detectors is the Compact Muon Solenoid (CMS) experiment in which, as part of the upgrades, new components will be installed, including the GE2/1 et ME0 detectors (GEM detectors) which are the subject of this thesis.

Among the electronic systems of the GEM detectors is the OptoHybrid board, a part of the data acquisition chain based around a Field-Programmable Gate Array (FPGA), the Xilinx Artix-7. FPGAs are programmable electronic devices, which can implement any sort of circuit based on a design they keep in memory (the firmware). When working in an environment of intensive ionizing radiation, FPGAs will tend to malfunction, and could even be damaged if the absorbed dose is high enough. This is due to the interactions the incident particles have with the semiconductor material of the electronic system. FPGAs which store their configuration in Static Random-Access Memory (SRAM) cells, such as the Artix-7, are particularly vulnerable as these cells can be upset, and the programmed circuit can therefore be modified. In the context of CMS the main consequence of radiation will be Single Event Effects (SEE), a collection of disruptive effects created by the interaction of a single particle with the FPGA, but which are most often reversible. These SEEs will be caused mainly by neutron radiation; insuring that the FPGA is protected against them is essential to the viability of the OptoHybrid board in CMS operation.

Simulations have shown that the background neutron radiation in the CMS cavern will increase significantly during Phase-2. This will make the electronic systems, and in particular the FPGAs, more prone to errors. The main contribution of this thesis is first to provide experimental data on the interaction cross-section of neutron particles with the Artix-7 FPGA, as well as statistical data on the rate of Multiple Cell Upsets (MCU). MCUs are the consequence of an SEE creating multiple bit flips in SRAM cells, which is particularly problematic in the context of SEE mitigation for reasons which will be presented in Chapter 3. Previous work on this subject has been done for the GE1/1 detector [29] [28], which also contained an OptoHybrid board, but based around an older FPGA model, the Xilinx Virtex-6. As the Artix-7 is based around a new 28nm technology node, and considering that the ME0 detector will experience a neutron flux ten times higher than the GE1/1, the interaction of the Artix-7 with the radiation environment of CMS must also be assessed.

The second contribution of this thesis will be an investigation into the efficiency of the mitigation solution currently being considered for the OH FPGA. Because the mitigation solution is limited in capability, a model is proposed to predict its reliability over time. Exact predictions are not given for the reliability of the OptoHybrid as its firmware could not be used for the experiment, instead data gathered from an experimental firmware is applied to the reliability model in order to give a sense of where the bottlenecks are. Indications are also given on how to do the same for the OptoHybrid firmware.

This master thesis investigates the causes and consequences of radiation effects in FPGAs (Chapter 2), and proposes potential mitigation solutions (Chapter 3). An experimental protocol is designed for testing the interaction between the FPGA and a proton beam, and for testing the efficiency of the mitigation system (Chapter 4). The results of irradiation tests carried out during the thesis at the UCLouvain's cyclotron facility are then analysed and presented (Chapter 5).

Chapter 1

Context

1.1 Large Hadron Collider

The Large Hadron Collider (LHC) at CERN is the world's largest and most powerful particle accelerator. It consists of a 27-kilometre ring of superconducting electromagnets, along which are spaced several accelerating structures to boost the energy of the particles. Inside the vacuum of the accelerator, high energy particle beams travel at close to the speed of light in both directions around the ring, and collide at four particle detectors: ATLAS, CMS, ALICE and LHCb (Figure 1.1) [49]. The Compact Muon Solenoid (CMS) is of particular interest, as this work is directly intended to help its upgrade to the future high-luminosity phase.

Luminosity and Phase-2

In accelerator physics, luminosity L is defined as the proportionality factor between the number of events (collisions) per second $\frac{dN_c}{dt}$ and the interaction cross section σ_p (the area transverse to their motion within which the particles must meet for a collision to occur, the unit is cm^2) [20].

$$\frac{dN_c}{dt} = L \cdot \sigma_p$$

The unit of luminosity is therefore $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. A related quantity is the *integrated luminosity*, which is the integral of the luminosity with respect to time, usually expressed in *inverse femtobarn* ($fb^{-1} = 10^{39}\,\mathrm{cm}^{-2}$).

$$L_{int} = \int L \cdot dt$$

During the lifetime of the LHC, the luminosity of the accelerator will steadily increase from a current integrated luminosity of $300\,\mathrm{fb}^{-1}$ to a target of $3000\,\mathrm{fb}^{-1}$ in the upcoming upgrade, the High-Luminosity LHC (HL-LHC). The activity of the LHC is separated into periods of data acquisition and periods of *long shutdowns* (LS), during which repairs and upgrades are undertook. As can be seen in Figure 1.2, we are currently in the second Long Shutdown (LS2), in which installations for the HL-LHC have already started. The current "Phase-1" of LHC operation will continue until the year 2023, when the LS3 will finalize the HL-LHC upgrade. If all goes well, this "Phase-2" of LHC operation will start in 2026. This increase in luminosity will mean increased stress on the system at several levels. Increased data rates on the one hand will require improvements in the data acquisition chain but, more significantly for this thesis, it will also mean higher background radiation levels, which will increase the risk of failure in electronics.

1.2 Compact Muon Solenoid

The CMS is a general purpose detector, built around a huge solenoid magnet generating a field of 4 tesla, almost 100,000 times the magnetic field of the Earth. It has a cylindrical shape, 28.7 m long and 15 m in diameter [28]. The detector is separated into two areas: the *barrel*, forming the cylinder, and two *endcap* regions, closing it off at both ends (Figure 1.3). The detector contains subsystems designed to measure the energy and momentum of photons, electrons, muons and other products of the collisions.

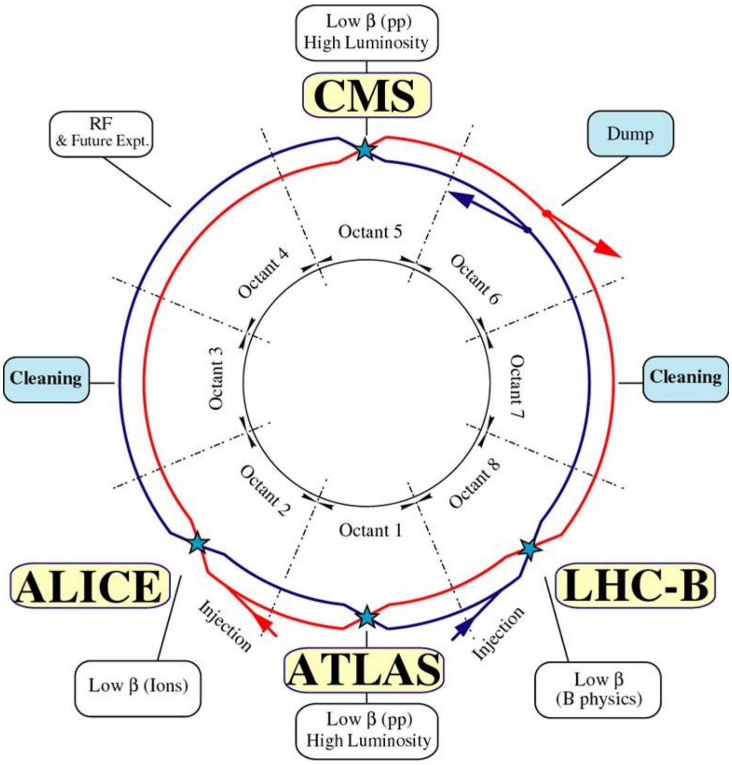


Figure 1.1: Schematic of the LHC [6]

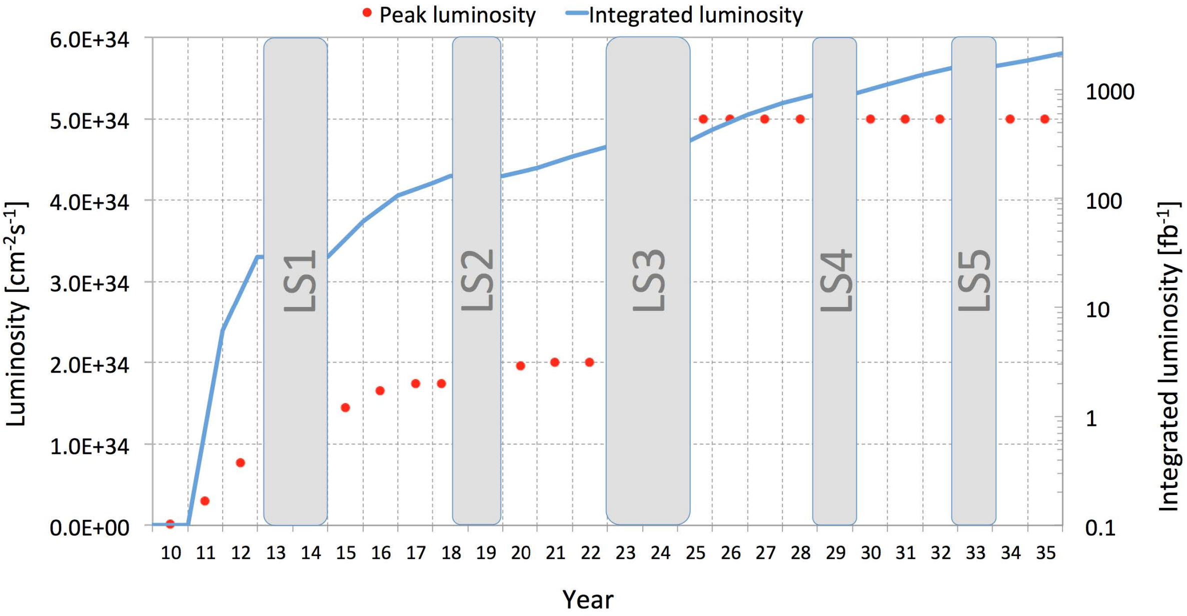


Figure 1.2: LHC luminosity evolution [17]

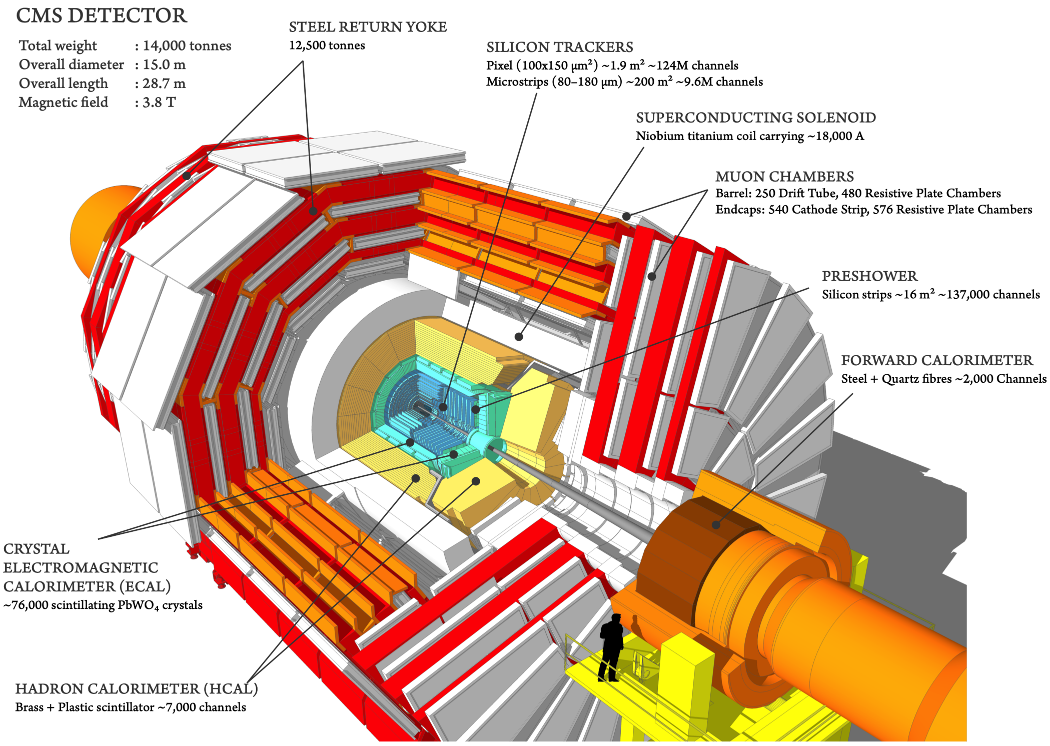

CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm²) ~1.9 m² ~124M channels
Microstrips (80–180 μm) ~200 m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000 A
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 540 Cathode Strip, 576 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16 m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating PbWO₄ crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

Figure 1.3: Cutaway diagram of the CMS detector [41]

At the center of the CMS, particles will collide, generating a shower of secondary particles, which will again generate more particles through interactions with material in the detector. Some of these particles' trajectories will go through the electronic equipment of the detector, and potentially disrupt their normal behaviour. As can be seen on Figure 1.3, there are several layers forming the detector, which will be briefly detailed in the following section, followed by a discussion of the CMS triggering system.

CMS layers

Figure 1.4 provides a more detailed view of the CMS barrel, on which one can identify six regions: the interaction point, the tracker, the Electromagnetic Calorimeter (ECAL), the Hadronic Calorimeter (HCAL), the superconducting magnet and the muon detector, interleaved with a steel return yoke used to guide the magnetic field. The arrangement is similar for the endcap regions.

The **silicon tracker** is used to determine the collision point and the momentum of charged particles. It can reconstruct the paths of high-energy muons, electrons and hadrons, as well as see tracks coming from the decay of very short-lived particles such as beauty or “b quarks”. The pervasive magnetic field will bend the path of charged particles, enabling momentum detection. The pixels, at the very core of the detector and dealing with the highest intensity of particles, and the silicon microstrip detectors that surround it are made entirely of silicon. As particles travel through the tracker the pixels and microstrips produce tiny electric signals that are amplified and detected [11].

The **ECAL** consists of tungstate crystals, a highly transparent material producing light when electrons or photons pass through it. Photodetectors are placed on the back in order to convert and record the produced light by converting it to electrical signals. A calorimeter's purpose is to detect the energy of the incident particles, which it does by allowing these particles to interact within the calorimeter and deposit their energy. The amount of interaction with the detector is measured and the energy is derived from it.

Another type of calorimeter in CMS is the **HCAL**, which is used to detect hadrons (particles made of quark and gluons), as the name suggests. It also provides indirect measurement of the presence of non-interacting, uncharged particles such as neutrinos. It is made up of alternating layers of "absorber" material, reducing the energy of the particle, and "scintillator" material which give short bursts of light when a particle passes through. By extrapolating the path of the particle from the scintillator data, one can compute the total energy exchanged between the particle and the absorbing material.

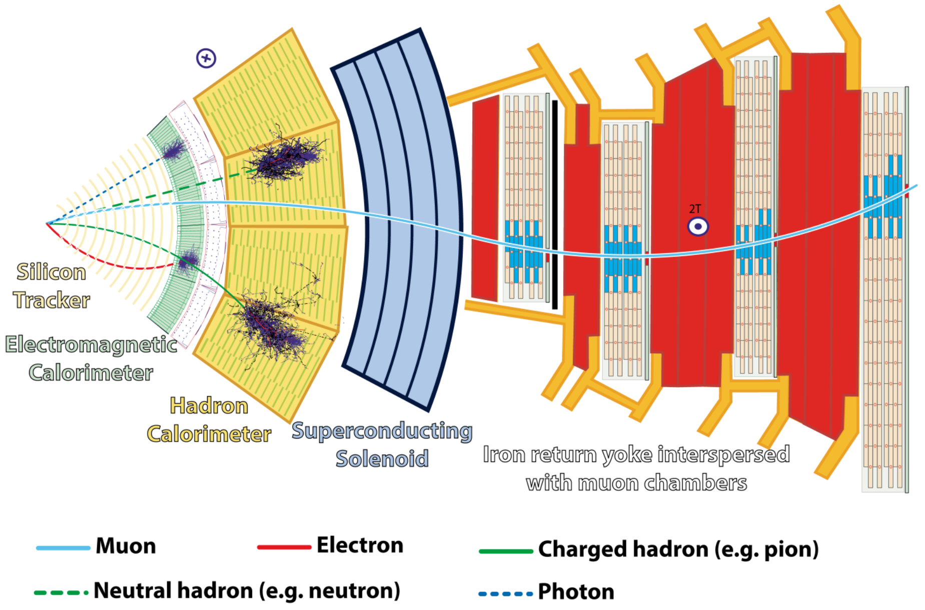

Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with muon chambers
2T
Muon
Electron
Charged hadron (e.g. pion)
Neutral hadron (e.g. neutron)
Photon

Figure 1.4: Cross-sectional slice of the CMS detector [4]

Muons and neutrinos are the only particles remaining after the solenoid due to their high penetration range (low interaction cross-section). Therefore the **muon detector** is placed there. [28]

1.2.1 Overview of the Muon detectors

CMS coordinate system

The schematic of Figure 1.5, which will be presented in the next section, is displayed in the CMS coordinate system. The origin is the nominal interaction point of the beams, the y-axis points upwards, the x-axis towards the center of the LHC loop, and the z-axis along the beam direction. The polar coordinates (r, θ) are defined in the x-y plane, and the **pseudorapidity** is defined as follows:

$$\eta = -\ln(\tan(\frac{\theta}{2})) \tag{1.1}$$

Detectors

The muon tracker is of particular interest as the target FGPA board of this thesis (the OptoHybrid) is part of the muon tracker data acquisition system. It is made up of three different detector types: Drift Tubes (DTs), Cathode Strip Chambers (CSCs), and Resistive Plate Chambers (RPCs), to which a new type will be added during Phase-2: the Gas Electron Multiplier (GEM). DTs and CSCs are placed in the barrel and the endcaps respectively, and RPCs are present in both regions. One quadrant of CMS is shown on Figure 1.5, highlighting the placement of the various technologies inside the detector. Three systems are to note on this diagram: GE1/1, GE2/1 and ME0. These are the three GEM-type detectors which will be added for Phase-2 [10].

The GE1/1 system is presented extensively in the PhD Thesis of Thomas Lenzi [29] and is currently being installed. We will focus on the GE2/1 and ME0 detectors, which are similar in design but use an updated version of the OptoHybrid board. The GE1/1 and GE2/1 systems cover the forward pseudorapidity range $1.6 < \eta < 2.4$ and are mainly designed to improve the trigger system in this region (see Section 1.2.2). The ME0 detector will increase the tracker coverage up to $\eta = 2.8$. The radiation flux increases with pseudorapidity (see Section 2.5), which means the ME0 will see more radiation than the GE1/1 and GE2/1 systems.

1.2.2 Trigger system

To maximize the number of interesting events for the LHC physics program, a very large amount of collisions per second is required. A single beam crossing produces around 1MB of data every 25ns

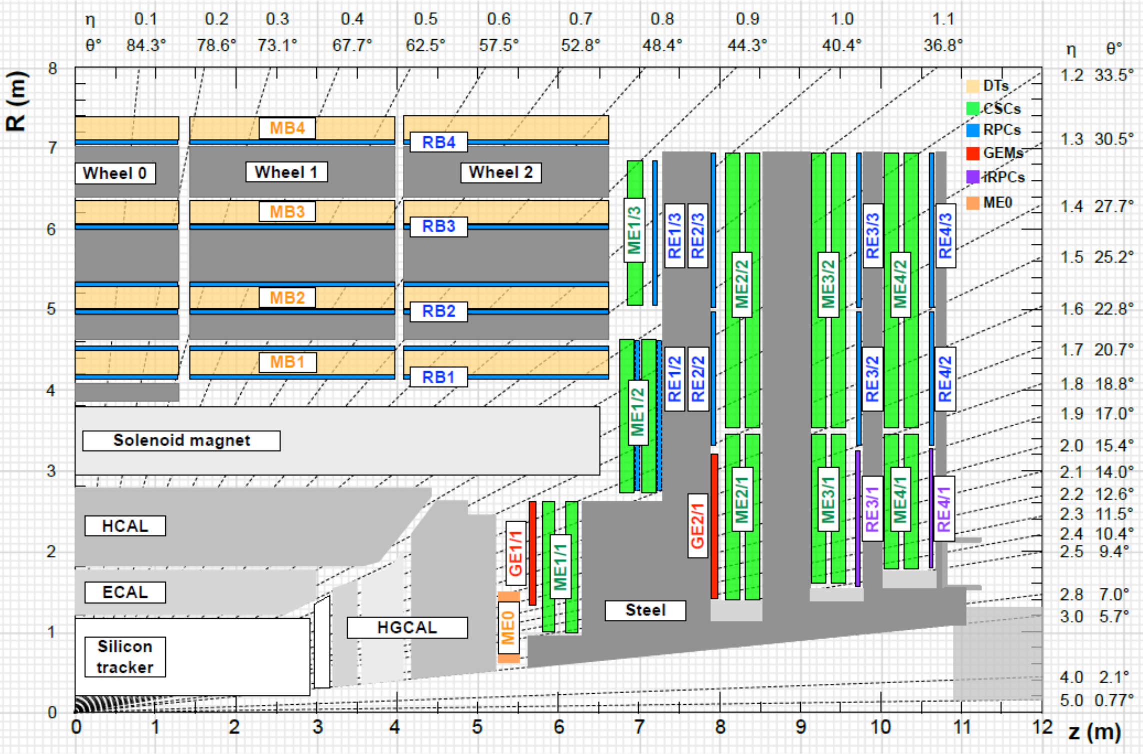


Figure 1.5: Cross-section of a CMS quadrant [10]

(40MHz crossing rate), which would result in 40 terabytes per second if all the sensor data was captured. Obviously this is not workable, therefore a *triggering system* is implemented. Whenever a trigger is generated, the full detector data is captured and stored. This allows a reduction in the data acquisition rate from 40MHz to 100kHz (500 kHz in Phase-2) (Level-1 trigger) and then to 1.5kHz (High-Level Trigger) [29].

The L1 trigger is made of custom electronics designed to quickly identify features of interest in the data, in order to discriminate between events. It must take a decision under 3.2 µs (12.5 µs in Phase-2) on whether to pass the data to the next trigger stage or not. At this stage, only the information from the HCAL, ECAL and muon chambers are used.

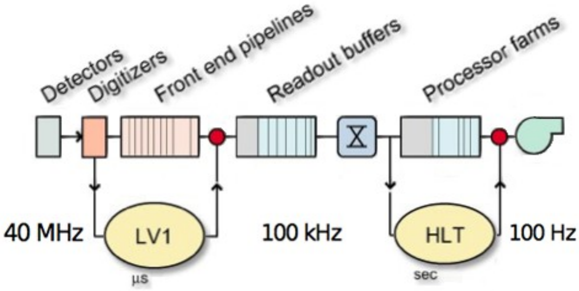


Figure 1.6: Block diagram of the trigger system pipeline [53]

If a collision passed the L1 trigger, all of the buffered data is sent through optical fibers to a computer farm, which will perform the High-Level Trigger (HLT). The lower event rate at this point allows time for a much more detailed analysis in software (mainly C++).

Global L1 Trigger

Only the information from the HCAL, ECAL and muon chambers are used at the L1 Trigger stage. Under the current system this means data from the CSCs, DTs and RPCs (see Figure 1.7). With the Phase-2 upgrade, hits from the new GEM chambers will be combined at an early stage with data from

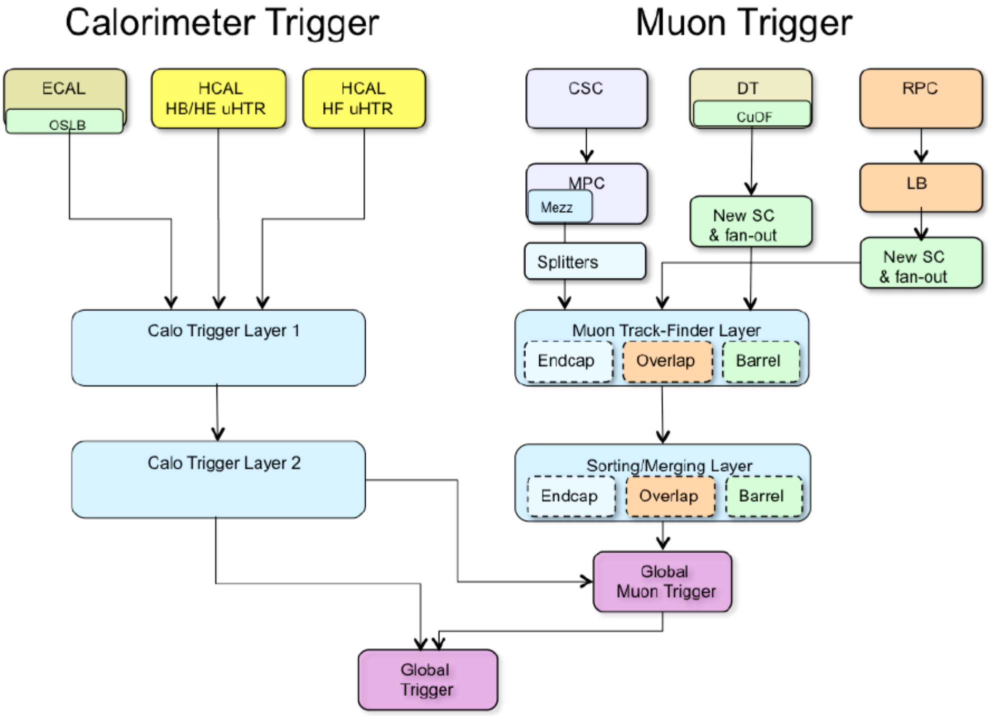


Figure 1.7: Global L1 trigger system [3]

the CSCs to improve efficiency (see Figure 1.8) [10].

The energy deposition data from the HCAL and ECAL is sent to the Calo Trigger Layer 1 which performs data formatting and concentration, and forwards events to the Calo Trigger Layer 2. Hits from the different muon detector chambers (barrel, endcap and overlap) are combined in the Muon Track-Finder layer, which extrapolates tracks from the data using pattern recognition algorithms. From then on, the sorting/merging layer selects the best event candidates, which are sent forward to the Global Muon Trigger (GMT). Further processing is done with the data from the calorimeters in the GMT and in the Global Trigger.

1.3 GEM detectors

The main purpose for installing the GEM detectors is the improvement of the triggering system of CMS. Without it, the coming increases in instantaneous luminosity would degrade the performance of the system due to the rate limitation of the ME1/1 stations (see Figure 1.5).

A GEM foil is a 50 µm-thick polymer foil, covered with 5 µm-thin copper sheets on both sides, and perforated at a high density by microscopic holes. When a high voltage is applied between the two layers of copper (typically on the order of 300V), field densities inside the holes reach approximately 80 kV cm^{-1} (see Figure 1.9 on the left). When a charged particle passes through the detector, the electrons released from the ionized gas in the chamber are accelerated towards the holes of the foil. These electrons will reach high kinetic energies inside the holes, and then ionize the medium themselves, producing an avalanche of secondary electrons, hence the name Gas Electron Multiplier. To achieve high gains, there are two solutions: increase the voltages on the foil or use multiple foils. When working with very high electric fields, discharges which can damage the detector are possible, the second option is therefore preferred. The choice was made to use three GEM foils, which is why they are called triple-GEM detectors (see Figure 1.9 on the right). [29]

1.3.1 Data acquisition system

A generic block diagram of a triple-GEM chamber is shown on Figure 1.10. The diagram illustrates the main components for the readout of a single GEM chamber. A specific diagram for GE2/1 is shown on Figure 1.11, note the direct connections to the CSC used for the trigger system (see also Figure 1.8).

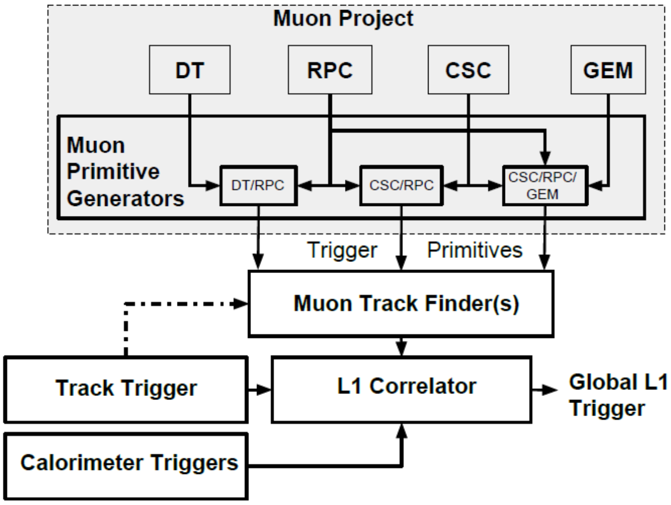


Figure 1.8: Upgraded L1 muon trigger [10]

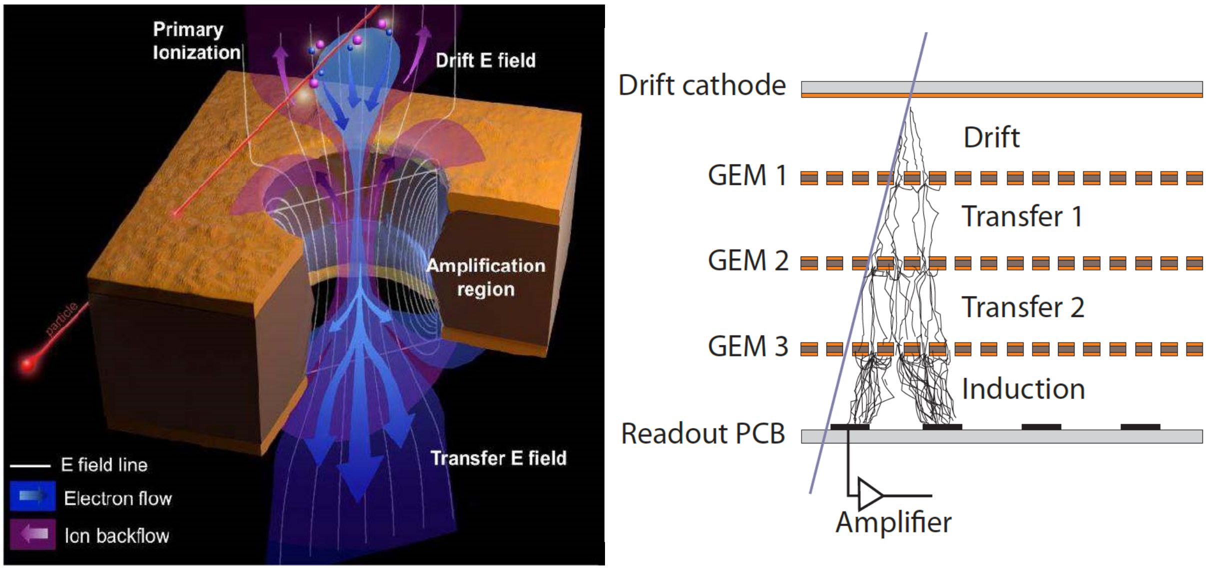


Figure 1.9: Schematic of the field lines inside a GEM foil (left). Principle of operation of a triple-GEM chamber (right) [12]

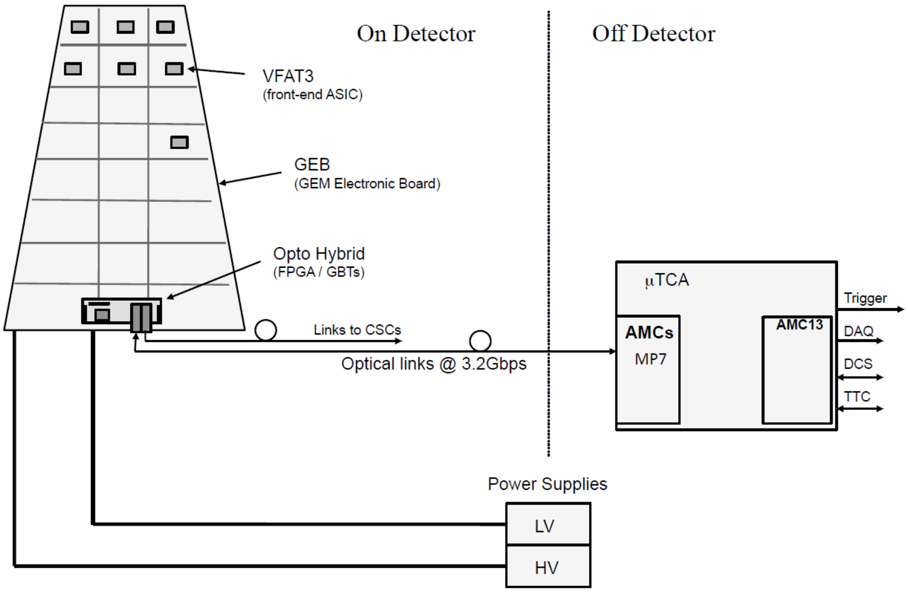


Figure 1.10: Generic GEM readout system [12]

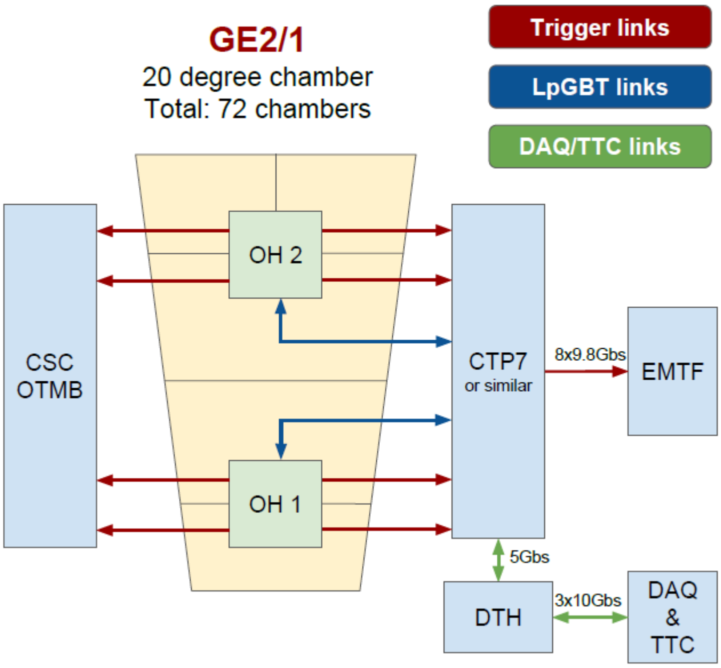


Figure 1.11: GE2/1 readout system [10]

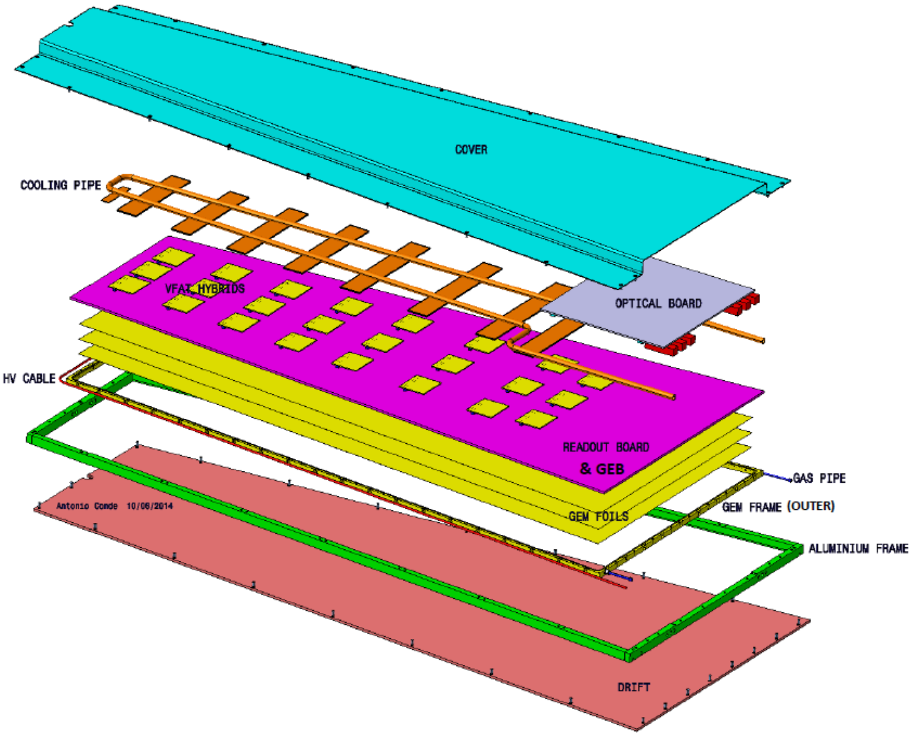


Figure 1.12: Exploded view of a single triple-GEM module [10]

GEM Electronics Board

The GEM Electronics Board (GEB) is the host of the data acquisition electronics in the GEM (see Figure 1.12). It routes signals from the VFAT3 Application-Specific Integrated Circuit (ASIC) to the OptoHybrid (OH), provides power to the chips and electric shielding to the detector. The following sections will serve as a brief tour of the important components.

VFAT3

The VFAT3 is the front-end data acquisition chip, optimized for gaseous detectors. Its function is to digitize the analog signals coming from the GEM foils and provide fast trigger and tracking data. The trigger data is sent over a fixed latency path to the L1 trigger system, while the tracking data holding the full granularity information of the events is stored in two Static Random-Access Memorys (SRAMs). If the L1 trigger algorithms decides to accept the event, the full tracking data will be sent over a variable latency path. It is comprised of an analog front-end, which does the signal shaping and digitization, and a digital back-end for control and readout (see Figure 1.13) [29].

OptoHybrid

The OptoHybrid board is the interface between the VFAT3 and the off-detector system (see Figure 1.14). It is centred around an Field-Programmable Gate Array (FPGA) (a Xilinx Artix-7 in the GE2/1 OH, see Annex A.1 for the details) and Integrated Circuits (ICs) dedicated to the operation of the optical link. The FPGA applies zero suppression algorithms to the trigger data, formats it, and sends it forward to the CSCs and the GEM trigger system separately over two optical links [29]. A new version of the OH board has been developed for the GE2/1 and ME0, switching from a Xilinx Virtex-6 FPGA to an Artix-7 model. Because this FPGA, like the previous one, is not made to be resistant to radiation (radiation-hardened), studies have to be made to predict its behaviour in a high-radiation environment, and some systems must be developed to counteract the radiation effects and insure proper behaviour of the electronics. Previous studies have been made on the Virtex-6 [28] [29], however as the technology node for the Artix-7 as been shrunk to 28nm (40nm for the Virtex-6), the radiation performance must be re-evaluated.

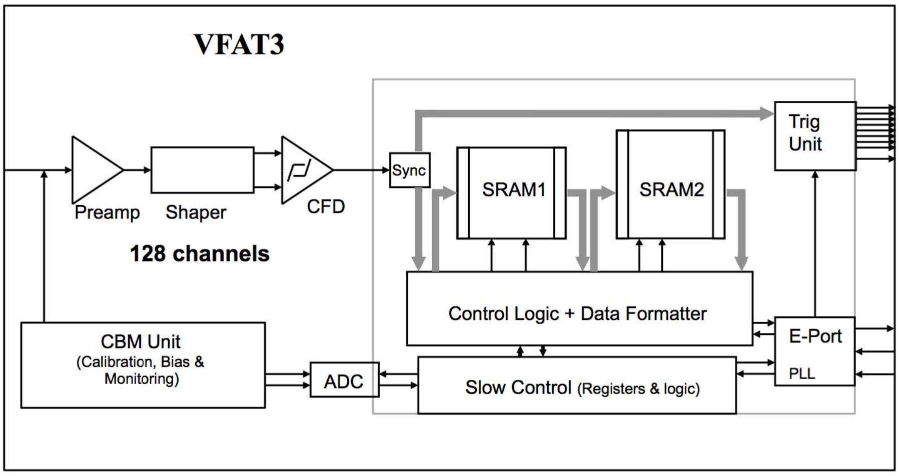

VFAT3
Preamp
Shaper
CFD
128 channels
Sync
SRAM1
SRAM2
Trig Unit
Control Logic + Data Formatter
E-Port
PLL
CBM Unit
(Calibration, Bias & Monitoring)
ADC
Slow Control (Registers & logic)

Figure 1.13: Block diagram of the VFAT3 ASIC [12]

GBT

The GigaBit Transceiver (GBT) architecture is part of the "Radiation Hard Optical Link Project" developed by CERN, which aims at providing radiation hard tools for optical communication. The technology provides bidirectional 4.8 Gbps (Gigabits per second) serial communication, and is designed to connect on-detector electronics to the off-detector systems. It is used for data acquisition, timing, trigger and control. This architecture is implemented in two system: the GBTx rad-hard ASIC, which is on the OH board, and the GBT-FPGA, which is a module that can be implemented inside an FPGA. Figure 1.15 illustrates the optical link setup with these systems at both ends.

Back-end electronics

The function of the back-end electronics is to: handle slow control requests for the detector subsystems, interpret Timing, Trigger and Control (TTC) signals, and readout the trigger and tracking data from the OHs. This is handled by the CTP7 (see Figure 1.11), a board in the Advanced Mezzanine Card format, equipped with a large Xilinx Virtex-7 FPGA for computational power and a Xilinx Zynq System on Chip (SoC) running a Linux operating system for monitoring [48].

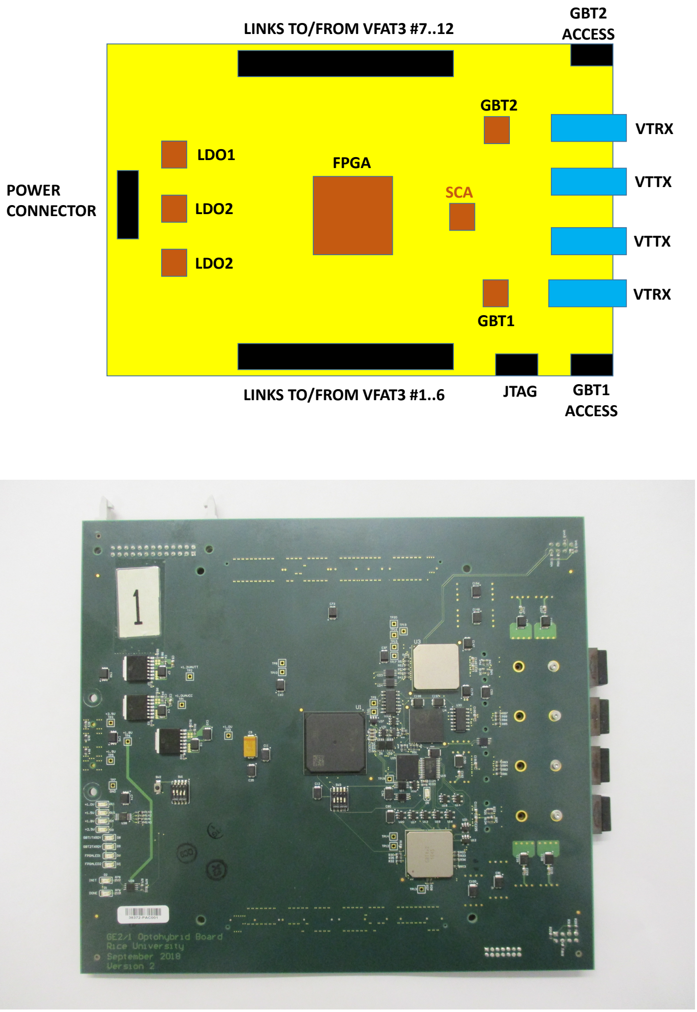


Figure 1.14: Diagram and top view picture of the GE2/1 OptoHybrid board (Mikhail Matveev)

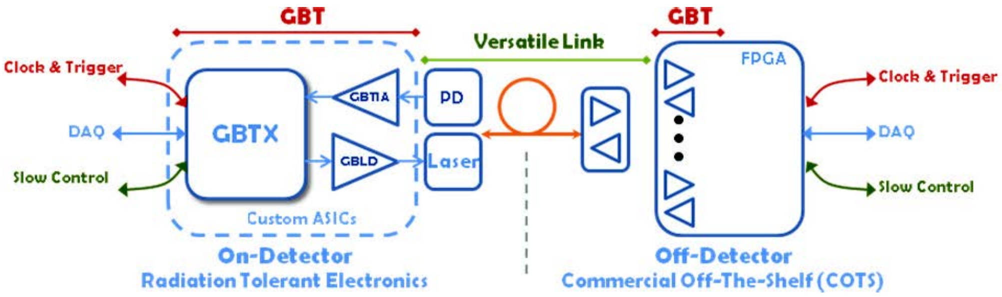


Figure 1.15: Architecture of a radiation hard optical link setup integrating the GBTx and GBT-FPGA [58]

Chapter 2

Radiation effects in FPGAs

In FPGAs, radiation effects can be classified in two broad categories: accumulated effects (i.e. degradation mechanisms), and Single Event Effects (SEEs). The former is caused by the accumulation of particle interactions with the semiconductor, degrading its performance over time, up to a total device failure. The latter encompasses all instantaneous effects caused by a single particle, which will usually not permanently damage the device (with some exceptions), but which can cause errors in various ways. A non-exhaustive summary of single event effects is shown on Figure 2.1, the various categories will be presented in Section 2.4.2.

2.1 Basics of FPGAs

A generic FPGA architecture is shown on Figure 2.2, it consists of logic blocks (which we call here configurable logic block or **CLB**, according to the Xilinx terminology) interconnected through programmable **routing matrices** (also called switch matrices). The CLBs contain look-up tables (LUT) implementing combinatorial logic functions (truth tables), storage elements (flip-flops) for sequential logic and pipelines, multiplexers and carry logic [1]. A 7-series Xilinx FPGA follows this general architecture, and has a total of 8 LUTs per CLB (divided into two *logic slices* containing 4 LUTs each), see Figure 2.3.

The functionality of an FPGA is described in Hardware Description Languages (HDL) such as VHDL or Verilog. In a nutshell, the HDL functional description will be *synthesized* into digital logic and *implemented* into the FPGA by configuring the memory of the LUTs, connecting them together and to flip-flops within the CLBs through the various multiplexers, and between CLBs through the routing matrices.

The result of this implementation is the *firmware*, a complete description of how the FPGA should be configured. This firmware is then translated into a *bitstream*, a sequence of bits which can be loaded into the FPGA. Loading the bitstream can be done through various interfaces, JTAG being the most commonly used during development through a USB-JTAG adapter connected to a computer.

As FPGA are programmable electronic devices, their configuration is stored in what is called the **Configuration Memory** (or CRAM). This refers to all the distributed elements of memory which implement the design (e.g. LUT tables, routing matrix configuration, etc), and its content is (part of) the content of the bitstream. In a 7-series Xilinx FPGA, the bitstream (an by extension the CRAM) is arranged in *frames* of 101 *words* (32 bits per word). The most common type of FPGA is SRAM-based (i.e. the design is stored in volatile SRAM memory cells), but other types include flash-based (using non-volatile flash memory) and antifuse-based (can only be programmed once). The Artix-7 is an SRAM-based FPGA.

2.2 Radiation characteristics

Let us define some commonly used terms for radiation and its interaction with electronic devices:

The particle **flux** f is the number of particle p that crosses the area a in one unit of time t. It is measured in $\mathrm{cm^{-2}\,s^{-1}}$, or equivalently in $\mathrm{Hz\,cm^{-2}}$.

$$f = \frac{p}{a \cdot t} \tag{2.1}$$

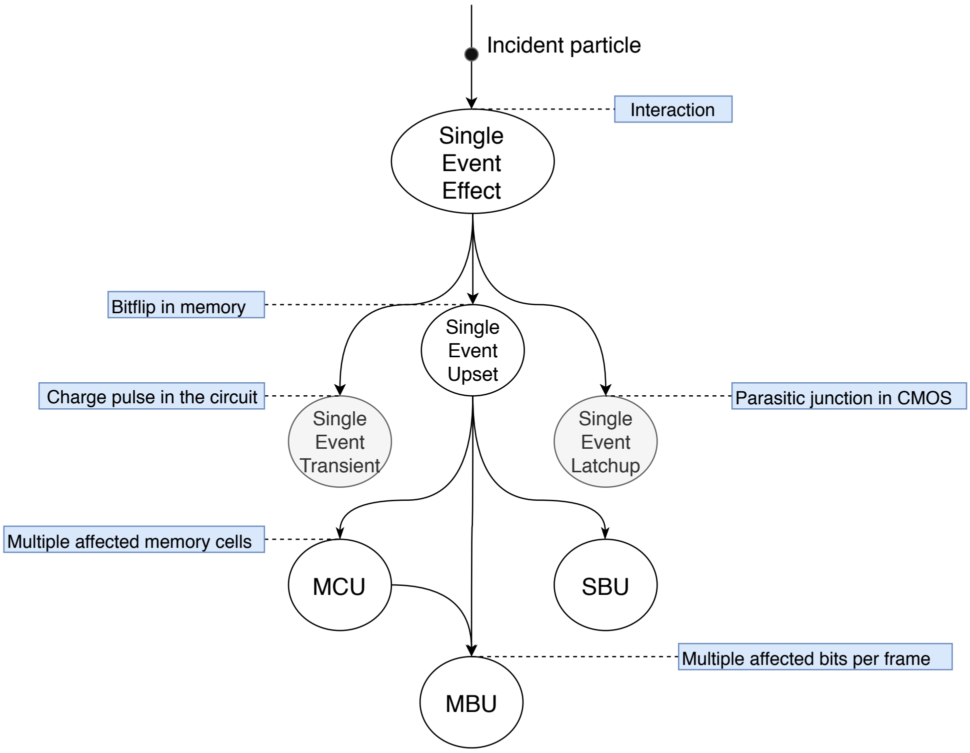


Figure 2.1: Summary of single event effects within an FPGA

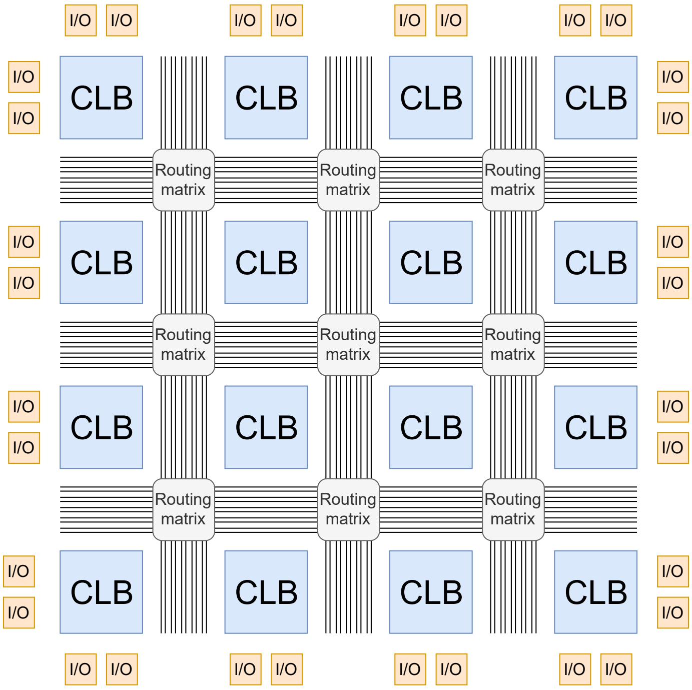


Figure 2.2: Generic FPGA architecture

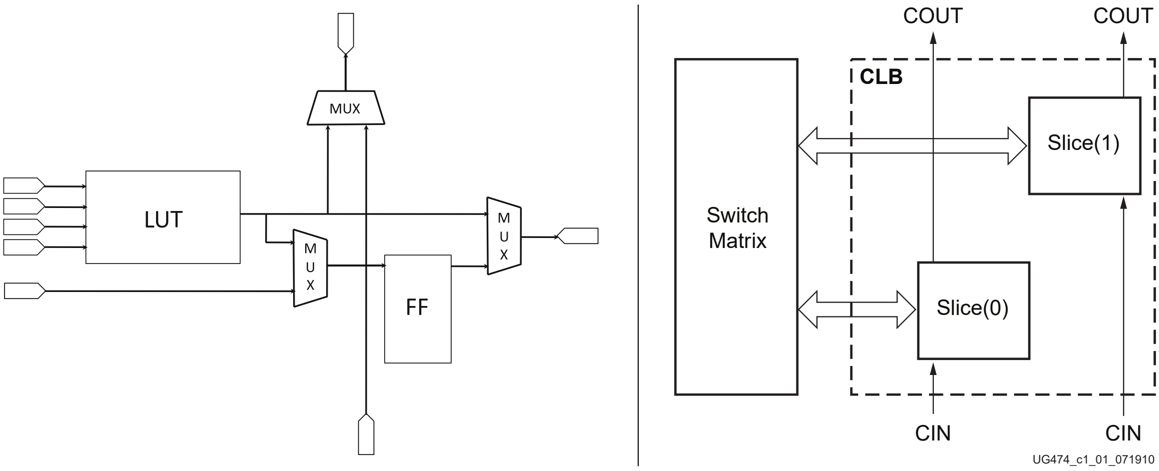


Figure 2.3: Generic configurable logic block with a single LUT (left). Xilinx CLB with two slices of 4 LUTs (right). [5]

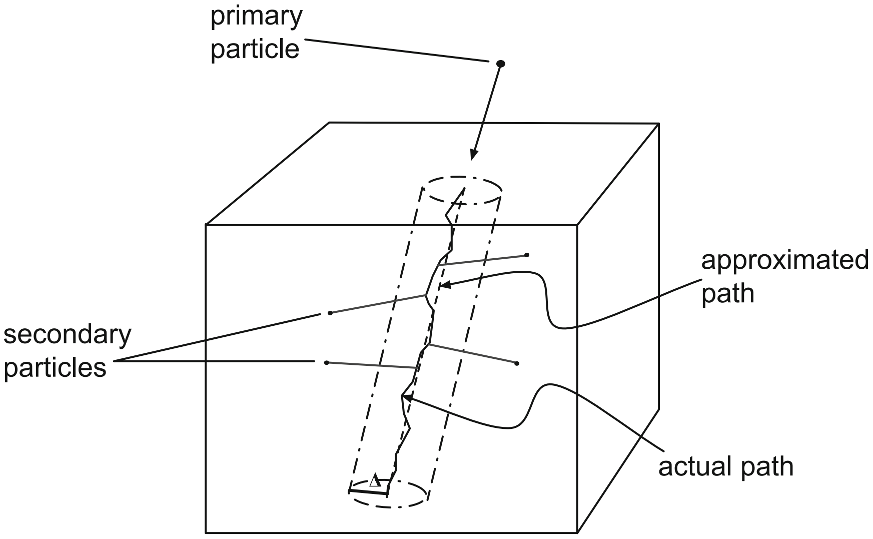


Figure 2.4: Penetration of a particle in a material [5]

The particle **fluence** Φ is the integral of the flux over time. It is measured in cm^{-2}.

$$\Phi = \int_0^t f(t)dt \tag{2.2}$$

The **cross-section** σ is a characteristic area of the device that represents the tendency of incident particles to interact with the target. In the context of electronic devices, it is defined as in Eq. 2.3, with N being the total count of some sort of event in the device (e.g. SEU, SET, see Section 2.4). It can be conceived of as the equivalent area of the device if all particles going through it produced an event.

$$N = \sigma \cdot \Phi \tag{2.3}$$

The **linear energy transfer** (LET) is a measure of the amount of energy per unit length transferred by an ionizing particle *to* an incident material [15]. It concerns only energy transferred by ionization, and in this it differs from the *stopping power*, which includes the nuclear collisions. One speaks of the *restricted* LET when it is focused on the energy transferred to the material in a *vicinity* of the track (See Figure 2.4) by means of secondary electrons with $E \leq \Delta$ emitted during the interactions (Δ being the energy required to leave the cylinder described in Figure 2.4). Indeed, higher energy electrons will travel further away into the material and will not affect the semiconductor along the track of the particle. The restricted LET is defined in Eq. 2.4, where dE_Δ is the energy lost by the particle while travelling a distance ds. The *unrestricted* LET is given by $\lim_{\Delta\to\infty} LET_\Delta$.

When studying the effects of radiation on electronic devices, LET is usually expressed in $\mathrm{MeV\,cm^2\,mg^{-1}}$, which is the standard definition of LET ($\mathrm{MeV\,cm^{-1}}$) divided by the material density.[5]

$$LET_{\Delta} = \frac{-dE_{\Delta}}{ds} \tag{2.4}$$

2.3 Accumulated effects

2.3.1 Displacement effect

When a heavy particle enters the semiconductor, it will be scattered by collisions with the nuclei of the material, dislocating atoms and creating point defects in the crystal lattice. The more energy is transferred from the particle to the material, the more the dislocations. These point defects are recombination centers for charge carriers in the material (so-called "carrier traps"), which will worsen the analog properties of the semiconductor. Such damage can be repaired by heating the device, providing energy to the lattice to repair itself (annealing). This effect is mostly caused by neutrons, protons, alpha particles, heavy ions and very high energy gamma photons [5].

2.3.2 Charge accumulation effect

Ionization is a mechanism by which charge is released within the material traversed by the particle. This can occur in two ways, through *direct* or *indirect* ionization. Direct ionization is caused by charged particles, such as heavy ions, electrons or positrons. When these particles enter the material, they can tear electrons away from neutral atoms, or give electrons to ionized atoms, provoking a movement of charge along the particle's path. Indirect ionization is caused by neutral particles, such as neutrons or photons, which can ionize the atoms within the crystal lattice through collisions, creating electron-hole pairs [5].

Charge accumulation is a consequence of both ionization and displacement effects. Charge within the semiconductor can become trapped in lattice dislocations, or in the insulation oxides around and within the transistor's structure. In particular, charge trapped near the gate oxide will create a permanent gate bias, influencing the transistor threshold voltage. This effect will lead to a gradual degradation of performance, up to a total loss of functionality when a transistor becomes permanently open (or closed) [10] [5].

2.3.3 Total Ionizing Dose

The Total Ionizing Dose (TID) is a measure of the total energy released into the material by ionizing radiation. Its SI unit is the Gray ($1\,\mathrm{Gy} = 1\,\mathrm{J\,kg^{-1}}$) but it is often measured in rad ($1\,\mathrm{rad} = 0.01\,\mathrm{Gy}$) [35]. It depends on exposure time, the flux of particles, and their LET. TID models the effects of charge accumulation as well as displacement damages, it is therefore associated to a global worsening in performance. Another effect of TID is a change in SEE sensitiveness, one consequence of which is that SEUs which would cause reversible *bit-flips* in memory elements (see Section 2.4.2) could cause so-called *stuck bits*, whose correct value cannot be restored. TID effects can usually be repaired through annealing [5].

2.4 Single Event Effects

Single event effects are the category of effects directly caused by an incident particle. They can be transient or permanent, repairable (*soft error*) or not (*hard error*) depending on the area affected. Among the soft error category are single event upsets (SEU), single event transients (SET) and single event functional interrupts (SEFI). This type of error can typically be repaired by reloading the FPGA firmware, or by cycling the power. One example of a hard error is the single event latch-up (SEL), which can cause permanent damage to the electronics. All of these will be presented in the following sections. [10]

2.4.1 Origin

An ionizing particle penetrating into a semiconductor device through its depletion layer will create a track of ionization composed of electrons and holes from the semiconductor atoms. The presence of this

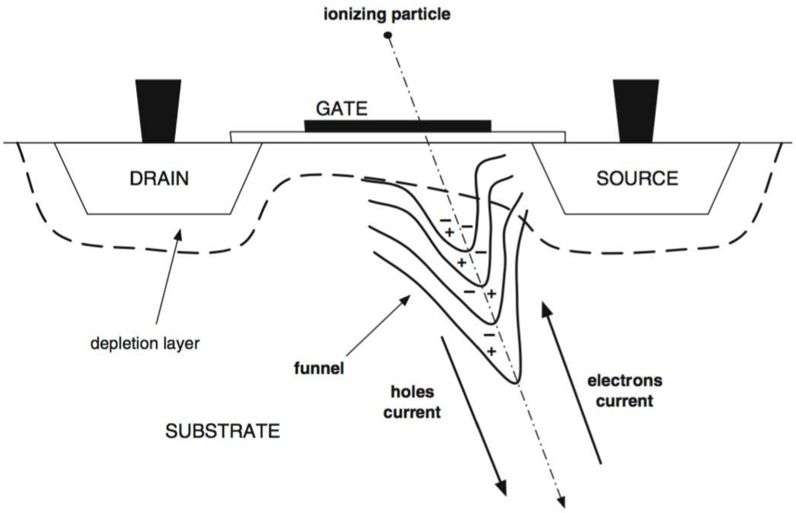


Figure 2.5: Illustration of the funnelling effect in a planar transistor [5]

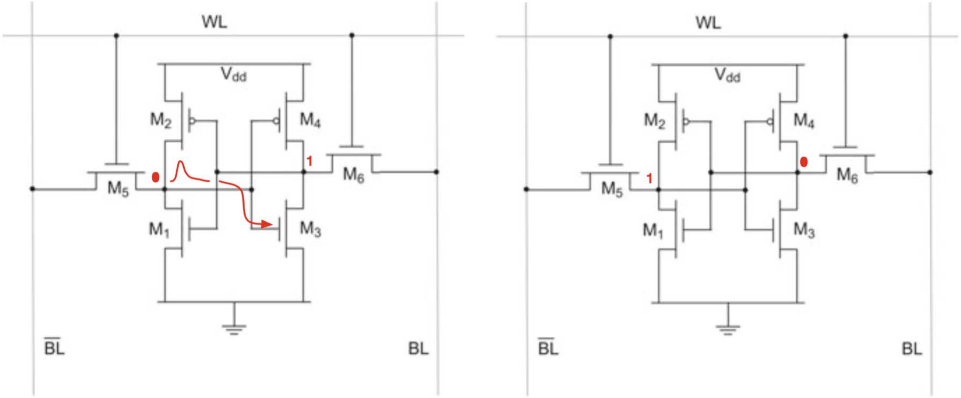


Figure 2.6: SEU in six-transistor (6T) SRAM memory cell [45]

track temporarily distorts the equipotential surfaces of the depletion layer's electric fields. This leads to funnel-shaped equipotential surfaces which can reach into the substrate (see Figure 2.5). Currents of electrons and holes are therefore created, and the deposited charge could be collected by one of the device nodes, resulting in a transient pulse in the circuit. This is especially likely if the particle strikes near the nodes. The amount of charge deposited, and therefore the magnitude of the pulse generated, will depend on the LET of the particle [5].

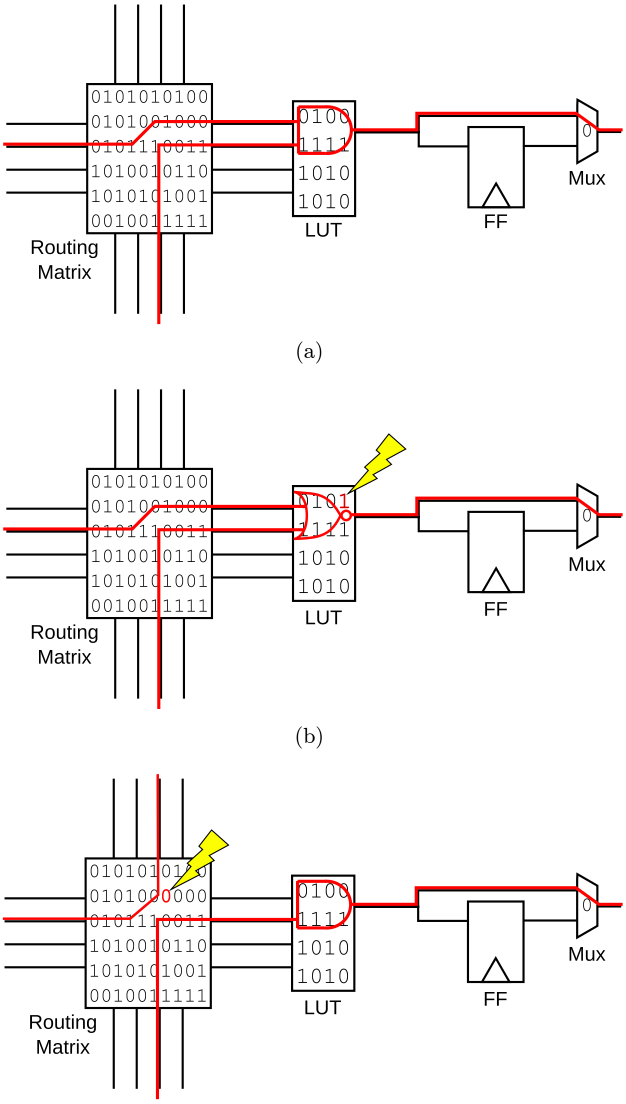
2.4.2 Categorization

Single Event Transients

Whenever the charge created by an ionization event discharges through the transistor, a energetic pulse lasting between picoseconds and nanoseconds (depending and pulse width and amplitude) will propagate in the device. This pulse is called a Single Event Transient, and will usually not cause issues in digital devices. If, however, this transient is sampled by a flip-flop, or if it occurs in a memory element, then it becomes a Single Event Upset.

Single Event Upsets

A Single Event Upset (SEU) is any change of state in an electronic device caused by a single particle. As mentioned above, this could be a flip-flop sampling a transient pulse or, as is much more frequent in SRAM-based FPGAs, a bit-flip directly in a memory element. The mechanism by which the value of an SRAM cell is modified by incident radiation is shown on Figure 2.6. In order for a memory element



(a)

(b)

(c)

Figure 2.7: SEU effects in configuration memory. Representing an upset in an LUT module (b) and in the routing matrix (c). [36]

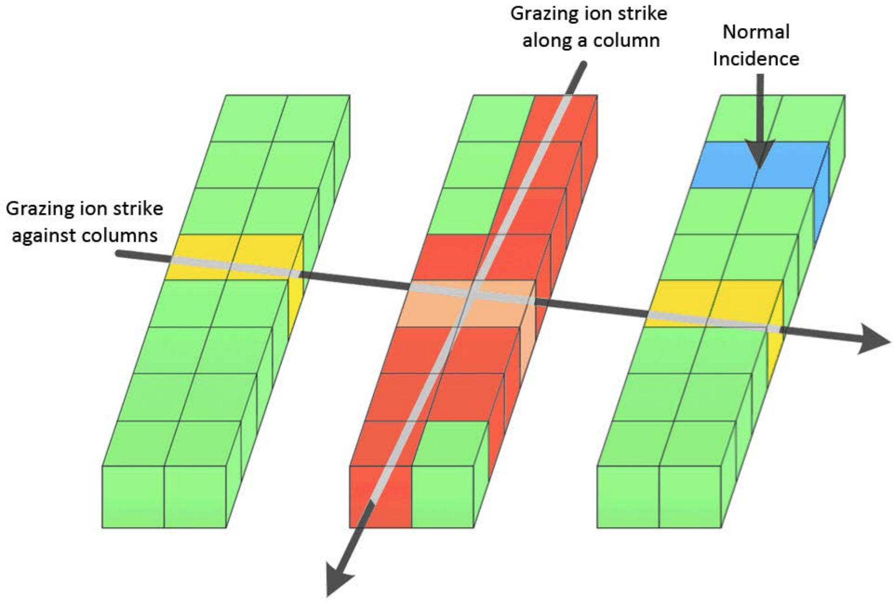

Grazing ion strike along a column
Normal Incidence
Grazing ion strike against columns

Figure 2.8: Multiple Cell Upsets [54]

to flip, the pulse must have a certain *critical charge* Q_c, which has been suggested to be quadratically dependent on feature size L ($Q_c \propto L^2$) [8], although predicting the effects of modern sub-µm scaling on SEU susceptibility is not as straightforward [22]. As this is the main source of concern for radiation-induced soft errors in this thesis, an SEU will from now on be referring to an SEU in memory cells.

An SEU in the configuration memory of an FPGA will potentially modify the design and can cause any number of issues, the nature of which is difficult to predict (see Figure 2.7). In most firmwares, the number of bits in the bitstream which will change the circuitry is only a fraction of the total bitstream (called *essential bits*)[9], the firmware used in Section 4.3 has an essential bit proportion of about 25%. SEUs can also occur in user memory (e.g. FIFO buffers), which is based around Block RAM (BRAM) in Xilinx FPGAs. Flash and antifuse-based FPGAs are much more resistant to SEUs in their configuration memory, up to total immunity for antifuse FPGAs, but tend to be less performant and both have limitations on their reprogrammability [55].

A particle traversing the FPGA can affect multiple memory cells, in that case the SEU is a **Multiple Cell Upset (MCU)**. This can occur when the particle is grazing the device at an angle, or with a normal incidence at the border between memory cells (particles tend to scatter around the silicon rather than follow a linear path), see Figure 2.8. Because a grazing particle is much more likely to go through multiple memory cells, the probability of an MCU will depend on the angle of the device relative to the incident beam of particle. One can also note that the internal layout of an FPGA is such that configuration memory cells are arranged in *columns* (Figure 2.8), therefore one orientation of the angle-of-incidence (along the columns) is much more likely to provoke MCUs than the other (across the columns).

MCUs affecting bits in the same memory frame are a special case, for reasons that will be explained later on, we therefore use a specific term, the **Multiple Bit Upset (MBU)**. If on the other hand the SEU only produces a single bit-flip, then it will be referred to as **Single Bit Upset (SBU)**.

Other sources in the literature might also define SEUs as being single bit upsets (i.e. a 3-bit MCU would count as 3 SEUs), please note that in this thesis an SEU will refer to the **particle event**, which causes one or more memory upsets.

Single Event Functional Interrupt (SEFI)

> "A soft error that causes the component to reset, lock-up, or otherwise malfunction in a detectable way, but does not require power cycling of the device (off and back on) to restore operability, unlike single-event latch-up (SEL), or result in permanent damage as in single event burnout (SEB)" - JEDEC [15]

A special case of SEU which causes functional problems with the FPGA, but not due to change in the design or user memory. This is often associated with an upset in the control registers of the device.

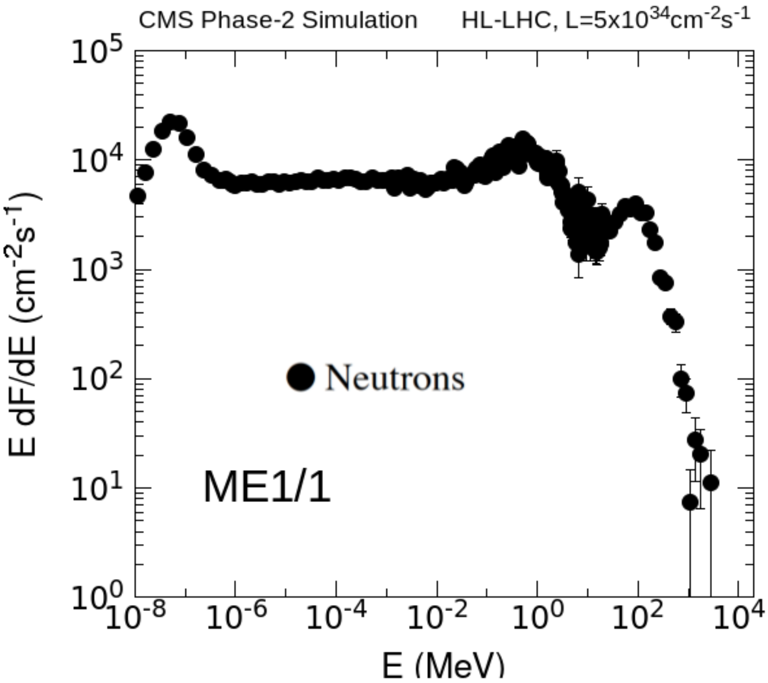


Figure 2.9: Energy breakdown of neutron radiation in CMS Phase-2 [10]

Particle	Possible effects	Interaction	Prevalence
Neutron	Ionization, displacement	High	High
Electron/Positron	Ionization	Medium	Low
Photon	Ionization	Low	Medium
Muon	Ionization	Low	High

Table 2.1: Summary of background radiation particles in CMS

Due to the extremely small number of bits existing as control registers, the frequency of SEFI events is usually considered negligible [42].

Single Event Latchup (SEL)

> "An abnormal high-current state in a device caused by the passage of a single energetic particle through sensitive regions of the device structure and resulting in the loss of device functionality. An example of SEL in a CMOS device occurs when the passage of a single particle induces the creation of parasitic bipolar (p-n-p-n) shorting of power to ground. SEL may cause permanent damage to the device. If the device is not permanently damaged, power cycling of the device (off and back on) is necessary to restore normal operation." - JEDEC [15]

Under heavy ion radiation and extreme conditions (above 90°C and with supply voltages exceeding maximum specifications), some SEL-like current signatures have been observed in a 7-series Xilinx FPGA [25].

2.5 CMS Environment

Now that the most prevalent radiation effects in FPGAs have been presented, we must go back to the CMS to identify which of the background radiation particles will be likely to cause SEUs in the OH FPGA.

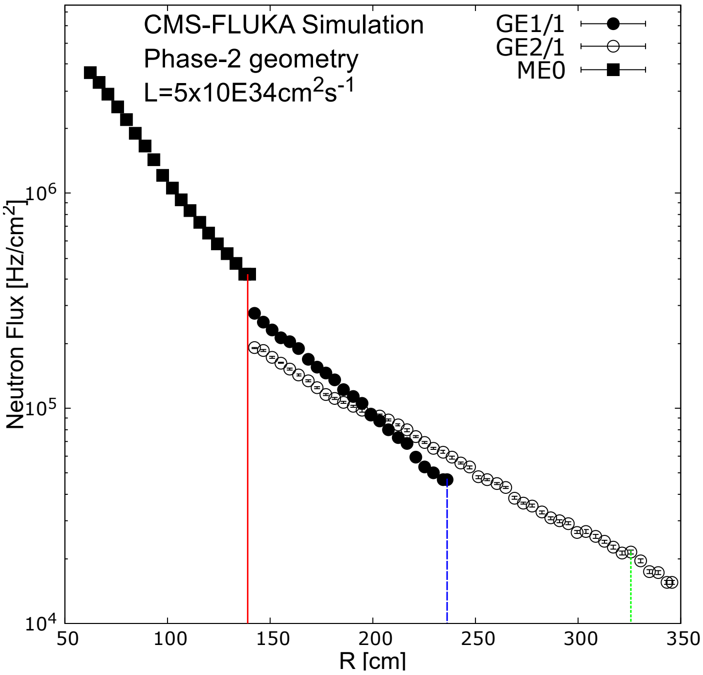


Figure 2.10: Expected neutron flux w.r.t. detector coordinates in CMS Phase-2. Expected flux for ME0 OH, GE1/1 OH, and GE2/1 OH drawn in colour for comparison [10]

.

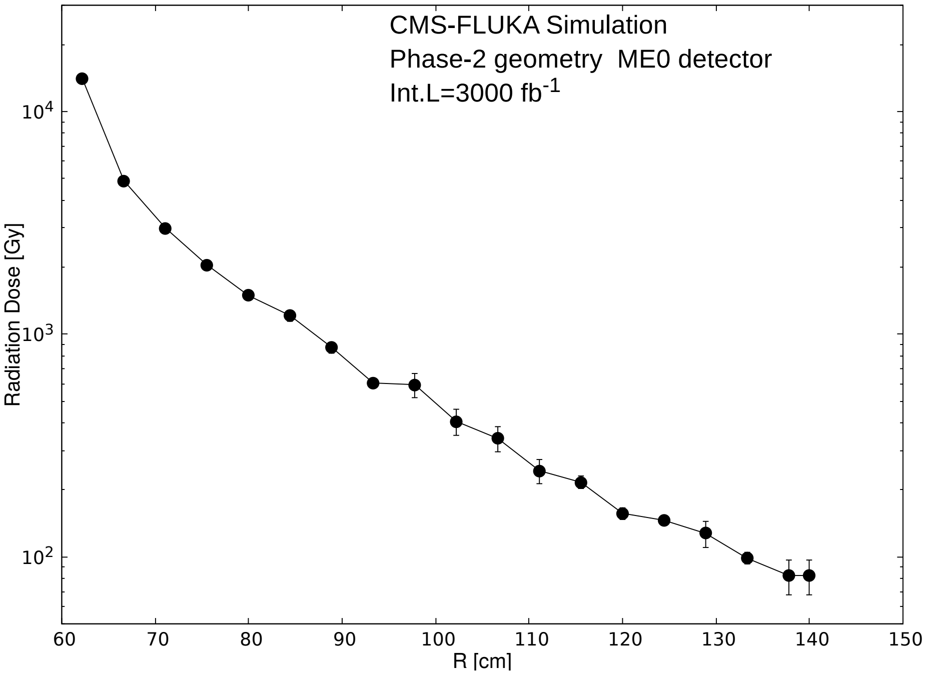


Figure 2.11: Expected TID w.r.t. detector coordinates in CMS Phase-2 [10]

.

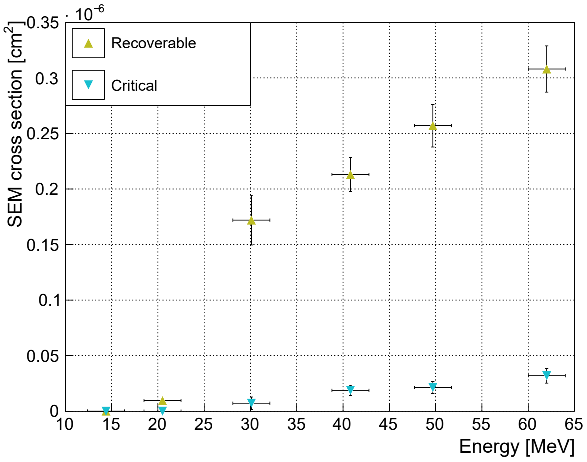


Figure 2.12: CRAM proton cross-section data related to proton energy for the Virtex-6 [29]

The CMS background radiation is mainly composed of neutrons, electrons (and positrons), photons and muons. The dominant source of SEUs will be neutrons. Muons do not have a significant impact on SRAM memory cells [51], they will therefore not influence the SEU rate. The combined flux of electron and positrons is over two orders of magnitude lower than neutrons, and the electron cross-section at the CMS energy levels is two order of magnitudes lower than the neutrons'. Photons have a similar flux but will not interact with the silicon of the FPGA [29]. Table 2.1 summarizes the effects and prevalence of the principal background radiation particles in CMS.

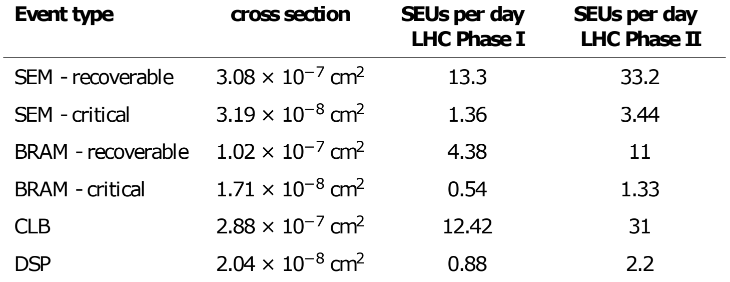
The Technical Design Report for the Phase-2 Upgrade of the CMS detector contains information on the environmental radiation [10]. These values were obtained through complex computer simulations of the CMS detector and cavern after a beam crossing, including among other things the neutrons produced by the hadronic showers in the detector. Figure 2.9 contains the expected neutron energy distribution in the detector. This can be used along with an energy-related SEU cross-section (e.g. CRAM, BRAM) to predict an effective SEU rate in CMS. Figure 2.10 shows the expected neutron flux for various elements depending on their position within CMS coordinates, one can see from this graph that an approximate increase of one order of magnitude in the neutron flux is to be expected for ME0 relative to GE1/1 (the OH location on the detector board will be the furthest away from the center of the CMS, around $R = 150\,\text{cm}$, see Figure 1.5). This would be in the $10^5\,\text{Hz}\,\text{cm}^{-2}$ range. The neutron flux expected for GE2/1 is comparable to the GE1/1 levels, in the $10^4\,\text{Hz}\,\text{cm}^{-2}$ range. Finally the Total Ionizing Dose (TID) for the ME0 OptoHybrid is estimated at around 30 krad (Figure 2.11). As both the ME0 and G2/1 use the same Artix-7 FPGA, it is sufficient to qualify the board for the highest radiation level at the ME0 location.

An additional consideration is that the proportion of MCUs among SEUs is related to the tilt of the beam relative to the surface of the FPGA [26] (see Section 5.1.2). In the case of the ME0 detector, the incident angle relative to the interaction point is $\approx 15°$, but in the CMS cavern, particles can be expected from all direction. Simulation data on the probability distribution of the particle angle-of-incidence would therefore be useful to give accurate predictions on the proportion of MCUs and MBUs, however this is not currently available.

2.6 Existing studies

GE1/1

Previous studies on the radiation resistance of the OptoHybrid board's FPGA were done at the Inter University Institute For High Energies (Brussels) in 2016 for the GE1/1 detector [28] [29], which used a Virtex-6 FPGA. The tests were conducted at the UCLouvain's Cyclone particle accelerator, using a beam of protons (which have an interaction cross-section very similar to neutrons [52]). The results obtained by T. Lenzi are reproduced on Figures 2.12 and 2.13. The SEM data refers to the SEU detection events reported by the SEM IP Core (see Section 3.2.5). The number of SEUs per day is extrapolated from the expected flux at GE1/1, the neutron energy distribution, and the cross-section data related to proton energy shown on Figure 2.12. One can observe that the cross-section reduces greatly at 20MeV.



Event type	cross section	SEUs per day LHC Phase I	SEUs per day LHC Phase II
SEM - recoverable	3.08×10^{-7} cm^2	13.3	33.2
SEM - critical	3.19×10^{-8} cm^2	1.36	3.44
BRAM - recoverable	1.02×10^{-7} cm^2	4.38	11
BRAM - critical	1.71×10^{-8} cm^2	0.54	1.33
CLB	2.88×10^{-7} cm^2	12.42	31
DSP	2.04×10^{-8} cm^2	0.88	2.2

Figure 2.13: Proton cross-section data for the Virtex-6 [29]

Energy	σ_{upset} per device	σ_{upset} normalized by CRAM size
35 MeV	$15 \times 10^{-8}\,\mathrm{cm}^{-2}$	$7.94 \times 10^{-15}\,\mathrm{cm}^{-2}\,\mathrm{bit}^{-1}$
100 MeV	$10 \times 10^{-8}\,\mathrm{cm}^{-2}$	$5.3 \times 10^{-15}\,\mathrm{cm}^{-2}\,\mathrm{bit}^{-1}$
200 MeV	$6.75 \times 10^{-8}\,\mathrm{cm}^{-2}$	$3.57 \times 10^{-15}\,\mathrm{cm}^{-2}\,\mathrm{bit}^{-1}$

Table 2.2: Cross-section data for the Kintex-7 [33]

It should be noted that cross-section results, and the calculated time between upsets is based on the total number of bits in the configuration memory. Of those bits, only a fraction (the exact value depends on the firmware) are actually essential to the functionality of the design. Should a non-essential bit be affected, the design will not change.

CRAM SEU cross-section

Xilinx provides some neutron cross-section data for all their FPGA families in the "Device Reliability Report", though the Los Alamos Neutron Science Center (LANSCE) facility where their tests were run is a wide-spectrum neutron beam, therefore there are no cross-section values for specific energy levels. At the time of the preliminary research for this thesis, besides the Xilinx report, no specific irradiation data for the Artix-7 FPGA was found, however several studies had been made on the Kintex-7 FPGA, which has a very similar architecture [1]. The results obtained by one of those studies is reproduced in Table 2.2. The cross-section results are normalized by CRAM size, as a larger FPGA (i.e. with more logic cells) will evidently have more errors than a smaller one with the same density. Comparing the trends in Figure 2.12 and Table 2.2, we can see that in one case the cross-section rises with energy, and in the other it falls with energy. The latter could be explained by the fact that higher energy protons (and neutrons), which can travel further into the silicon, actually have a lower LET (see Figure 2.15) [19]. The former may be occurring because the energy of the incident protons falls below the threshold of the nuclear reactions required for ionization of the silicon [7]. This explanation may be somewhat simplistic, however going into further detail would require a more in-depth approach to neutron interactions in the silicon lattice, which is outside the scope of this chapter. It is sufficient for our purposes to note that this behaviour has been observed in other SEE studies [7].

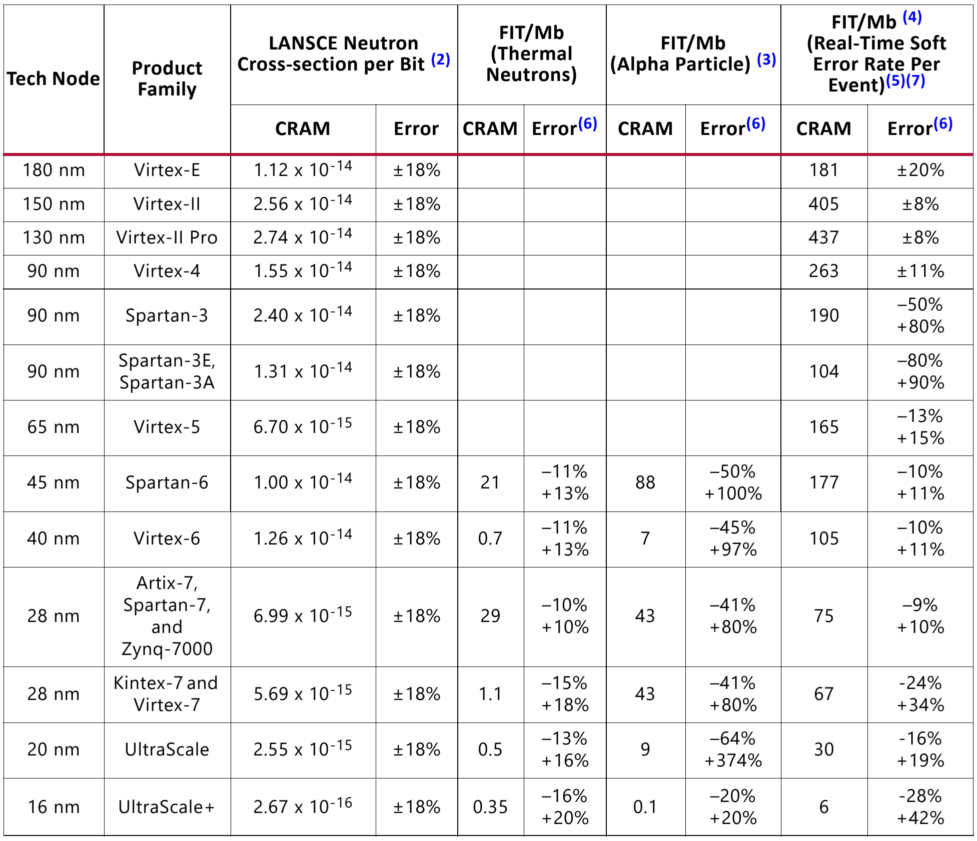
These results can be compared to the Xilinx data, reproduced in Figure 2.14. It can be seen that the Artix-7 and Kintex-7 cross-sections are close, but slightly higher for the Artix-7. The Kintex-7 results from Xilinx were also independently verified by a research group at the NSF Center for High Performance Reconfigurable Computing [56].

MCU proportion

Existing data already shows that LET and angle of incidence have a large impact on the proportion of MCUs in Kintex-7 FPGAs [26] [57]. There is no reason to think the same should not expected for the Artix-7. At normal incidence, the proportion of MCU in the Kintex-7 was measured at $\approx 10\%$ for intra-frame MBUs, and $\approx 35\%$ for all MCUs [56] at the LANSCE wide-spectrum neutron beam.

TID

Concerning the TID, previous irradiation campaigns on the Kintex-7 have reached values of 500krad [33], and even 1 Mrad [34], without permanent damage to the device. This is much higher than the 30krad



Tech Node	Product Family	LANSCE Neutron Cross-section per Bit [2]		FIT/Mb (Thermal Neutrons)		FIT/Mb (Alpha Particle) [3]		FIT/Mb [4] (Real-Time Soft Error Rate Per Event)[5][7]	
		CRAM	Error	CRAM	Error[6]	CRAM	Error[6]	CRAM	Error[6]
180 nm	Virtex-E	1.12 x 10^{-14}	±18%					181	±20%
150 nm	Virtex-II	2.56 x 10^{-14}	±18%					405	±8%
130 nm	Virtex-II Pro	2.74 x 10^{-14}	±18%					437	±8%
90 nm	Virtex-4	1.55 x 10^{-14}	±18%					263	±11%
90 nm	Spartan-3	2.40 x 10^{-14}	±18%					190	–50% +80%
90 nm	Spartan-3E, Spartan-3A	1.31 x 10^{-14}	±18%					104	–80% +90%
65 nm	Virtex-5	6.70 x 10^{-15}	±18%					165	–13% +15%
45 nm	Spartan-6	1.00 x 10^{-14}	±18%	21	–11% +13%	88	–50% +100%	177	–10% +11%
40 nm	Virtex-6	1.26 x 10^{-14}	±18%	0.7	–11% +13%	7	–45% +97%	105	–10% +11%
28 nm	Artix-7, Spartan-7, and Zynq-7000	6.99 x 10^{-15}	±18%	29	–10% +10%	43	–41% +80%	75	–9% +10%
28 nm	Kintex-7 and Virtex-7	5.69 x 10^{-15}	±18%	1.1	–15% +18%	43	–41% +80%	67	-24% +34%
20 nm	UltraScale	2.55 x 10^{-15}	±18%	0.5	–13% +16%	9	–64% +374%	30	-16% +19%
16 nm	UltraScale+	2.67 x 10^{-16}	±18%	0.35	–16% +20%	0.1	–20% +20%	6	-28% +42%

Figure 2.14: Xilinx configuration memory cross-section data [14]

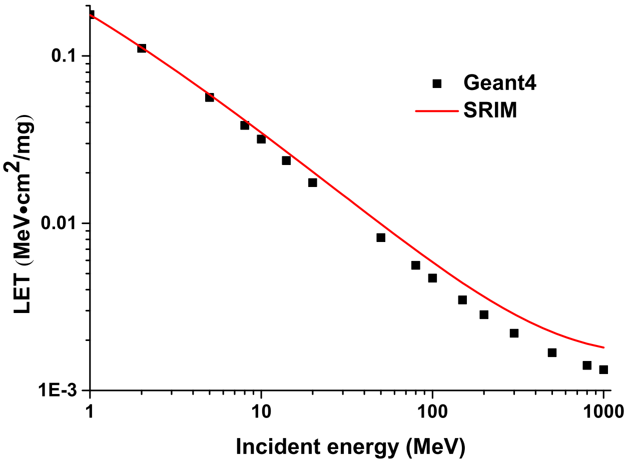


Figure 2.15: Simulated proton LET in silicon [19]

Family	Device	Logic cells	BRAM (Max Kb)	Bitstream length (bits)
Virtex-6	VSX315T	314,880	25,344	104,465,888
Virtex-7	VX330T	326,400	27,000	111,238,240

Table 2.3: Comparison of two FPGAs with similar logic capabilities across two families.

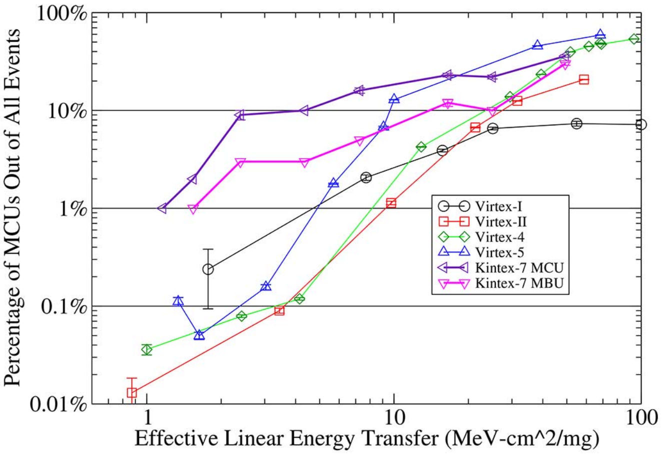


Figure 2.16: MCU proportion out of all events, related to heavy ion LET and device family [57]

expected over the lifetime of the ME0 detector, therefore no significant accumulated effects are expected for the Artix-7.

2.6.1 Scaling trends

SEU cross-section

Looking at the historical cross-section data in Figure 2.14, it can be seen that the SEU rates per CRAM bit are going down with shrinking technology nodes, which seems counter-intuitive. Xilinx reports that this as been achieved through improvements in circuit design and layout techniques. By comparing two models in the Virtex-6 and Virtex-7 families with a similar amount of logic elements (see Table 2.3) we can see that the bitstream length (i.e. the number of configuration bits in the FPGA) remains stable. This seems to indicate that some technological improvement towards radiation hardening was indeed made in the circuit design.

As technology nodes go beyond 20nm, FinFET architectures can be expected to replace the current planar devices. Indeed, the high-end Xilinx Ultrascale+ family is now built on a 16nm FinFET process. Interestingly, experimental data seems to indicate that, when compared to 20nm and 28nm planar devices, the SEU cross-section is lower at low LETs ($< 10\,\text{MeV}\,\text{cm}^2\,\text{mg}^{-1}$), while staying stable at higher LETs [32]. From the Xilinx data we can see an order of magnitude reduction in SEUs for the 16nm FinFET Ultrascale+ when compared to the planar 20nm Ultrascale, with the LANSCE neutron beam.

MCU proportion

Although the SEU cross-section may improve in new technology nodes, the proportion of MCUs in planar devices is unmistakingly increasing, see Figure 2.16 (note the logarithmic scale). As for FinFET devices however, preliminary testing on the Ultrascale+ family with heavy ions reveals a significantly reduced MCU response relative to the Kintex-7 at $\approx 7\%$ in the $20\,\text{MeV}\,\text{cm}^2\,\text{mg}^{-1}$ region [27]. As we shall see in the next chapter, MBUs are quite problematic in simple SEU mitigation schemes.

Chapter 3

SEU mitigation techniques

3.1 Radiation hardening solutions

Radiation hardening is the process of making an electronic device/system resistant to damage or malfunctions caused by radiation. The means by which radiation can affect an FPGA have been presented in the previous chapter. As it was established that accumulated effects would not be a problem in the context of CMS, we will focus on solutions to single-event effects. In particular, we will focus on mitigating the effects of single event upsets in the configuration memory.

Just as there are multiple levels of abstraction for VLSI design (Figure 3.1), there are multiple levels of abstraction for radiation hardening depending on when it is applied in the development of an electronic device. Figure 3.2 illustrates the 3 types of radiation hardening and relates them to FPGA-appropriate abstraction levels.

The lowest level solution is Radiation Hardening by Process (RHBP), where the transistors of the electronic devices have been fabricated with semiconductor process technologies which temper the component against the effects of incident particles. This can involve specific material selection, insulation layers (e.g. Silicon-On-Insulator devices are immune to single event latch-up [30]), doping levels, and others. This is an expensive solution, which relies on dedicated foundries. Because of this, and the small size of the market, RHBP electronics tend to lag behind commercial-off-the-shelf (COTS) equipment by a decade or more [23].

Radiation Hardening By Design (RHBD) is the next level over, it implies that the electronic device or component has been radiation-hardened by specific choices in layout and circuit architecture. Examples of RHBD include increased component spacing, isolation trenches in the chip layout, component redundancies built into the circuit, and the memory frame interleaving scheme in Xilinx FPGAs (see Section 3.2.5). Although these methods can be applied with normal manufacturing processes, there is a trade-off being made with other device characteristics, be it in integration density (e.g. component spacing), power consumption (e.g. redundancies) or cost, to name a few. For these reasons, mainstream COTS devices will only include a limited amount of RHBD features at best.

Finally, Radiation Hardening by Architecture (RHBA) includes all techniques which can be applied when integrating non-hardened components into the final system/device. In an ASIC (Application-Specific Integrated Device), this could be adding redundancy by duplicating a non-hardened standard cell. For an FPGA, this will principally be schemes applied in the firmware. At the firmware level, the standard scheme for SEU mitigation in the configuration memory of an FPGA is done by combining two complementary techniques: **Memory scrubbing** and **N-Modular redundancy**. Scrubbing means going through the memory and iteratively repairing damaged sections. This of course requires some sort of data redundancy, either in the form of error-correction codes (ECC), or access to a clean version of the firmware. Modular redundancy is a way to mask the effects of an SEU in the CRAM, until the scrubbing can repair it. The principle is simple, duplicate the firmware module which requires protection N-times, and add a majority voter at the output. The devil, though, is in the details.

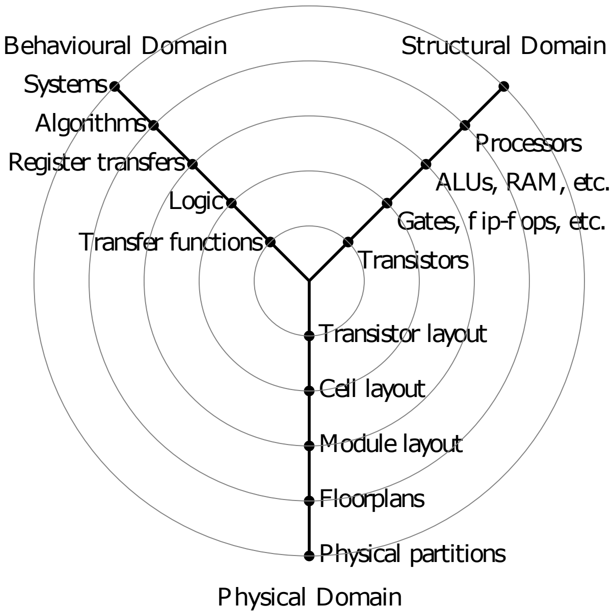


Figure 3.1: Gajski-Kuhn Y chart [18]

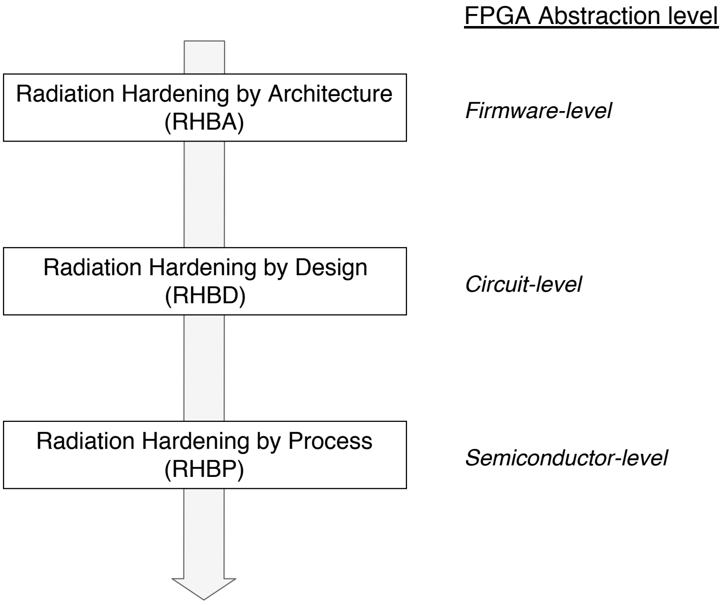

FPGA Abstraction level
Radiation Hardening by Architecture
(RHBA)
Firmware-level
Radiation Hardening by Design
(RHBD)
Circuit-level
Radiation Hardening by Process
(RHBP)
Semiconductor-level

Figure 3.2: Radiation hardening types, related to the abstraction level of an FPGA

3.2 Scrubbing architectures

3.2.1 Blind scrubbing

The simplest approach to memory scrubbing is to systematically rewrite every frame of memory. This is called blind scrubbing as no attempt is made to verify the content of the memory before rewriting it. This approach requires the use of a **golden copy**, a copy of the firmware known to be correct, stored in an accessible location (e.g. a rad-hard memory near the FPGA, or a low latency off-detector copy).

The total scrubbing time for blind scrubbing is

$$T_{s_{blind}} = N \cdot T_c \tag{3.1}$$

where N is the number of frames in the CRAM and T_c is the latency of one write operation to a CRAM frame [47]. It may be desirable to add delays to the write operations to ease processing bandwidth resources, in this case then

$$T_{s_{blind}} = N \cdot (T_c + T_{delay}) \tag{3.2}$$

Advantages	Disadvantages
Supports correction of all SEU types (SBU, MCU, MBU)	No detection capability
	Requires additional hardware
	Requires a rad-hard golden copy

3.2.2 Readback scrubbing

Readback scrubbing, as opposed to blind scrubbing, will systematically check every frame of memory for errors, and will repair/replace if needed. This approach is best suited for use with ECC-protected memory frames, but one might also use a golden copy for error detection and/or correction. In Xilinx FPGAs, each frame has built-in Single Error Correction, Double Error Detection (SECDED) ECC redundancy, along with a device-wide CRC check for supplemental error detection capability. In addition, a built-in ASIC module enables the configuration memory soft error correction capabilities (Readback CRC). A fully functional readback scrubbing system is also available from Xilinx: the SEM IP Core, which extends the Readback CRC and provides a simple user interface with additional features.

The worst case scrubbing time for readback scrubbing is

$$T_{s_{readback}} = N \cdot T_d + u \cdot T_c \tag{3.3}$$

Where u is the number of frames with upsets in the CRAM, and T_d is the amount of time required to read back and compare one frame. Assuming an upset occurs in a frame right *after* it has been checked for errors, then the scrubber will check the whole memory before detecting it, and Eq. 3.3 holds.

ECC-based

Advantages	Disadvantages
Easy to implement with Xilinx FPGA	No MBU correction capability with built-in Xilinx ECC
No additional hardware required	

As mentioned in Section 2.6, MBUs (intra-frame MCUs) are already quite frequent in current FPGA technologies. As the transistor density increases, this trend will continue. It is possible to improve the correction capabilities by using a more complex error correction code, but to do this one would have to forgo the built-in ECC, find another way of providing memory redundancy, and re-implement the scrubbing architecture.

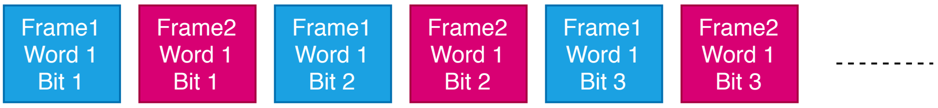


Figure 3.3: Diagram of the memory interleaving principle

Golden copy

The principle is the same for readback scrubbing with a golden copy, the frames are checked for error before initiating the rewriting operation. One could also use an error-detection code on the memory frames (e.g. Xilinx built-in ECC) to avoid fetching every frame of the golden copy. With the SEM IP Core, the correction latency in golden-copy mode, which uses the ECC for error detection, is higher than ECC-correction by $\approx 35\%$ [42]. Additional latency may be incurred in case of I/O bottleneck.

Advantages	Disadvantages
Supports correction of all SEU types (SBU, MCU, MBU)	Slower than ECC-based
Easy to implement with Xilinx FPGA	Requires additional hardware
	Requires a rad-hard golden copy

This type of Readback scrubbing is quite attractive as it can correct MBUs, it is also easy to implement with Xilinx FPGAs as it is natively supported by the SEM IP Core. However a rad-hard memory is not available in the current version of the OptoHybrid board.

3.2.3 Hybrid schemes

Combining the speed of ECC correction with the MBU-correction capabilities of golden copy based solutions is an attractive proposition, as it can be implemented on top of the existing Xilinx soft error correction system. One proposed architecture [47] consists of using the built-in Readback CRC hardware for error detection and single-bit correction, and to use the golden copy when a non-correctable error is found. This requires custom firmware/software.

Again, this is not currently a workable solution for the OptoHybrid board, as it does not contain rad-hard memory.

3.2.4 External solutions

A recurrent problem with the architectures presented above is that the logic required to perform the scrubbing is implemented inside the FPGA (either in firmware, or in software with a soft CPU for more complex algorithms) which, besides using up valuable ressources, is of course also vulnerable to radiation.

To go around this problem, one could use the external configuration interfaces of the FPGA (e.g. SelectMAP) along with a secondary processing unit. This secondary processing unit does not have to be as fast as the main FPGA, which means it could be rad-hard. This could be a microcontroller or another FPGA, depending on the requirements. The hybrid scheme mentioned in Section 3.2.3 was implemented using a Xilinx Zynq SoC (FPGA and microcontroller integrated in the same package).

3.2.5 Xilinx Soft Error Mitigation solution

Although the 7-series Xilinx FPGA family is not specifically designed for radiation intensive applications, Xilinx provides a suite of tools for soft error mitigation (SEM). At the IC level, there is a built-in error detection and correction system, called "Readback CRC". Additionally, the scaling trends shown in Section 2.6.1 seem to indicate that improvements are also being made at the circuit level to reduce the SEU cross-section. Finally, the configuration memory is interleaved in such a way that logically adjacent bits are stored in physically separated memory cells [21]. The interleaving layout used by Xilinx is not known, but Figure 3.3 illustrates this principle. As MCU events tend to happen in adjacent memory cells, this trick will transform some potential MBU events into an inter-frame MCUs which are correctable by the built-in SECDED ECC.

The Xilinx SEM systems will be presented here briefly, the interested reader can find many details on its functioning and use in the MSc. Thesis of Aaron Gerald Stoddard [47]. Because the information

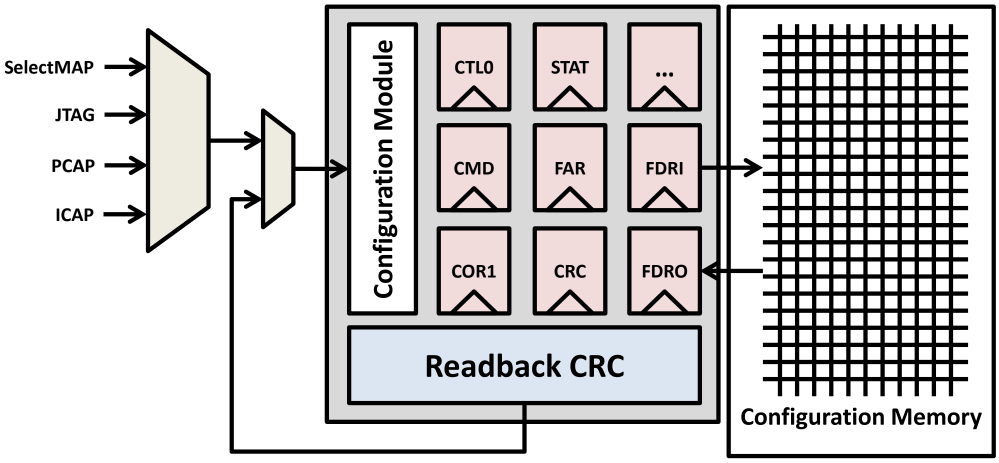


Figure 3.4: Xilinx FPGA configuration systems [47]

contained within it is available from Xilinx mostly in the form of scattered datasheets, it is a useful reference for anyone wishing to build a custom scrubbing architecture.

Memory frame redundancies

A 7-series Xilinx configuration memory frame is composed of 101 words of 32 bits. Each frame in the CRAM has a built-in ECC SECDED word at the 51st word. The 13-bit syndrome computed from it can detect the position of single bit errors, and therefore correct them. It can also detect the presence of two-bit errors as well as any odd-numbered error, though even-numbered errors are not guaranteed to be detected. Additionally, a 32-bit Cyclic Redundancy Check (CRC) is pre-computed for the entire configuration memory. A CRC works similarly to a hash function, in that its value will change even for a very small change in its input (i.e. the configuration memory), even a single bit, such that it can be used for device-wide error detection. The drawback is that it cannot find the position of an error (and therefore can't correct it), but used in conjunction with the ECC, it provides a redundant check, and permits detection of even-numbered errors.

Readback CRC

The Readback CRC is a dedicated hardware engine introduced with the Virtex-5 and available in all 7-series models. It continuously scans all the configuration memory, runs individual ECC checks, and re-computes then compares the CRC value for the entire device. It is built with dedicated on-board circuitry, which enables it to perform very fast SBU correction and MBU detection.

The Readback CRC can be monitored through the FRAME_ECC2 primitive, which provides information such as the current processed frame address, ECC/CRC error detection flags, etc. The information provided by this primitive is essential to many scrubbing architectures, including the SEM IP Core provided by Xilinx.

The behaviour of the Readback CRC upon detection of an error is configurable in the user constraints file (UCF). The user can configure it to either continue without correction, stop, correct then stop, or correct then continue. The Readback CRC interacts with the configuration memory at the lowest priority (see Figure 3.4), therefore an alternative/additional scrubbing system can augment it by using its data through the FRAME_ECC2 primitive, and perform its own CRAM operations by taking control of the configuration module. One such system is the Xilinx SEM IP Core.

SEM IP Core

The Soft Error Mitigation (SEM) Controller, is an FPGA module available from Xilinx (an *IP Core*), it is an easy-to-use pre-verified solution to detect and correct soft errors in the configuration memory. As mentioned before, it is built on top of the Readback CRC hardware module, and provides the following features:

- ECC-based correction

Start-up	123.8 ms
Detection (average)	4.0 ms
Detection (maximum)	8.0 ms
Correction (1 bit)	610 µs
Classification	750 µs

Table 3.1: SEM operation delays for XC7A75T at maximal ICAP frequency (100 MHz)

- Enhanced correction (some MBU correction capability at the cost of higher correction latency)
- Replace correction (i.e. golden copy)
- Optional essential bit classification (disregard uncorrectable errors on CRAM bits which do not affect the design)
- Error injection interface
- Status flags (heartbeat, initialization, observation, correction, classification, injection, essential, uncorrectable)
- ASCII interface for monitoring and command

Although the Readback CRC already performs ECC correction, the SEM core has additional correction capabilities which are implemented through the Internal Configuration Access Port (see Figure 3.4). Delays for detection and correction of all repair modes for various models at various operating frequencies can be found in the SEM datasheet [42]. Table 3.1 summarizes the delays for the ECC-based repair mode with the XC7A75T model at maximal ICAP frequency (100 MHz). Equation 3.4 can be used to convert to lower frequencies.

$$Latency_{ACTUAL} = Latency_{ICAP\_F_{max}} \cdot \frac{ICAP\_F_{max}}{Frequency_{ACTUAL}} \tag{3.4}$$

ECC-protected BRAM

The standard Block RAM primitive in 7-series Xilinx FPGAs is 72-bit wide and 512 words deep (36Kb total). It can be configured as a single 512 x 64 RAM using the 8 extra bits for Hamming code error correction. The eight redundancy bits (ECCPARITY) are generated during each write operation and stored with the 64-bit data into the memory. These ECCPARITY bits are used during each read operation to correct any single-bit error, or to detect (but not correct) any double-bit error. The built-in error correction system has two flags to communicate the ECC status after a read operation (SBITERR for single bit error and DBITERR for double bit error). It can be configured to repair the detected errors during a read operation, or only report them. There is also an error injection interface for testing purposes.

3.3 Modular redundancy

An upset in the configuration memory has the potential to modify the behaviour of the firmware. The goal of modular redundancy is to mask the effect of such an upset, until it is corrected. The principle is illustrated in Figure 3.5, with a triplicated configuration. Of course this type of protection will fail if more than one module is affected by an upset. The likelihood of such a failure depends on how the modular redundancy is implemented, this will be discussed in the following sections.

3.3.1 N-modular redundancy

Figure 3.5 used a triplicated configuration, which is how modular redundancy is usually implemented. A naïve way to improve the resiliency of the protection is to increase the duplication factor to more than three modules. The reliability of an N-modular system can be estimated using a Markov chain stochastic model. Through this method, it can be shown [43] that only modest improvements are achieved with an increase in the module duplication factor. An increase in the duplication factor is associated with a higher resource usage, which leads to higher cost and power consumption, and a more complex majority voter, which will also be more vulnerable because of the increase in signal routing (routing matrices are

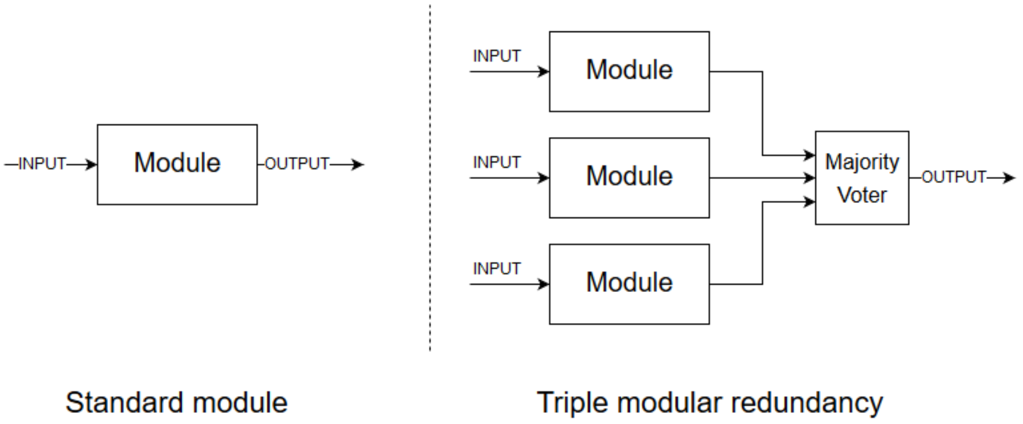


Figure 3.5: Illustration of (triple) modular redundancy

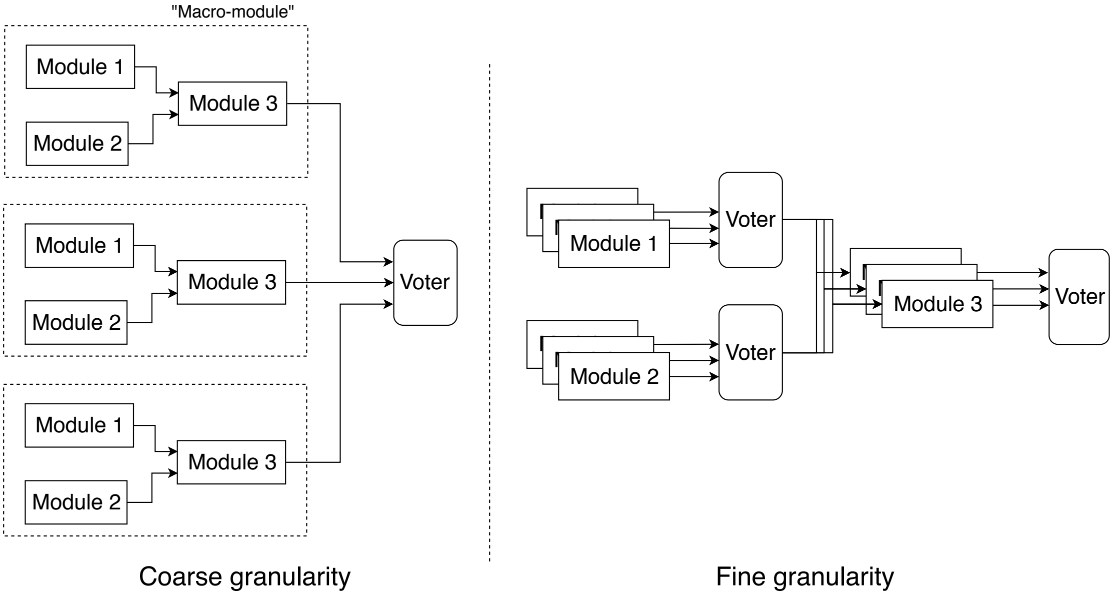


Figure 3.6: Example of granularity in the context of Triple Modular Redundancy

a major source of SEU-related errors [39]). Triple modular redundancy (TMR) is therefore the obvious choice.

3.3.2 Granularity

Until now, the term "module" was used in this thesis as an abstract notion of some functional block within the FPGA design. To implement TMR, a choice must be made regarding the logical granularity of the triplication. An illustration of the concept of granularity is shown on Figure 3.6. Finer granularity is associated with a lower failure rate (see Section 5.2.2), but at the cost of a potentially higher resource usage (more voters) and higher delays (propagation through each voter) or higher latency (with synchronized voting). The complexity of a voter depends on the width of the data bus (see Section 3.3.5), which means the actual relation between resource usage and granularity is very design-dependent. If we assume the example design used in Figures 3.5 and 3.6 has a constant data width for every bus, the resource usage of the voters will scale linearly with the number of modules triplicated.

3.3.3 Signal triplication and TMR logical areas

In the TMR implementations shown on Figure 3.6, the voters are single points of failure, and any error in the data path will propagate towards the output. This is also true for any signal (i.e. wire, connection) that is not triplicated as the routing matrices can be upset, including the input and output signals. A

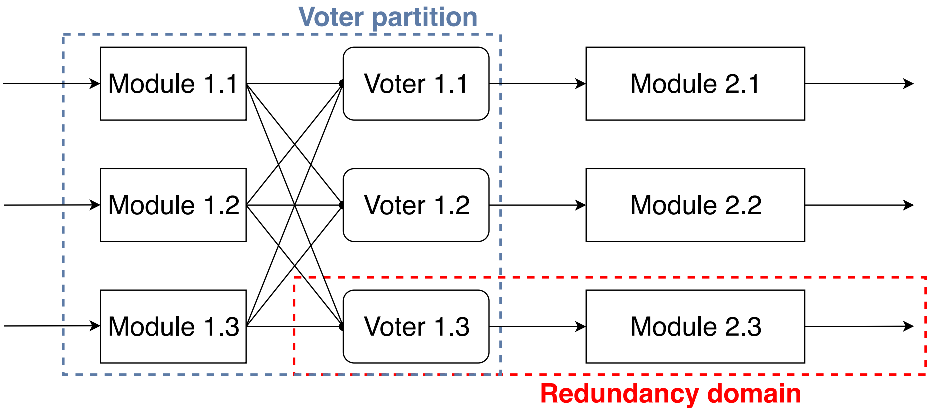


Figure 3.7: Voter triplication and logical areas in TMR

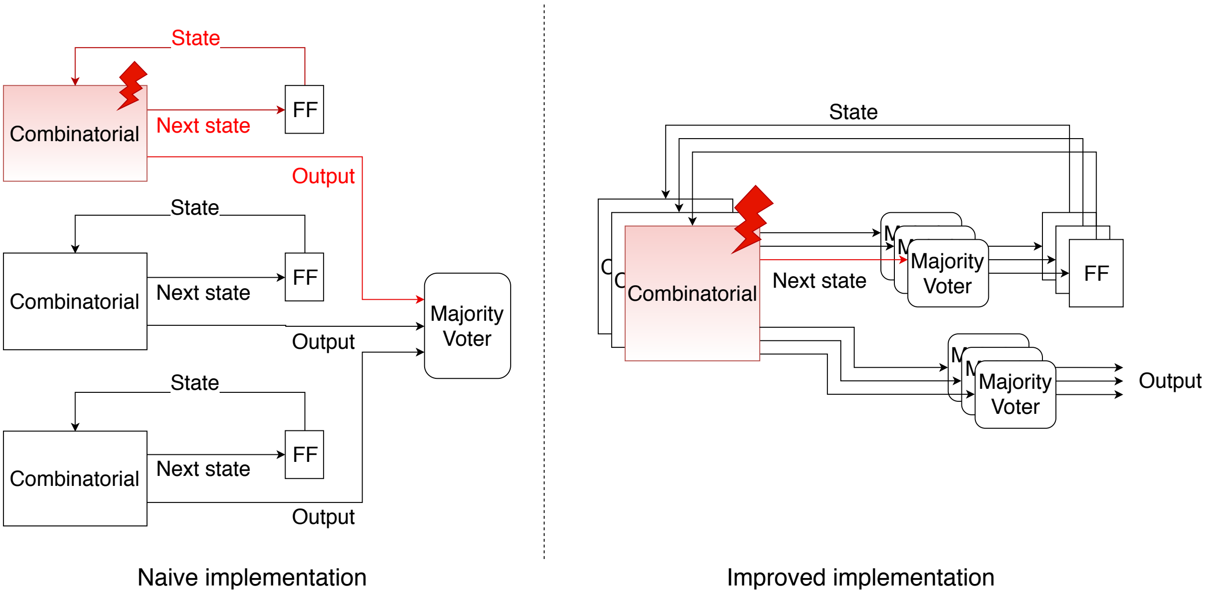


Figure 3.8: Sequential logic with TMR

fully-redundant TMR implementation will also triplicate the voters and the I/O of the FPGA (Figure 3.7). This will increase the resource usage, but not the delay/latency.

One can also identify in Figure 3.7 the **redundancy domain** and the **voter partition**. A redundancy domain can be defined as a logical area within which a soft error will propagate to one and only one of the voter inputs at the end of a partition. As such, a breakdown in the TMR protection occurs when soft errors are present in two domains within the same partition.

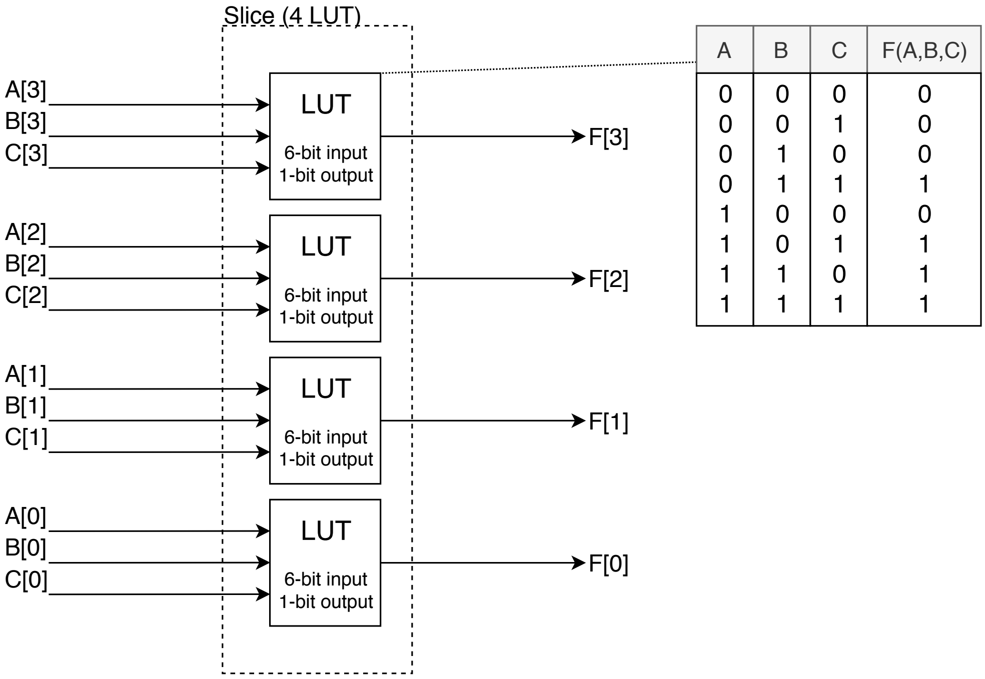
3.3.4 Sequential logic

For sequential logic elements, masking the effects of an upset is slightly more complicated as combinatorial errors can modify the state of the module and become persistent errors, even after the CRAM upset is corrected. The most straightforward approach to solve this problem is to add voters to the feedback loop (see Figure 3.8). With this improved approach, a soft error in the state computation of a single combinatorial block will not result in an erroneous state.

3.3.5 Implementation

Voter implementation

A majority voter is a combinatorial element with a very simple logic function (truth table) (Figure 3.9). In a 7-series Xilinx FPGA, these are implemented in the CLB, which contains two *slices* of 4 LUTs each,



A	B	C	F(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figure 3.9: 4-bit majority voter implementation in a Xilinx logic slice

for a total of 8 LUTs. These LUTs can be configured as either 6-bit input with 1-bit output or 5-bit input with 2-bit output. TMR majority voting is a 3-to-1 operation, therefore only one triplicated signal bit can fit in an LUT. This means that one full CLB can be used to implement an 8-bit TMR majority voter. Figure 3.9 shows a 4-bit majority voter implemented on a single slice.

HDL approach

A pure HDL implementation of TMR is practical only for coarser granularities. A basic example in VHDL is given in Annex C. Note that the synthesis engine might inadvertently break some redundancy protection during the optimization phase. In the Xilinx flow, the attribute *<dont_touch>* can be used to indicate to the synthesis engine that a signal or module should be left as-is. There are two ways one might approach TMR in HDLs, either writing the entity architectures (VHDL nomenclature) with triplication and voting built-in, or writing the modules normally and triplicate them on a higher hierarchical level, this was used in Annex C. The main advantage of the latter is a simplified development process, as TMR can be applied relatively easily, and without complicating the descriptions of the functional modules. The problem comes when describing sequential elements with protected states (Section 3.3.5). As TMR must be applied on the state signals, which are internal to the module, then one would have to explicitly split the description of all Finite State Machines (FSM, i.e. sequential logic) into a combinatorial part and a synchronous part (flip-flops). Although this is often done anyway, this proscribes the use of a completely behavioural-style architecture for sequential logic.
This HDL approach is also limited in other ways:

- Project granularity must match TMR granularity (i.e. TMR-protected modules must be individual entities)
- Common synthesis engines will not optimize for reliability
- Very fine granularity is not practical.

Automated approaches

TMR can also be implemented automatically by both commercial tools (e.g. Synopsys Synplify Premier, Mentor Graphics Precision, Xilinx TMR-tool), or academic tools (e.g. BL-TMR (Brigham Young University)). For compatible models, it can be a very practical way to implement TMR, as projects can be designed more or less normally, and triplication is applied at or after synthesis. By applying TMR

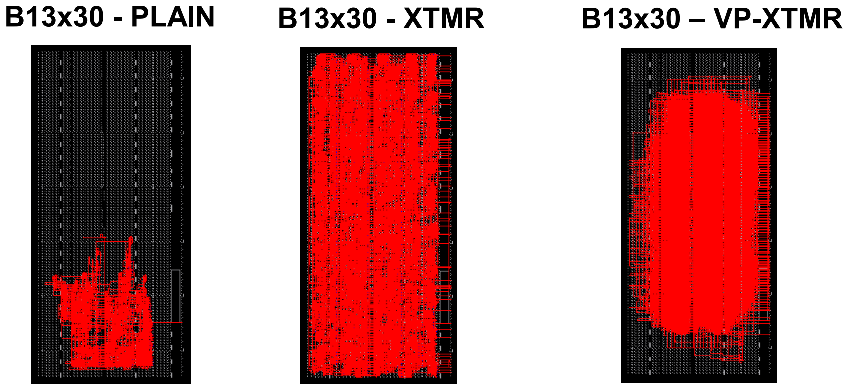


Figure 3.10: VERI-Place placement constraints example [46]

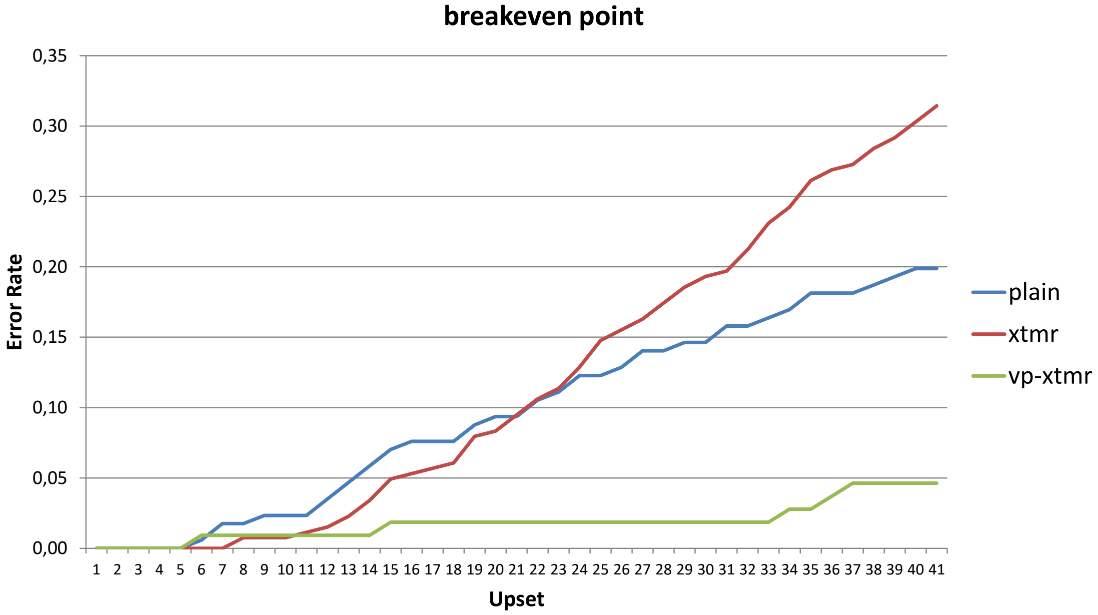


Figure 3.11: Improvement in error rate with VERI-Place constraints [46]

after design optimization (at the netlist level), or through a reliability-focused synthesis engine, these approaches also insure that the intended reliability design is not modified.

Place & Route

As was mentioned earlier, a normal synthesis engine will optimize for performance and resource usage rather than reliability. This can work against TMR, not only because redundant modules can be modified, but also because the placement and routing of the design has a great impact on reliability [44]. For example, redundancy domains within a same partition should not be packed together, and routes to and from these domains should not go through the same routing matrix. Some of these problems can be addressed manually for coarse grain TMR using manual placement, but an automated tool would be useful. Commercial tools targeted for reliability applications may provide some of this functionality (e.g. Synopsis Synplify can do automatic physical separation of TMR modules), but these have not been tested during the course of this thesis. An academic tool called VERI-Place was developed at Politecnico Di Torino, it can provide recommended placement constraints based on an analysis of the design at the development stage (no testing on the device) [46] [13]. It can also provide information such as a list of *critical bits*, which are the bits within the firmware bitstream which will cause a logical error if flipped. This is a subset of the *essential bits* mentioned in Section 2.4.2, which are the bits which will change the circuitry when flipped, but not necessarily cause an error. Critical bits usually comprise between 2 and 10% of the bitstream [42]. Figure 3.10 and Figure 3.11 show the influence of those placement constraints on a design, as advertised by the authors.

3.4 Designing a mitigation solution

As was seen in the previous section, there are many ways of implementing TMR and Scrubbing, but we are limited by the resources available on the GEB. Namely, their resource overhead should be small enough to fit on the FPGA, and the scrubbing must be internal to the FPGA (no secondary processing unit available on the board). These constraints limit the amount of redundancy which can be added, and disqualifies any type of scrubbing based on a golden memory (no rad-hard external memory available).

A few basic scrubbing architectures were presented in Section 3.2, of these only the ECC-based readback scrubbing is possible given the constraints. It was mentioned that the built-in system could be improved with a custom ECC, without mentioning a specific approach. Given the CMS conditions, it seems a priori that the effort required to implement this is not worth the reward, especially given the fact that the scrubbing logic would have to be implemented into the FPGA, which would make it more sensitive to SEUs than the built-in Readback CRC. The scrubbing architecture chosen for this study is therefore the Xilinx SEM IP core, both because it is expected to be sufficient and because implementing and testing a custom architecture would be an research project in its own. Note that in CMS, a global reset of electronic equipment is initiated every 30 minutes, which includes reprogramming the FPGAs.

For the sake of completeness, two experimental methods which improve on the built-in ECC will be briefly mentioned. One proposed method called Frame-level Redundancy Scrubbing consists of exploiting the redundancy in the configuration memory created by a coarse grain grain TMR implementation to detect and repair errors. The idea is that each TMR domain contains the same frame data, therefore any frame of the TMR design can be repaired using the information of the other two identical frames [50]. Another similar approach, which works in parallel with the Readback CRC, uses *erasure codes* to allow MBU correction [40].

TMR can be implemented in multiple ways based on granularity (triplicating a whole pipeline, or each modules individually, or the submodules, etc), voting system (triplicating the voter or not), and place-and-route methods. The TMR design is essential to the mitigation system. An improperly protected design will have many single points of failure (i.e. points where a single SBU is sufficient to break the protection). Unfortunately much of the intricacies of proper TMR design are determined at a lower level than HDL. Indeed the placement and connection of resources will influence the efficiency of the redundancies, and many synthesis optimization will identify and remove parts of the redundancies in an attempt to optimize the design [39] [38]. Tools such as VERI-Place can be used to further improve the efficiency of the redundancies, but it is experimental and not compatible with the Artix-7. For the purposes of this study, TMR will be implemented manually, as the firmware scale is quite small, and it gives the most control over the TMR design, of which several variations will be compared.

BRAM

For the BRAM, similar techniques can be applied: triplication of the memory block, along with a memory scrubbing mechanism. Note that this must be designed separately, as CRAM and BRAM are manipulated differently. ECC correction is available in the Xilinx BRAM modules, but the correction is only done when data is being read. To implement scrubbing, a custom module should be added to read through the BRAM continuously and fix any single errors.

3.5 System reliability

3.5.1 Limitations of the mitigation solution

The combination of TMR and Scrubbing, which we shall call the SEM solution, is an effective scheme, but providing a protection against SEUs is not sufficient to validate the OptoHybrid for use in a radiation environment. The SEM system has limitations, both in the TMR and in the Scrubber, and these limitation entail an imperfect reliability, which must be quantified. Let us first answer the question: what are the limitations of the SEM solution ?

TMR is limited by the fact that it will break if errors are present in more than one redundancy domain within the same partition (recall Figure 3.7). There are two mechanisms by which this can happen: first is the accumulation of soft errors in the device, assuming each particle only causes errors in at most one domain. The second mechanism occurs when this assumption does not hold, i.e. when a single particle event (SEU) causes errors in more than one redundancy domain, this is called a Domain Crossing Error (DCE). To help distinguish between non-DCE which have no impact on the functionality

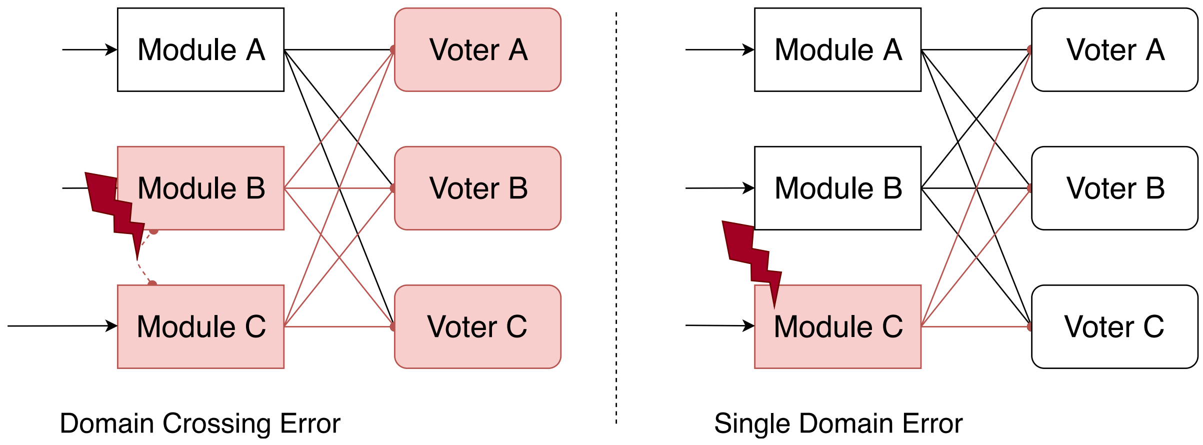


Figure 3.12: Domain Crossing Errors and Single Domain Errors in a voter partition

(non-critical bits), and those who do create an error in a single domain we shall call the latter Single Domain Errors (SDE) (see Figure 3.12). Intuitively, one would link DCEs with MCUs, as increasing the number of flipped bits will increase the probability of affecting multiple domains. It was however found that SBUs can also cause DCEs [44], but it seems from experimental results that this is mostly linked to incompletely implemented TMR (i.e. the presence of single points of failure) [39], and that MCUs are the main source in fully redundant systems. Additionally, it was found that in 75% of cases, DCEs occurred in the routing network [39], making it a very vulnerable area, and reinforcing the notion that place & route is critical to proper TMR design. It was also suggested that there is no correlation between the number of DCEs and TMR granularity [37].

The Scrubber has two limitations: correction delays, and uncorrectable error types. The former impacts the reliability by increasing the probability of SEU accumulation in the device. Upon occurrence of the latter, several situations may occur. The error itself may be a DCE, in which case the SEM has failed, it may be non-correctable SDE which increases the chance of failure, or it may be non-critical. An important fact to note is that if an MBU occurs in *essential bits*, the Xilinx SEM IP Core will stop all detection/correction operations until reconfiguration.

3.5.2 Notions of reliability theory for electronic systems

With the limitations of the mitigation system in mind, models for predicting their reliability over time will be introduced. First however, in order to understand the scope of these models we must take a brief look at a few notion of reliability theory, beginning with a formal definition of reliability.

The *reliability* $R(t)$ of a system at time t is the probability that the system operates without a failure in the interval [0, t], supposing that the system was working at $t = 0$ [16].

The *failure rate* (or instantaneous failure probability) is the expected number of failures per unit time. If one assumes that a system has a constant failure rate λ over its lifetime, then the reliability of the system decreases exponentially with time

$$R(t) = e^{-\lambda t} \tag{3.5}$$

The *Mean Time to Failure* (MTTF) of a system is related to the reliability as follows

$$MTTF = \int_0^\infty R(t)dt \tag{3.6}$$

Therefore if the reliability function obeys the form (3.5), we have

$$MTTF = \frac{1}{\lambda} \tag{3.7}$$

The *Mean Time to Repair* (MTTR) of a system is the average time required to repair the system. It is related to the *repair rate* μ, which is the expected number of repairs per unit time by

$$MTTR = \frac{1}{\mu} \tag{3.8}$$

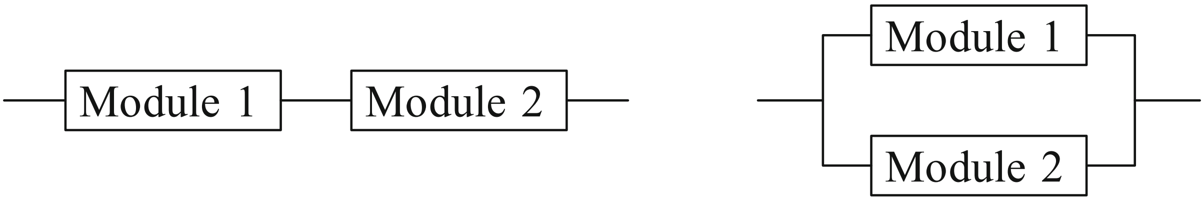


Figure 3.13: Reliability Block Diagram for a series (left) and parallel system (right)

The following sections will propose reliability models for a single TMR module. There might be multiple TMR modules a system, in which case their combined reliability can be modelled using a Reliability Block Diagram (see Figure 3.13) and the following formulas. The blocks are connected in series if all of them are required for the system to work, and in parallel if only one of them is sufficient.

$$R_{series} = \prod_{i=1}^{n} R_i(t) \tag{3.9}$$

$$R_{parallel} = 1 - \prod_{i=1}^{n} (1 - R_i(t)) \tag{3.10}$$

3.5.3 Simple reliability model

For the first, simpler, reliability model, two assumptions must be made

Assumption A: Only one SEU is present in the device at any time

Assumption B: DCE are only caused by MCUs

To verify that assumption A holds most of the time, one needs to verify that *MTTR* $\ll$ *Mean Time to SEU (MTTSEU)*. This can be done by using the latency data from Table 3.1, and

$$MTTSEU = (Rate_{SEU})^{-1} = (\sigma_{SEU} \cdot f)^{-1} \tag{3.11}$$

We will see that this is verified with the experimental cross-section data in Section 5.1.2. With these assumptions, system failure can only come from DCEs, not SDE accumulation. The failure rate therefore equals the DCE rate, which is bounded by the MCU rate. We can then say

$$\lambda = Rate_{DCE} <= P[MCU|SEU] \cdot Rate_{SEU} \tag{3.12}$$

And we therefore have a simple lower bound for the reliability of the system by using (3.5). This approach is simple, but somewhat unrealistic, because it fails to account for the following factors:

1. Some errors are non-repairable (MBUs)
2. SEUs are not guaranteed to always occur after the mean period, there will be some deviation, which means accumulation, though rare, is possible.

For these reasons, we introduce a more complex model, based on Markov chains.

3.5.4 Markov chain model

Simply put, a Markov chain is a stochastic model which is very similar in representation to a state machine, but in which the transition between states is described by an instantaneous probability (see Figure 3.14). We can use a Markov chain to describe all the possible sequences of event between a functional system and a failure.

The Markov chain proposed in this thesis is based upon previous work in [31]. The chain proposed in this paper models SDE accumulation, and repair through scrubbing (see Figure 3.14), concluding that TMR with scrubbing greatly improves the reliability (see Figure 3.15). To this model, it is proposed now to add the effect of DCEs, whose rate shall be noted ϵ, as well as two non-correctable states, in which it is still possible for the system to fail, but no more repair can be done. This models the fact that the

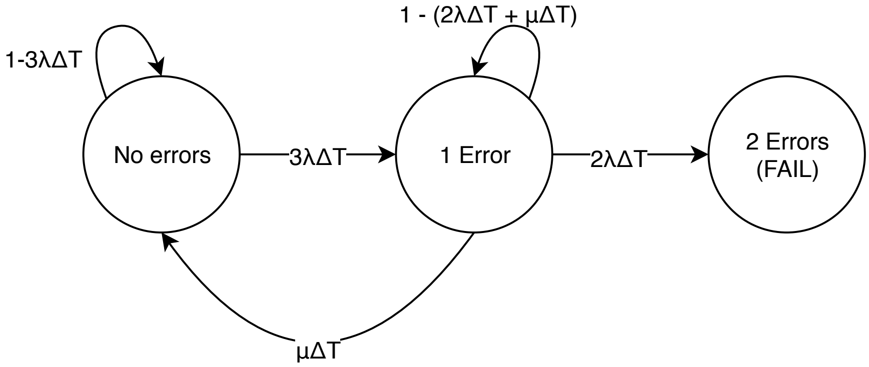


Figure 3.14: Markov chain representing TMR with repair through scrubbing

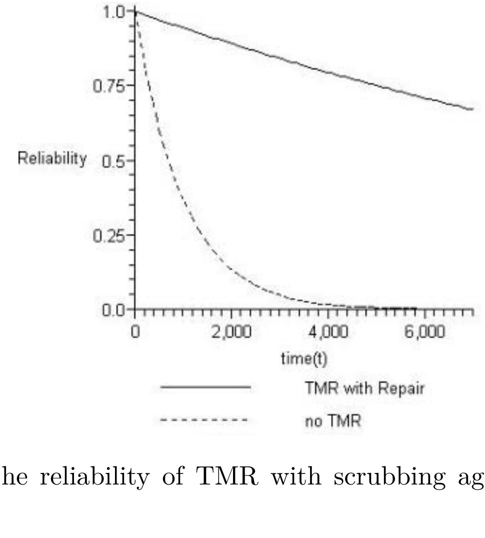


Figure 3.15: Comparison of the reliability of TMR with scrubbing against no TMR. $\lambda = 0.001$ and $\mu = 0.1$ [31]

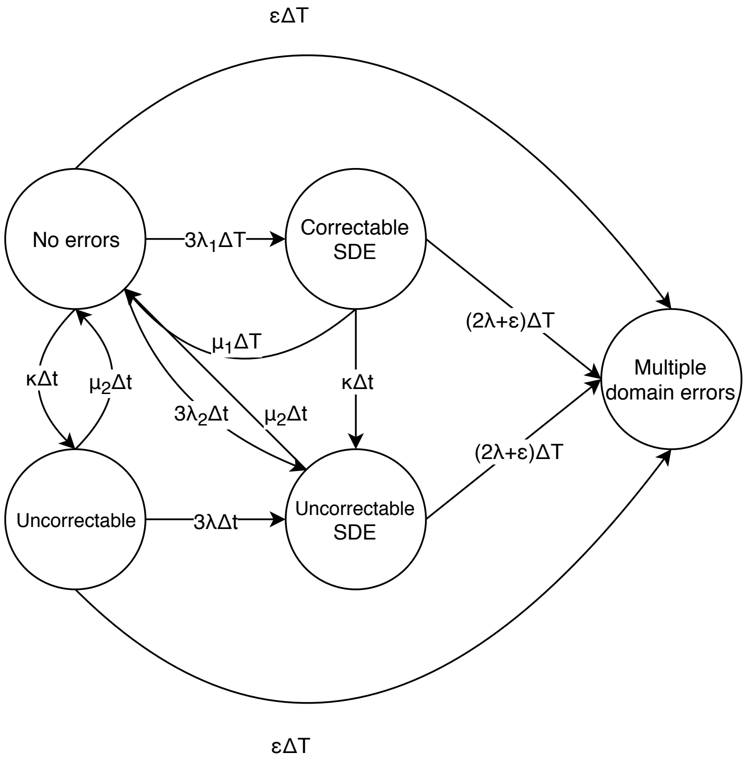


Figure 3.16: Proposed Markov chain model for the mitigation system in a single TMR module

Xilinx SEM IP Core will stop all scrubbing activity upon detection of an (essential) MBU. Recovering from this state is done by reconfiguring the FPGA, which occurs every 30 minutes in CMS operation. This is modelled by a MTTR $\mu_2^{-1} = 30\,\text{min}$. The "Uncorrectable" state represents an MBU somewhere in the device but not in one of the redundancy domains of the TMR module. The "Uncorrectable SDE" represents either an MBU causing an SDE, or an MBU somewhere in the device along with an SDE. This improved model is represented in Figure 3.16, where the probabilities of staying in the same state have been omitted for clarity.

The process for solving this Markov chain can be found in Annex D. The analytical solution is extremely complex due to the number of parameters in the model, therefore the solutions will be found computationally using the transition matrix T.

$$\begin{pmatrix} P_0'(t) \\ P_1'(t) \\ P_2'(t) \\ P_3'(t) \\ P_4'(t) \end{pmatrix} = \underbrace{\begin{bmatrix} -(3\lambda+\kappa+\epsilon) & \mu_2 & \mu_1 & \mu_2 & 0 \\ \kappa & -(3\lambda+\mu_2+\epsilon) & 0 & 0 & 0 \\ 3\lambda_1 & 0 & -(\mu_1+2\lambda+\epsilon+\kappa) & 0 & 0 \\ 3\lambda_2 & 3\lambda & \kappa & -(\mu_2+2\lambda+\epsilon) & 0 \\ \epsilon & \epsilon & (2\lambda+\epsilon) & (2\lambda+\epsilon) & 0 \end{bmatrix}}_{T} \cdot \begin{pmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{pmatrix} \tag{3.13}$$

The state numbers correspond to "No errors", "Uncorrectable", "Correctable SDE", "Uncorrectable SDE" and "Multiple Domain Errors", respectively. $P_i(t)$ corresponds to the probability of being in state i at time t, starting from the initial conditions. Reliability is therefore given by

$$R(t) = 1 - P_4(t) \mid P_0(0) = 1 \tag{3.14}$$

The functional error rates of the Markov model can then be related to measurables quantities through the FPGA's SEU cross-section, and the following ad-hoc parameters, measurable for a specific module design.

$$
\begin{aligned}
\lambda_1 &= Rate_{\overline{MBU}} \cdot P(SDE|\overline{MBU}) \\
\lambda_2 &= Rate_{MBU} \cdot P(SDE|MBU) \\
\lambda &= \lambda_1 + \lambda_2 \\
\epsilon &= Rate_{SEU} \cdot P(DCE|SEU) \\
\kappa &= Rate_{MBU} \cdot (1 - P(SDE|MBU)) \\
\mu_1 &= {MTTR_{SEM}}^{-1} \\
\mu_2 &= 30\,\mathrm{min}^{-1}
\end{aligned}
$$

With $\overline{MBU}$ begin the set complement of the MBU type, i.e. SBU and inter-frame MCU, we therefore have $\lambda = Rate_{SEU} \cdot P(SDE|SEU)$. These ad-hoc parameters could be found through beam testing or error injection (see Section 5.1.4). With error injection, one could find, for a given module design, parameters relating DCE and SDE probability to each specific type of SEU (e.g. 3-bit MCU, 2-bit MBU, etc, see Section 5.1.1), and then extrapolate the ad-hoc parameters for different radiation conditions by using the cross-section data and MCU statistics of these conditions (see Section 5.1.2). For this thesis, parameters specific to an experimental firmware will be found following a beam testing campaign.

Chapter 4

Experiments

4.1 Beam testing campaign

To obtain data on the response of the Artix-7 to neutrons, a beam testing campaign was carried out at the UCLouvain's cyclotron facility. The facility is equipped with a proton beam line capable of accelerating protons up to 62 MeV, which were selected as a neutron analogue considering their very similar SEU cross-sections [52]. Various tests were carried out during 10h of beam time, which will be presented in the following sections. The preparation of the experimental protocol, software, hardware and firmware required to carry out the tests composed a large part of the work undertook for this thesis.

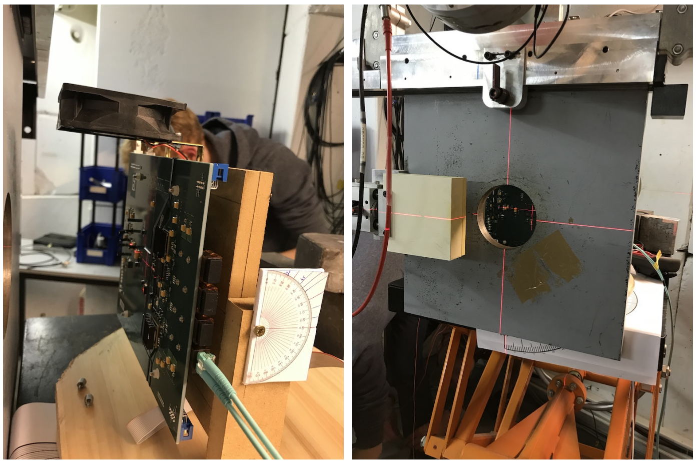


Figure 4.1: Experimental setup of the OptoHybrid during the beam testing campaign

The installation used for the tests is shown in Figure 4.1. The energy of the protons could be adjusted to several discrete levels by placing attenuators between the beam and the device. The OptoHybrid GE2/1 board was aligned such that the FPGA was at the center of the beam. A lead beam mask was placed in front of the FPGA to limit the irradiation of the other board components. The homogeneity of the beam has been measured by the facility at ±10% over 80mm in the horizontal and vertical directions, the FPGA package dimensions being 23x23mm.

During the tests, the device was placed in the isolated beam chamber and the research team stayed in a nearby control room protected by a concrete wall through which as small opening allowed cables to go from the control room to the Device Under Test (DUT).

4.2 SEU cross-section test

4.2.1 Motivation

Predicting the reliability of an electronic system in a radiation environment requires knowing how this environment interacts with the system. As we have established in Chapter 2, the principal source of concern for the FPGA in the OptoHybrid board are SEUs in the Configuration Memory (i.e. LUTs,

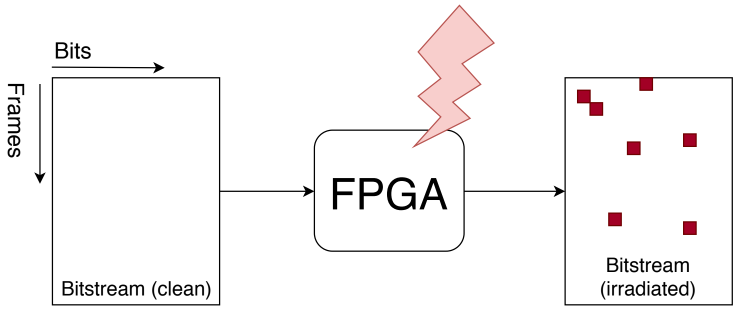


Figure 4.2: Bitstream readback after irradiation

routing matrices, etc), and in the Block Memory. The goal of this test is to quantify the cross-section values for SEUs in the Artix-7, and to produce as much information as possible on MCUs which will be particularly problematic with the proposed mitigation system.

4.2.2 Principle

The experimental protocol for this test is based in large part by the work in [57]. The simplest scheme for finding the SEU cross-section of an FPGA is to load it with a known bitstream, expose it to radiation at a determined fluence, read the bitstream back from the FPGA and compare the two (called a *static* test, see Figure 4.2). The cross-section can then be calculated from Eq. 2.3. The problem with this method is that an MCU in the CRAM will produce multiple bit upsets, and a simple comparaison of the clean and irradiated bitstream will count those as multiple upsets. Thus we can only obtain the 1-bit upset cross-section, which will always be higher than the SEU cross-section if we define an SEU as the effects caused by a single particle.

Given the size of the FPGA Configuration Memory (24 Mbits for the XC7A75T model used in the OptoHybrid GE2/1), it can be expected that if the total number of upsets is relatively low, the probability of two *independent* upsets being physically close together is low as well. The problem is that the relation between the physical layout of the device and a position in the bitstream is not known. The approach proposed in [57] is to first perform a statistical analysis of the irradiated bitstream, identifying frequently occurring upset patterns, which are hypothesized to be indicative of physical proximity. This hypothesis was validated by comparing the results from the proposed method with those obtained using proprietary information for the Virtex-5. The *physical adjacency model* created can then be used to extract MCUs from the irradiated bitstream.

4.2.3 Experimental architecture

The experimental architecture for this test is represented by the diagram on Figure 4.4. The Artix-7 was loaded with a very basic firmware, instantiating the CLBs and BRAM primitives. It was connected through a USB-JTAG adapter to a computer in the control room, from which the firmware could be sent to and read from the DUT using the Xilinx Vivado software suite. In parallel, an Arduino microcontroller board was used to read data from 3 shunt current sensors monitoring the 1.5V, 2.5V and 1.8V lines of the OptoHybrid. The USB-JTAG adapter was placed next to the OH in the irradiation chamber, protected by several blocks of lead.

Using this architecture, small irradiation runs were carried out, with varying proton energy levels and angles of incidence. The amount of upsets per run was limited to avoid the creation of false MCUs. A simulation was carried out in [57] estimating the probability of false positives depending on the total amount of upsets in the bitstream (Figure 4.3, left). Additionally, a simulation was run prior to the tests estimating the number of upsets occurring more than once in the same cell in an irradiated bitstream depending on the total amount of upsets, and assuming a homogenous probability density for SEU location (Figure 4.3, right). Based on this data, a maximal amount of 2000 upsets per run was chosen as a compromise between practicality during the tests and the risk of false positive (8.33×10^{-5} percent of the CRAM, see Table A.1). The fluence per run was chosen to reach ≈ 2000 upsets based on the data in Section 2.6, and was manually adjusted to $1.5 \times 10^{10}\,\mathrm{cm}^{-2}$ during the tests.

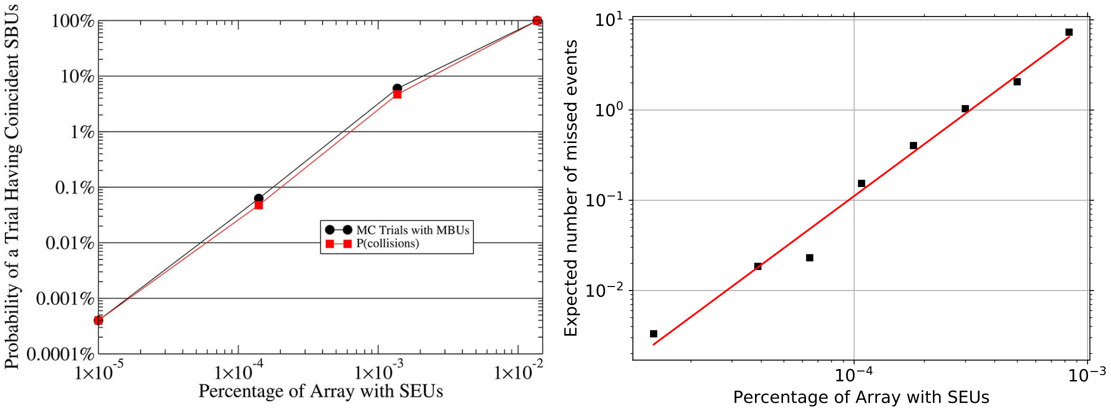


Figure 4.3: Simulation relating the amount of SEUs in an irradiated bitstream with the probability of false MCUs (left) [57] and missed events (right).

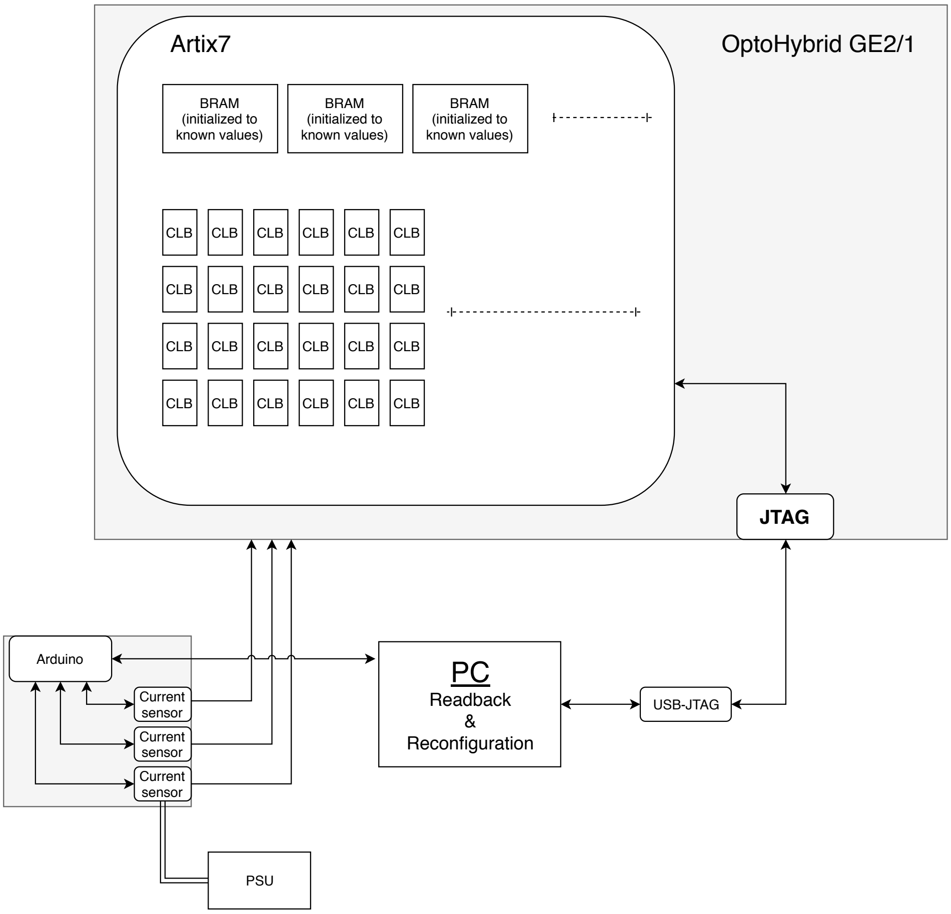

Artix7
OptoHybrid GE2/1
BRAM (initialized to known values)
BRAM (initialized to known values)
BRAM (initialized to known values)
CLB
JTAG
Arduino
Current sensor
Current sensor
Current sensor
PSU
PC
Readback & Reconfiguration
USB-JTAG

Figure 4.4: Experimental architecture for the cross-section static test

Parameter	Data points			Runs per point	Target upset count
Energy	25.8 MeV	43.9 MeV	62 MeV	5	10,000
Horizontal tilt	0°	30°	45°	3	6,000
Vertical tilt	0°	30°		3	6,000

Table 4.1: Parameters for the cross-section test

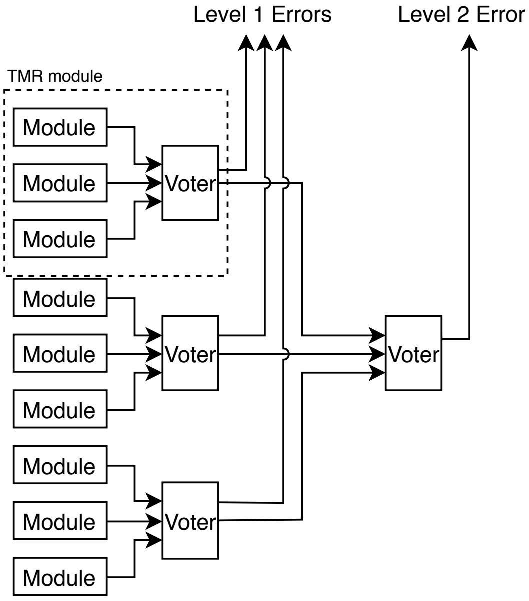


Figure 4.5: Two-level TMR

4.2.4 Parameters

Two parameters affecting MCU proportion were identified in Section 2.6: particle energy and angle of incidence. Additionally, the data from GE1/1 tests showed a clear influence of proton energy on the SEU cross-section. Because the angle of incidence can either be along or across the memory cell columns of the FPGA, the DUT was tilted vertically and horizontally, to identify which of the two is the critical orientation (see Figures 4.1 and 1.14 to identify the tilting directions relative to the board).

The energy levels chosen for the tests were based on the results obtained in the GE1/1 tests. A sharp drop in cross-section was expected around 20 MeV therefore the range was chosen from 25.8 MeV to the maximal energy of 62 MeV. The parameters of the test are summarized in Table 4.1. The number of runs for the tilt experiments were reduced to accommodate all the desired data points following delays during the tests, the nature of which will be outlined in Chapter 5.

4.3 Mitigation system test

4.3.1 Motivation

In Chapter 3, a solution for mitigating the effects of SEUs in the CRAM was proposed, within this solution is the well-established Xilinx SEM IP core, which has been tested extensively and for which much information is available in the datasheet. It is taken to be functional as-is. The second leg of the mitigation system is TMR, it is also a well-established solution but it can be implemented in several ways depending on e.g. granularity, placement constraints, and voter triplication. To better understand the impact of these parameters, several implementations of TMR should be compared.

In addition, these tests can provide information for predicting the reliability of the mitigation system using the models proposed in Section 3.5. For the simple model, the validity of Assumption B can be tested. For the Markov chain model, the ad-hoc parameters can be estimated.

4.3.2 Principle

A TMR module can be in one of three states: no errors, error in one output, error in two outputs (redundancy breakdown). Detecting one error is simple, the majority voter can output an error flag if only two of three outputs match. Detecting a redundancy breakdown is harder, the solution proposed for the test is to use a two-level redundancy, such that an error in the first level can be detected by the majority voter of the second level (see Figure 4.5).

The TMR module itself will have multiple implementation types, and the value of the flags will be extracted in real-time during the irradiation of the device. An L1 error corresponds to an SDE, and an L2 error corresponds either to an accumulation of SDEs, or a DCE. Of the two, the information of interest is the occurrence of a DCE (the accumulation of SDEs can be predicted from the SDE probability and the SEU rate). To avoid SDE accumulation, there should only be one SEU in the system at any time. The Xilinx SEM IP Core will be used to scrub correctable errors, and the particle flux will be set low enough to allow ample correction time. On occurrence of an uncorrectable error, the *uncorrectable* status flag of the SEM IP Core will be set, which will be used to trigger a reconfiguration of the FPGA.

It is impractical to use modules from the development firmware of the OH GE2/1 FPGA (e.g. trigger logic) as their ressource footprint is quite high. This would greatly limit the amount of triplicated modules which can be placed in the experimental firmware, and therefore the amount of variations which can be tested. Although one cannot assume the ad-hoc parameters extracted from the experimental firmware would also apply to a different circuit design, they are still valuable to get a sense of what could be expected.

To test the validity of Assumption B, one should observe that DCEs occur only with an MCU (or the vast majority of the time). To associate DCEs with an SEU type, the status flags of the SEM IP Core will also be extracted in real-time. The *uncorrectable* flag will trigger for an MBU, while an inter-frame MCU will trigger the *correction* flag multiple times in short succession (with a maximum delay equal to (3.4)).

4.3.3 Experimental architecture

The experimental architecture for this test is represented by the diagram on Figure 4.6. The Artix-7 was loaded with a firmware composed of three component types: the SEM IP Core, the communication systems (GBT Link Multiplexer + PISO), and 24 TMR sectors. A single TMR sector is composed of 2 levels of some variation of TMR (see Section 4.3.4), as in Figure 4.5. Each sector has 4 output signals, three L1 flags and one L2 flag. The communication between the computer and the DUT was done through the Gigabit Link Interface Board (GLIB), an evaluation platform developed by CERN containing a Virtex-6 FPGA. The GLIB generates the master clocks, and also handles the (re)configuration of the OH through what is called PROMless configuration, using a copy of the OH bitstream stored in an on-board SRAM. The reconfiguration procedure is triggered from the computer, and is completed in a few milliseconds through the SelectMAP interface. Both the GLIB and the computer were placed in the control room.

Data from the OH is sent to the GLIB at 320Mbps, which the GLIB then buffers in its BRAM. The computer extracts the raw data from the BRAM through the IPBus protocol over Ethernet. Although it was not designed for data transfer, this protocol was developed for use with the GLIB, which makes it convenient to use. The specifications indicate that the throughput is sufficient, provided the data is extracted in large batches [24]. For this reason, the GLIB will buffer all the data it receives from the OH FPGA into a large FIFO memory, which will then be emptied into the IPBus packets. Because IPBus was not designed for this purpose, some tricks had to be employed to insure the GLIB buffer was never overflowing. The data pre-processing was done in software, it consisted of logging any change in the DUT internal signals, timestamping the data, and triggering a reconfiguration if an uncorrectable SEU was detected.

The architecture of the basic low-granularity TMR module is represented on Figure 4.7 (left). The base module is purely combinatorial; each submodule (CombX) is composed of 7 CLBs, and is designed to maximize the chance of obtaining a logical error from an SEU (to get as much data as possible). This required cycling the inputs of the TMR modules, which was done by using a 6-bit counter for each sector (64 possible inputs for the 6-bit LUTs of the CLBs). The internal architecture of the CombX is represented in Figure 4.7 (right). The 48-bit input vector of each CLB is simply a repetition of the 6-bit counter, each going to one LUT. By using 6 CLBs per layer, one has 48 LUTs with 6-to-1 logic functions, and therefore the output of that layer can be used to feed another CLB or layer of CLBs. For these tests only one layer was used. As with the static test, an Arduino was monitoring the currents.

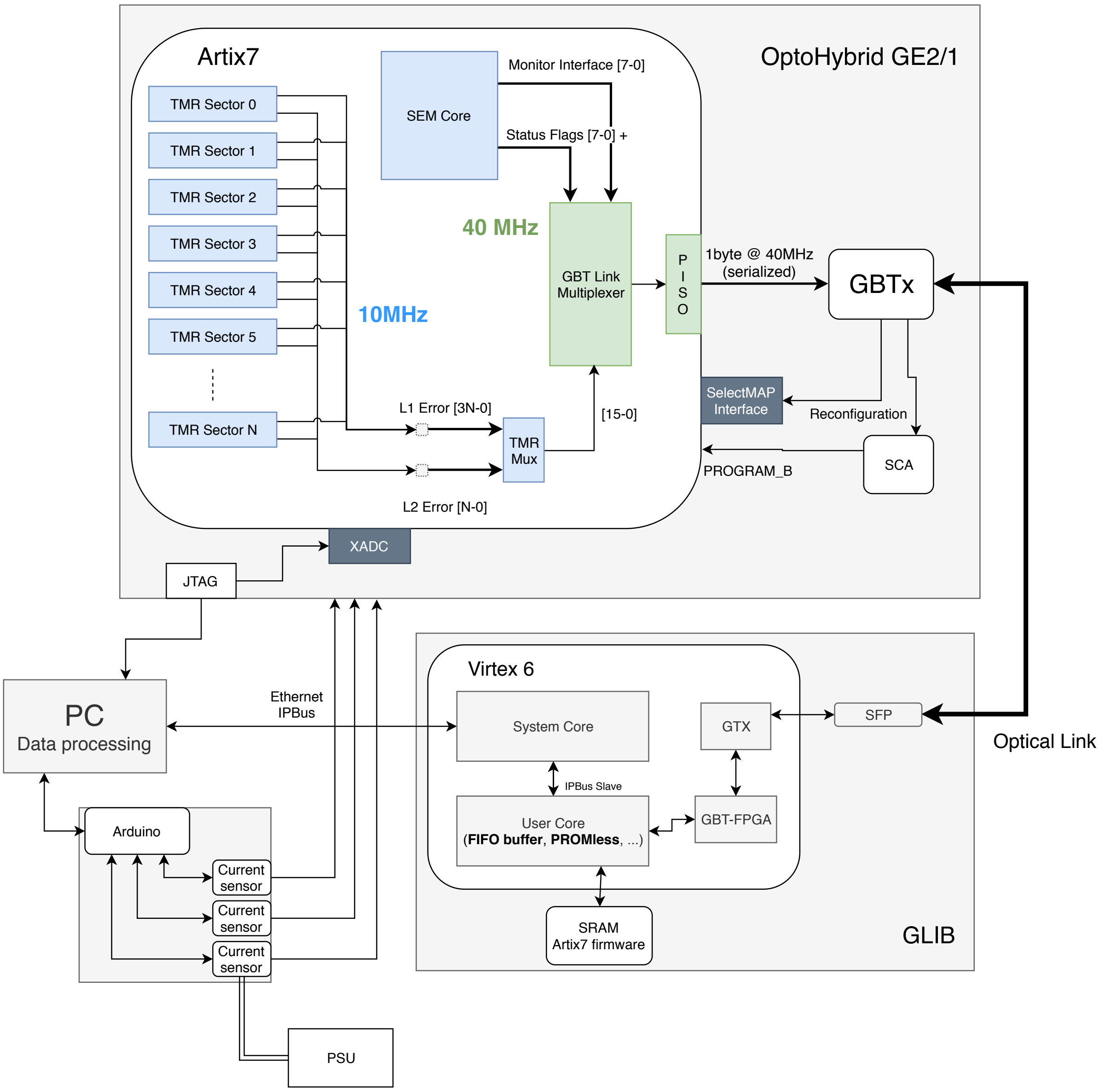


Figure 4.6: Experimental architecture for the mitigation system test

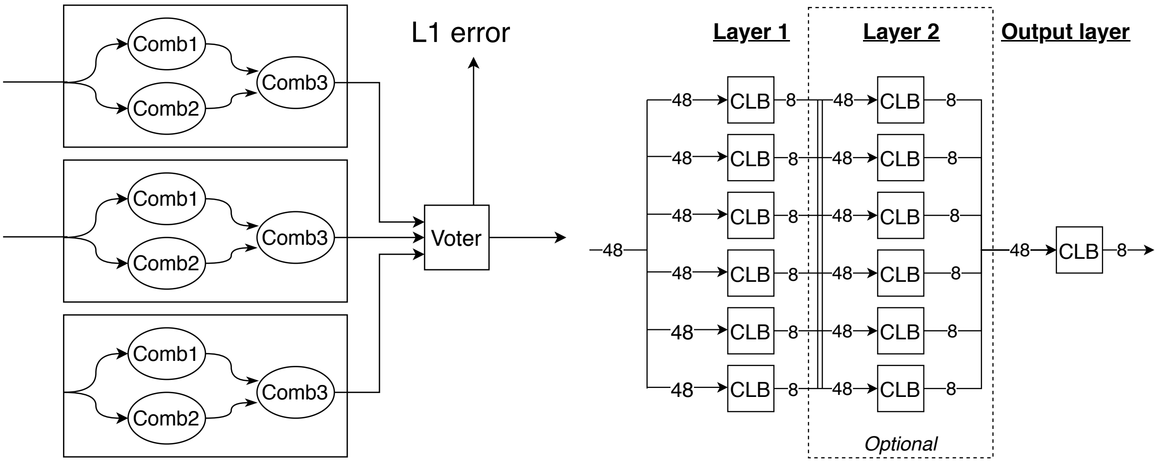


Figure 4.7: Coarse granularity TMR module (left), and CombX submodule architecture (right).

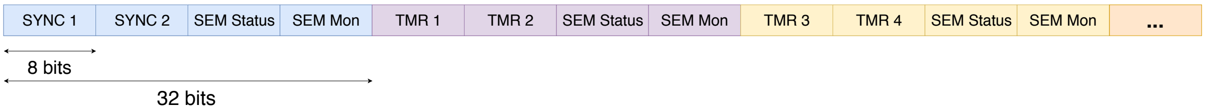


Figure 4.8: Data stream leaving the OH FPGA

Data extraction protocol

Information extracted from DUT:

- **Status flags (8 bits)**
  - status_heartbeat
  - status_initialization
  - status_observation
  - status_correction
  - status_classification
  - status_injection
  - status_essential
  - status_uncorrectable
- **Monitoring data (8 bit-wide ascii stream)**
  - Error detection reports (single/double error, logical location, ...)
- **TMR sectors (24 sectors, 192 bits total)**
  - L1 flags (3x2 bits per sector)
  - L2 flag (1x2 bits per sector)

The communication between the OH FPGA and the GLIB is done over optical link through the rad-hard GBTx chip on the OH and the GBT-FPGA core in the GLIB (see Section 1.3.1). The link between the OH FPGA and the GBTx is a single bit signal running at 320MHz, which effectively gives an 8-bit bus from the point of view of the FPGA which runs at 40MHz. In order to minimize the risk of data corruption, no processing is done in the irradiated FPGA, and the raw signals are all sent through the 320Mbps link at the native clock frequency. The problem here is obvious, as the data link is 8-bit wide and the data to be sent is much larger. There are 24 TMR sectors, with 4x2 bits per sector (the flags are duplicated for data redundancy), 8 bits for the SEM status interface, and 8 bits for the SEM monitoring interface. To solve this problem, two types of time multiplexing are used. First, all the non-communication systems of the OH FPGA are run at 10MHz, which quadruples the effective bus width to 32 bits. Second, one can note that the sector flags will update much more slowly than the SEM. Indeed, in case of an L1 error, the flag will only disappear once the SEU is corrected. The average delay for this to occur is given by Table 3.1, and it is much slower than a 10MHz clock period, even when taking into account the 64 possible inputs to every TMR module which must all be sampled. With this knowledge in mind, a second time multiplexing is done by scanning the sector flags. Within this sequence, a 16 bit synchronization word will replace the TMR data at the beginning of every data frame in order the align the data at the receiving computer, a data frame containing one pass of the scan through all the sectors. The result of this is a data stream illustrated in Figure 4.8. With the scanning scheme in place, the TMR input counters are incremented after each complete scan, which insures that the sector flags for all possible inputs are sampled.

4.3.4 Parameters

The TMR modules are all based around the same combinatorial logic module from Figure 4.7 and declined in three versions, each replicated 8 times in the firmware, for a total of 24 sectors:

- Coarse granularity (Figure 4.7)
- Fine granularity, without voter triplication (Figure 4.9)
- Fine granularity, with voter triplication (Figure 4.9)

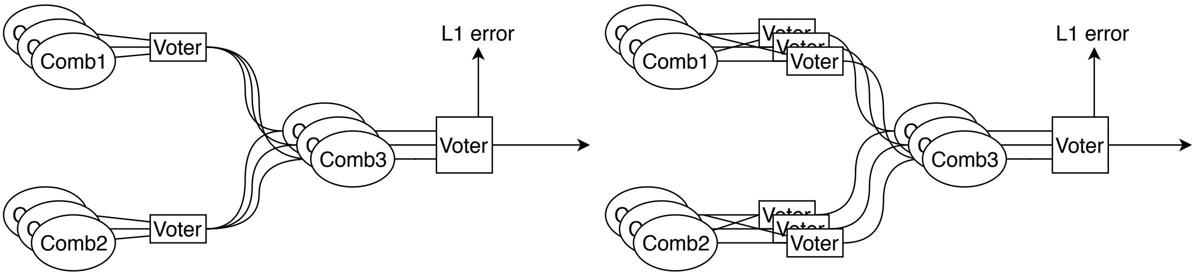


Figure 4.9: Fine granularity TMR module, with (right) and without (left) voter triplication

The layout of the TMR sectors in the FPGA in shown on Figure 4.10. Half of the TMR sectors were placed fully automatically within their designated areas, the other half were placed partly manually, trying to physically separate the redundancy domains in the voter partitions. The goal was to estimate what basic placement constraints could do to reduce the probability of DCE occurrence.

The proton energy was fixed at 62MeV during this test to maximize the rate of MCUs, in order to get as much L2 data as possible. The flux was chosen based on the Xilinx cross-section data from Figure 2.14 and by fixing the derivative of equation (2.3) to $1\frac{SEU}{s}$. This gives a flux of 5 MHz, but it was found to give a lower rate of SEUs than expected, possibly due to SEUs in unused design resource which are not scanned by the SEM IP Core [42]. The flux was then tuned up during the experiment to get the desired rate.

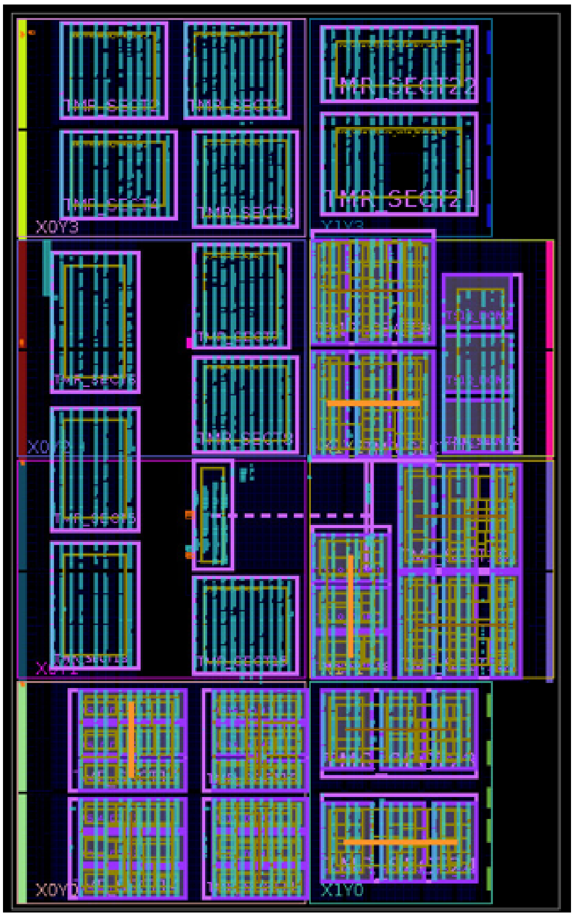


Figure 4.10: Physical layout of the mitigation system test firmware

Chapter 5

Data analysis

At the conclusion of the beam testing campaign, the ample amount of raw data obtained from the tests went through a long process of analysis. The static cross-section test ran through as expected, and yielded irradiated bitstreams from which MCUs could be extracted using the statistical analysis method mentioned in Section 4.2.2. As for the mitigation system tests, results were obtained, although some problems occurred which made data extraction more difficult, as well as reducing the overall amount of data gathered. The nature of the problems will be explained in Section 5.2.1, followed by the results, along with simulations of the Markov chain model presented in Section 3.5.4.

5.1 SEU data

5.1.1 MCU extraction

The MCU extraction scheme proposed and tested in [57] will now be explained in more detail. The following procedure will be an overview of the steps required to extract MCUs from an irradiated bitstream whose physical layout correspondance is unknown, a more complete description is available in the paper for the interested reader.

1. Use the bitstream mask generated by the development tools to mask out the non-CRAM, dynamic areas of the bitstream.

2. Arrange the clean and irradiated bitstreams into 2D arrays of dimensions such that the coordinates of an upset (u_i) in the bitstream is represented by a (x_i, y_i) tuple, where x corresponds to the frame number location and y the bit number within the frame.

3. Determine the logical distances between each upset pair in this array and create an histogram of common upset pairs.

4. Create the physical adjacency model (PAM) from this statistical data

5. Extract MCUs from the irradiated bitstream using the physical adjacency model.

For the XC7A75T model, the number of frames in the bitstream is 9465, with 3232 bits per frame. The last step implies returning to the irradiated after calculating the PAM, iterating over all the upsets from first to last, and for each reference upset, look for any upsets *after it* which fit in the PAM. The PAM calculated from our data is reproduced in Figure 5.1. The model is symmetrical as any upset pair can be taken from the reference point of one upset or the other. One can note that the most frequent pair is the (1,-1) diagonal pair, which is an inter-frame MCU, and therefore correctable by the built-in ECC. The second most frequent pair is the (0,-1), which is an MBU and therefore not correctable.

For every data point in Table 4.1, an histogram of the most frequent extracted MCU shapes was plotted. The histogram at 62 MeV, with 0° angle of incidence is shown on Figure 5.2. The complete set of histograms is reproduced in Annex B. From these histograms, four measurements can be computed for the CRAM: the upset cross-section (1 upset = 1 bit-flip), the SEU cross-section, the MCU proportion, and the proportion of inter-frame MCUs among all MCUs.

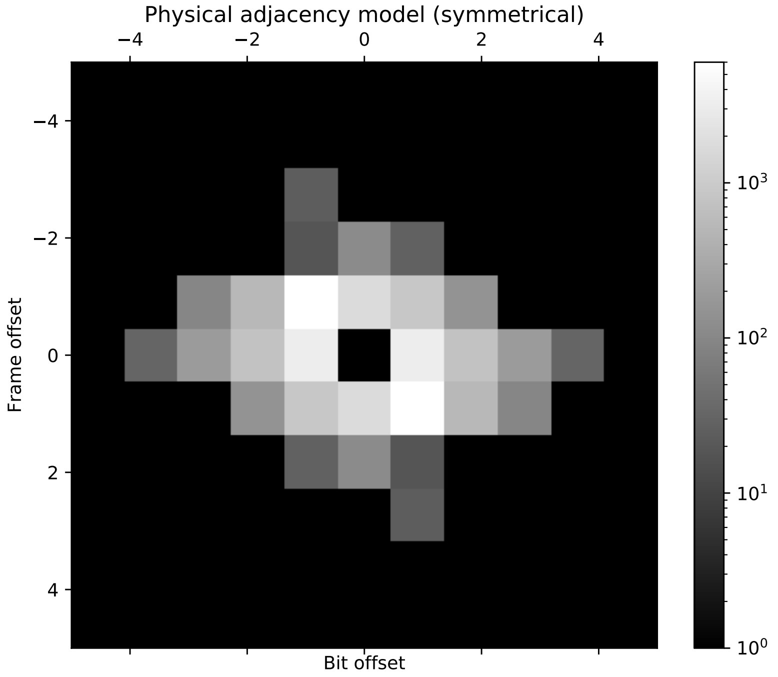


Figure 5.1: Physical adjacency model extracted from the irradiated bitstreams

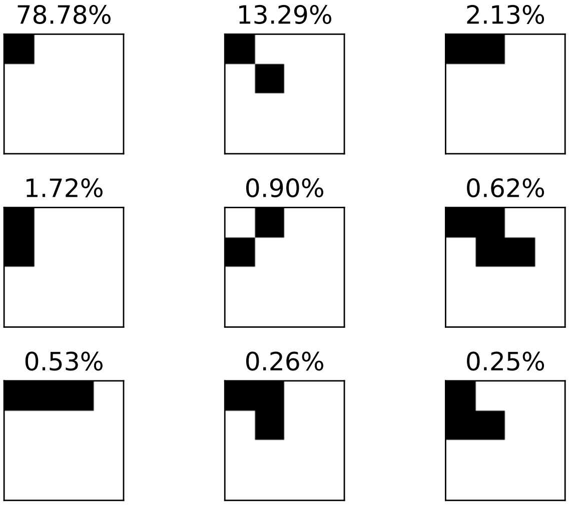


Figure 5.2: Histogram of the most frequent MCU shapes at 62 MeV, 0° angle of incidence

5.1.2 SEU cross-sections and MCU statistical data

The full dataset of the CRAM upset cross-section, CRAM SEU cross-section, MCU proportion, and proportion of inter-frame MCUs relative to energy, vertical angle of incidence and horizonal angle of incidence is given in Annex B. A selection of which is reproduced in Figure 5.3. The per-device cross-sections are the values obtained for the whole bitstream. The per-bit cross-sections are normalized by the size of the bitstream to allow meaningful comparison with other FPGA models. The following observations can be made:

- The per-bit SEU cross-section is comparable with the values seen in Section 2.6. At 48.9 MeV, we found a normalized upset cross-section of $6.0 \times 10^{-15}\,\mathrm{cm}^2\ \mathrm{bit}^{-1}$, about 25% lower than the results found for the Kintex-7 at 35 MeV (see Tables 5.1 and 2.2). Note that in the Kintex-7 results, the cross-section was found to reduce for energies above 35 MeV.
- The SEU cross-section does not vary much with energy in the tested range, including at the 25.8 MeV level where a large drop was expected. Although variations are to be expected with energy, it is likely that the tested range was simply too narrow to observe them.
- The MCU proportion rises with both energy and horizontal angle of incidence, the latter having a high of around 27% at (62MeV, 45°), up from 21% at (62MeV, 0°) . The vertical angle of incidence does not influence MCU proportion, indicating this direction is across the memory cells columns.
- Along with every increase in MCU proportion, a decrease in the inter-frame type proportion is noted. This translates to a larger number of non-correctable MCUs.
- Despite not being normalized by a $\frac{1}{cos(\theta)}$ factor, the SEU cross-section relative to the angle-of incidence remains stable. This is counter-intuitive as the cross-sectional area (N.B. not the cross-section) of the FPGA is smaller, i.e. fewer particles interact with it. An explanation can be hypothesized to be that as particles are traversing more memory cells due to the tilt, they are more likely to create an upset. It would appear from our data that in the tested range, this effect balances the reduction in cross-sectional area.
- Because the MCU proportion rises and the SEU cross-section with horizonal angle of incidence remain stable, the upset cross-section rises.

5.1.3 SEU location probability density function

From the irradiated bitstreams of every data point, a normalized binned histogram of the SEU locations was created (20 by 7 bins for a bitstream of dimensions 9464 by 3232). An interpolated version of the result for the data points (62 MeV, 0°), (62 MeV, 30° horizontal), (62 MeV, 30° vertical) and (48.9 MeV, 0°) is shown on Figure 5.4. The PDFs are mostly homogenous, the dark areas are zones which were masked out of the bitstream. Small hotspots are visible in some of the figures above, but it does not seem to be a trend in the data. The (62 MeV, 45° horizontal) has no significant hotspots, and neither does the (25.8 MeV, 0°). Consequently they are likely only statistical noise.

5.1.4 Proposed error injection scheme

Using the data presented here, one could inject into a bitstream identical SEUs to the ones obtained during our tests, in the same proportions, and with the same shapes. Given that the SEU location PDF was observed to be mostly homogenous, the location of each injected SEU can be chosen with a uniform random number generator. For the purposes of functional testing (e.g. finding the ad-hoc parameters of the proposed Markov chain reliability model) this could be a useful alternative to beam testing, as the controlled environment of error injection allows much more data to be extracted about the internal signals.

5.1.5 Conclusions

The cross-section data obtained during the static testing phase fits with previous results, which bolsters the credibility of the rest of the data obtained. It is clear that the drop in SEU cross-section expected around the 20 MeV mark was not visible at 25.8 MeV. As it is, the data obtained during these tests, along with existing data in the 100 and 200 MeV ranges is not enough to guarantee an accurate prediction for the amount of SEUs as the neutron energy distribution in CMS extends well below 25.8 MeV.

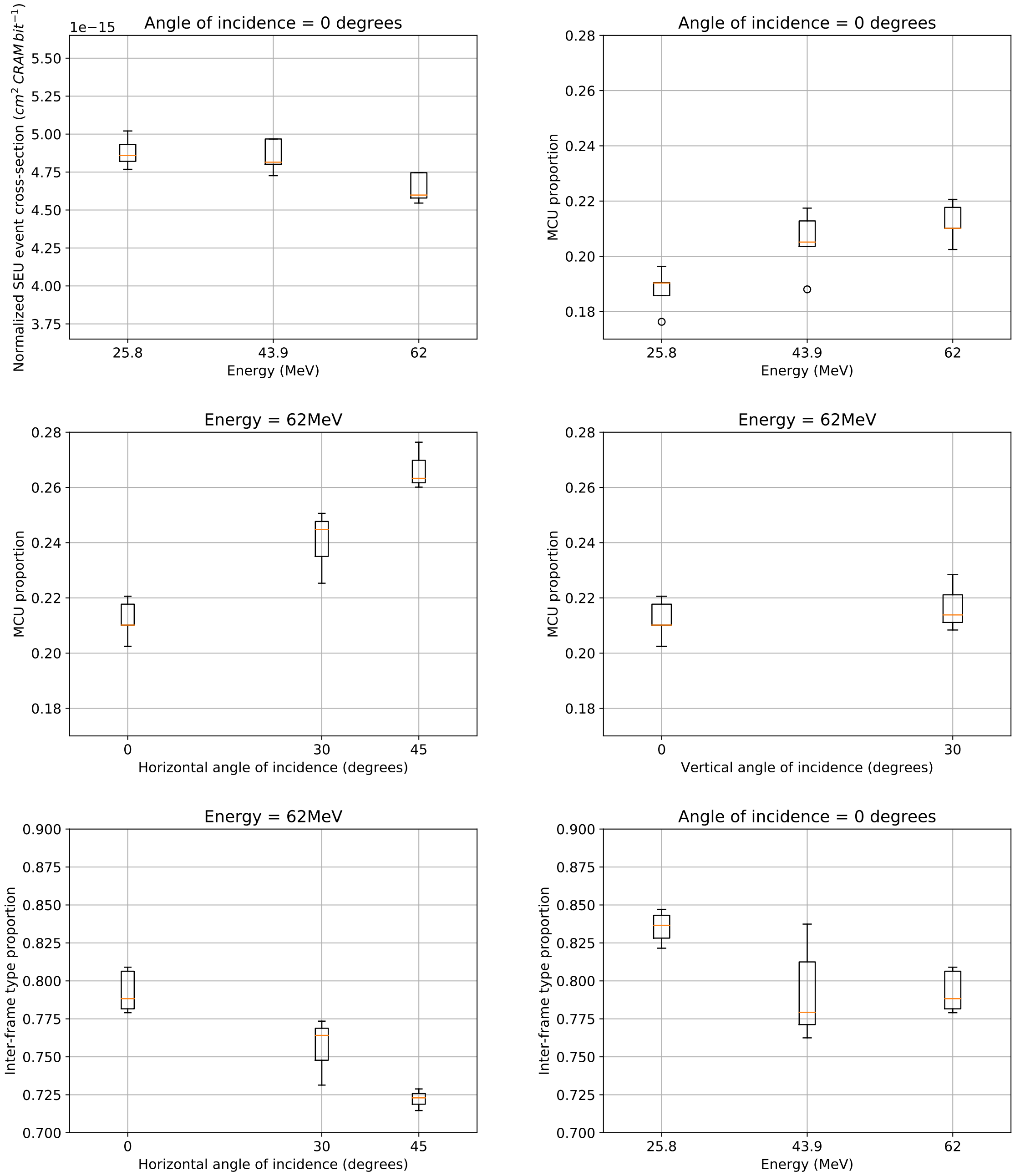


Figure 5.3: Selection of data from Artix-7 CRAM proton irradiation test

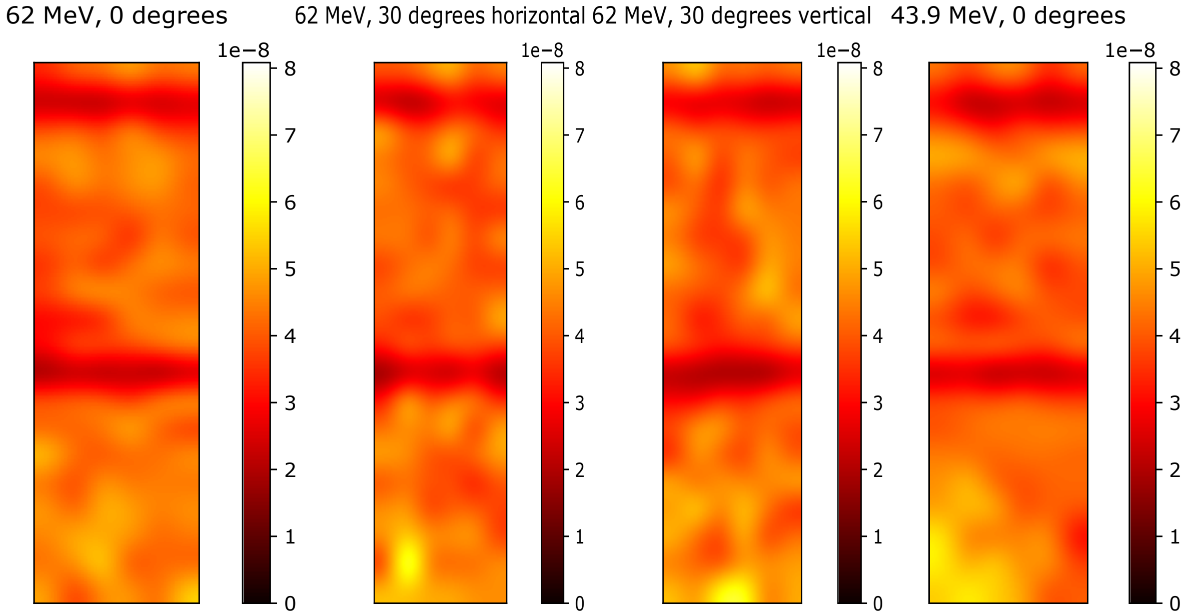


Figure 5.4: SEU location PDF of 4 data points

	Per device	Normalized per CRAM bit
Proton SEU cross-section (particle event)	$1.15 \times 10^{-7}\,\mathrm{cm}^2$	$4.75 \times 10^{-15}\,\mathrm{cm}^2\ \mathrm{bit}^{-1}$
Proton upset cross-section (1 bit)	$1.45 \times 10^{-7}\,\mathrm{cm}^2$	$6.0 \times 10^{-15}\,\mathrm{cm}^2\ \mathrm{bit}^{-1}$

Table 5.1: Approximate cross-section values obtained at $(62\,\mathrm{MeV}, 0°)$

The incident angle and direction of a particle was also found to have a clear impact on the proportion of MCUs generated. The existence of a relative angular intensity for incident neutrons in the CMS environment would be useful to make prediction. Such data is not currently available in the simulation results of the Muon detector Phase-2 Technical Design Report.

By approximating the relation between SEU cross-section and energy as constant over the CMS energy spectrum, an upper bound for the expected rate of SEUs in the ME0 and GE2/1 detectors during Phase-2 can be calculated (Table 5.2). Taking the values obtained at $(62\,\mathrm{MeV}, 0°)$ as an upper bound, the expected proportion of MCUs is 21%, with again 21% of these being uncorrectable MBUs for a total of 4.4% of MBUs among SEUs. Note that one should also take into account the percentage of essential bits in the firmware (which varies from one firmware to the other), such that the expected number of problematic events in the CRAM will be lower than the expected number of SEUs, the firmware of the mitigation system experiment was reported at 24.37% essential bit proportion.

Even with this upper bound value, the assumption made in Section 3.5.3 that the SEU rate will be much lower than the repair rate is indeed verified (see Table 5.2). Note that the relation between the upset cross-section obtained here for the Artix-7 (Table 5.1) and the correctable event cross-section from Figure 2.13, obtained for a Virtex-6, is consistent with the Xilinx cross-section data which indicated that the Artix-7 had a neutron cross-section 50% lower than the Virtex-6. The expected upsets per day is however much larger here, even though the cross-section is lower. This is because the values in Table 5.1 are calculated based on the entire neutron flux, even though much of it is at an energy level which will not affect the FPGA.

	Neutron flux	Approx. SEU events per day	Approx. upsets per day
GE2/1	$2 \times 10^4\,\mathrm{Hz\,cm}^{-2}$	200	250
ME0	$4 \times 10^5\,\mathrm{Hz\,cm}^{-2}$	4000	5000

Table 5.2: Phase-2 upper-bound SEU expectations in GE2/1 and ME0, for a single OptoHybrid

Sector type	L1 errors	Relative L1 errors	L2 errors
Coarse	892	100%	11
Fine	337	37.7%	23
Fine with triplicated voters	290	32.5%	17

Table 5.3: TMR sector results

SEM correctable	14432
SEM uncorrectable	2163
Total	16595

Table 5.4: Detected event counts of the SEM IP Core

5.2 TMR data

5.2.1 Data culling

During the mitigation system experiment, a problem appeared in the incoming data from the DUT. As mentioned in Section 4.3.2, when an uncorrectable upset is detected by the Xilinx SEM IP Core, the monitoring program in the control computer sends a signal to the GLIB to trigger a reconfiguration of the DUT. The problem was that in some fraction of cases, the reconfiguration of the DUT was not completed properly (the *INIT_B* pin was held low, indicating an error in the post-configuration CRC check [2]). The exact cause of this issue is not known, but the result was a stream of incoherent data. This issue, and the time required to identify it, unfortunately took away valuable beam time, such that the amount of data gathered for this test was smaller than hoped.

During the analysis phase, the incoherent data had to be excised from the real data. The output of the monitoring program was plain-text logs of all the changes in the internal signals represented by a sequence of bits, associated with a clock counter starting from the latest reconfiguration. A parsing program was written to go through the logs and reorganize the data into JSON files containing explicit signal names. From these files, the number of L1 and L2 errors for each sector, and each sector input, was counted. For each file, the variance in the L1 error count of each sector was computed and outlier files, which from visual inspection seemed like corrupted data, were ignored.

5.2.2 Results by sector type

The data obtained on the errors flags is compiled in Table 5.3. For each sector, the appearance of an error flag for any of the TMR inputs was counted as one error, whether errors appeared in multiple inputs or not. A clear improvement in L1 errors is observed for the finer granularity, as well as a more modest improvement with the addition of triplicated voters. As L2 errors appear to be fairly rare, the amount of data gathered on them limits how much interpretation can be done. The data gathered points towards a slight increase with fine granularity, somewhat resorbed with triplicated voter. Manual placement constraints did not show any impact on L1 or L2 errors.

To verify Assumption B from Section 3.5.3, that DCEs are mostly caused by MCUs, the idea was to correlate the timestamped error detection data to the L2 error flags. However due to the limited amount of L2 data, and a design flaw in the timestamping software (the clock counter was reset when reconfiguration was triggered, but this was not properly recorded in the sector flag logs), this could not be done.

The number of correctable and uncorrectable events detected by the SEM IP Core is reported in Table 5.4. From this data we can calculate a few metrics relating SEUs to functional error rates, see Table 5.5. We can remark that the device-wide L1 error rate is at 9.15%, and critical bit proportion in the bitstream is usually between 2 and 10%. Critical bits are defined as the bits which will cause functional errors in the circuit if flipped, it would therefore make sense for the two values to be linked. Because of the lack of L2 data, it was decided to use the total values for all sectors in these metrics, though of course specific values for each TMR implementation can be used as well.

Note that the SEM events count correctable (SBU, inter-frame MCU), and uncorrectable (MBU) bit upsets. This means that inter-frame MCUs are counted as multiple upsets, and therefore the total SEM event count would be slightly higher than the SEU event count. From this knowledge we can make the following approximations:

Metric	Device-wide	Normalized by sector
$\frac{Errors_{L2}}{Errors_{L1}+Errors_{L2}}$	3.1 %	/
$\frac{Errors_{L1}}{SEM\ Events}$	9.15 %	0.38 %
$\frac{Errors_{L2}}{SEM\ Events}$	0.289 %	0.012 %

Table 5.5: Functional error metrics for the experimental firmware

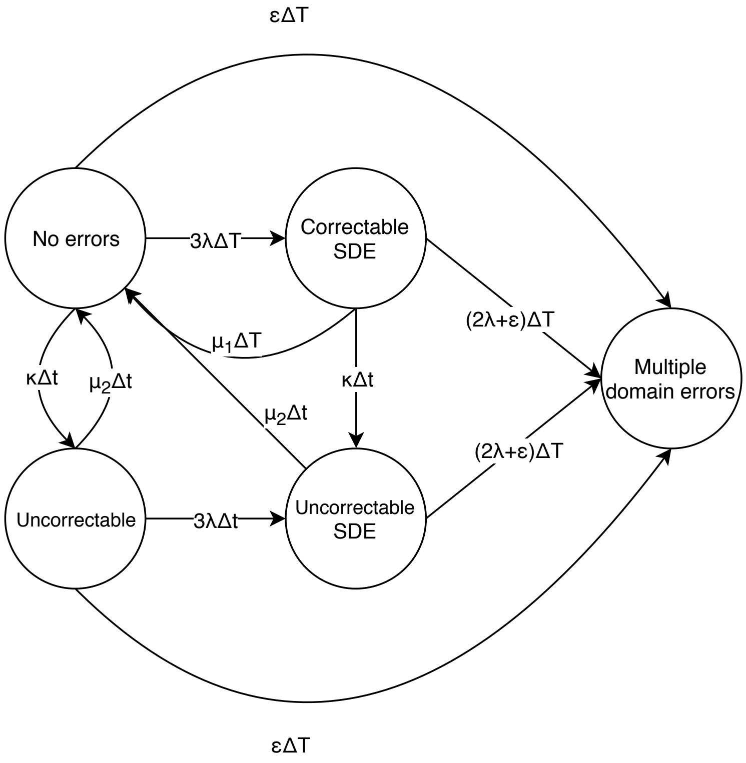


Figure 5.5: Simplified Markov chain model

$$\frac{Errors_{L1}}{SEM\ Events} \approx P(SDE|SEU)$$

$$\frac{Errors_{L2}}{SEM\ Events} \approx P(DCE|SEU)$$

5.2.3 Markov chain model simulations

Specific values linking SBU, MCU and MBU to SDE and DCE are not available from the data gathered, therefore in order to use the data in the Markov chain model, it is slightly tweaked as in Figure 5.5. In this version, the probability of going into an uncorrectable state is uncorrelated to functional errors in the TMR module, and it is assumed that $1 - P(SDE|MBU) \approx 1$. In a firmware similar to the one used for the tests, were each sector takes up a small amount of space in the FPGA, the probability of an SEU causing an SDE in any one sector is relatively small (see the normalized values in Table 5.5). Therefore this approximation is likely to hold.

Starting from the parameters used in Figure 3.15, the various parameters of the proposed model are increased in Figures 5.6 and 5.7. It is clear from the influence of κ that non-correctable errors have a huge impact in reliability, this makes sense as with the low rate of SEUs, SDEs would not normally have a chance to accumulate. However, when a non-correctable error appears, the scrubbing system stops and

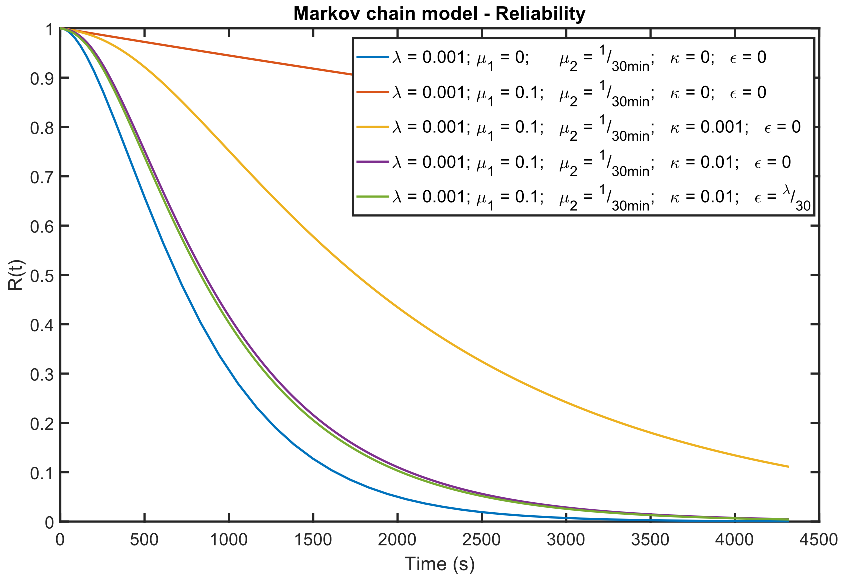


Figure 5.6: Influence of Markov model parameters on simulated TMR reliability

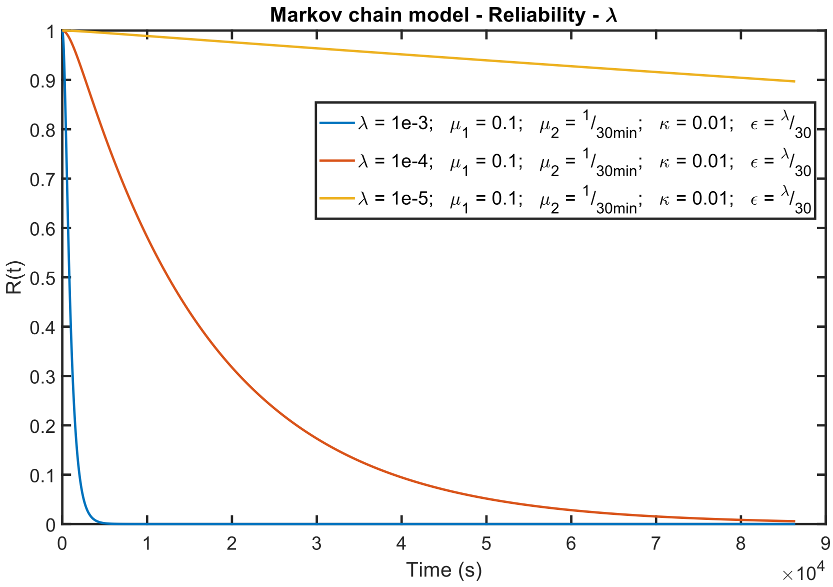


Figure 5.7: Influence of SDE rate on simulated TMR reliability

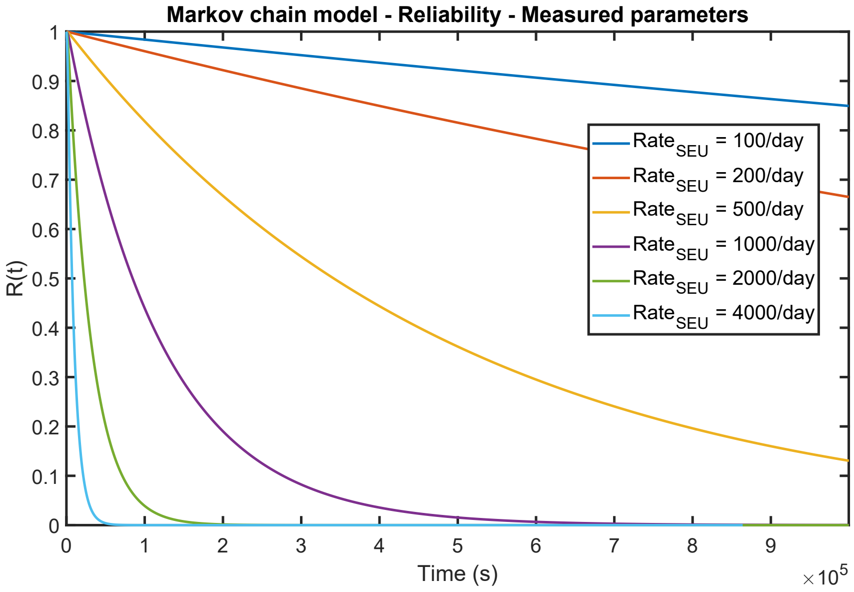


Figure 5.8: Impact of SEU rate on the reliability of the experimental TMR module based on measured parameters

$Rate_{SEU}$ (per day)	Mean time to failure (days)
100	70.5042
200	28.3551
500	5.6924
1000	1.4029
2000	0.3686
4000	0.1148

Table 5.6: Mean time to failure of the experimental TMR module based on measured parameters

they are free to do so. We can see that a sensible value of ϵ (informed by the data in Table 5.5) has a small but measurable impact on the reliability. By varying λ we can see that a lower $P(SDE|SEU)$ will also improve reliability significantly.

The data from Table 5.5 is then applied directly to the model, such that the parameters are

$$\begin{aligned}
\lambda &= Rate_{SEU} \cdot P(SDE|SEU) = Rate_{SEU} \cdot 0.0038 \\
\epsilon &= Rate_{SEU} \cdot P(DCE|SEU) = Rate_{SEU} \cdot 0.00012 \\
\kappa &= Rate_{MBU} = 0.04 \cdot Rate_{SEU} \\
\mu_1 &= 0.1 \\
\mu_2 &= 30\,\text{min}^{-1}
\end{aligned}$$

The reliability of the experimental TMR module with respect to the SEU rate is shown on Figure 5.8. Note that assuming a correction delay of 10ms, we should have $\mu_1 = 100$, but this was found to increase the simulation time tremendously, and results from $\mu_1 = 0.1$ and $\mu_1 = 1.0$ were extremely close. From the reliability, the mean time to failure can be calculated by Eq. 3.6. The result of this is shown on Table 5.6 for the values from Figure 5.8. Given that the upper-bound values from Table 5.2 are somewhat pessimistic, the mean time to failure for the experimental TMR module in CMS conditions can be expected to be above the 0.11 days simulated. With the addition of essential bit classification, the MBU rate can in this case be reduced by a quarter ($\approx 25\%$ of the CRAM is classified as essential in the experimental firmware). If one also uses the enhanced repair mode, which can correct two-adjacent-bit errors, the non-correctable error rate can be further reduced by about half (see Figure 5.2). In simulation, this has improved the mean time to failure in ME0 conditions from 0.11 days to 0.3 days (see Figure 5.9).

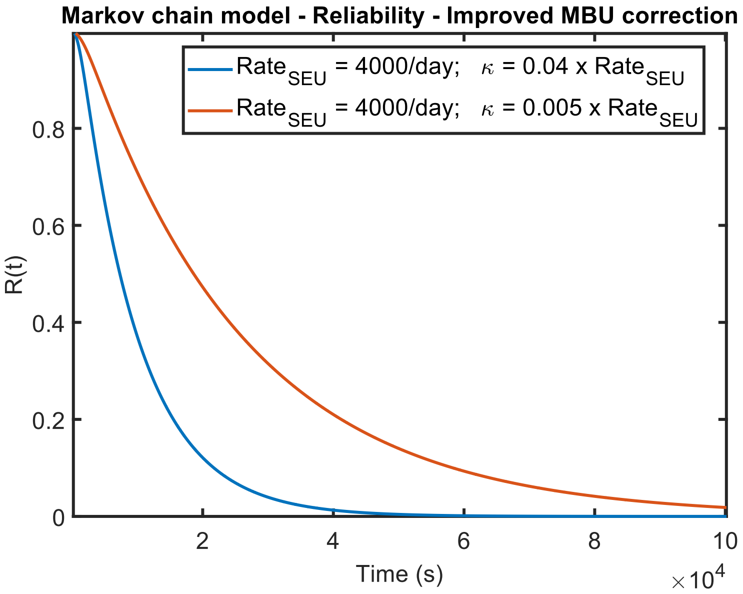


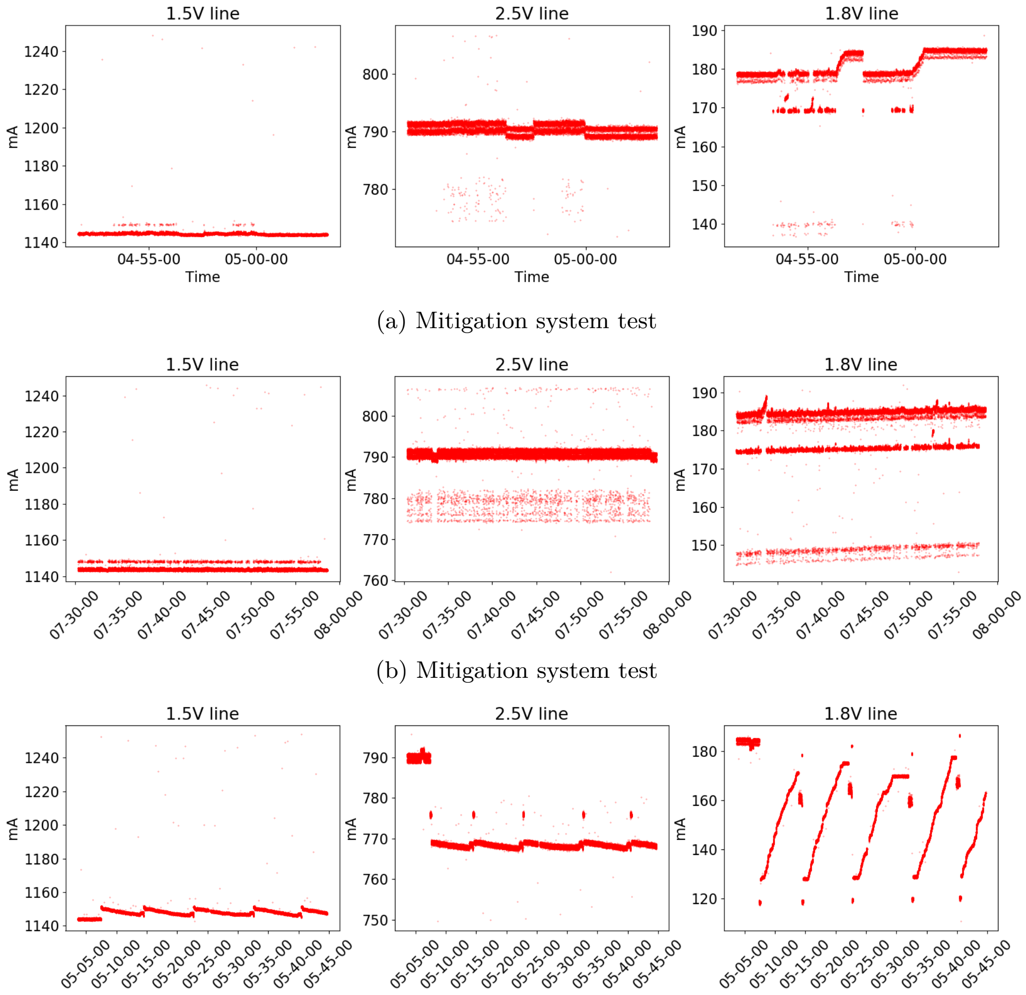
Figure 5.9: Impact of MBU correction improvements on reliability in ME0 conditions for the experimental TMR module

From the considerations above, it is clear that the principal influence in TMR breakdown with the proposed mitigation system would not be DCEs. Therefore the simpler reliability model proposed in Section 3.5.3 is misguided in its approach, although from computation it gives dramatically lower reliability, it is only supposed to be a lower bound which assumes all MCUs produce DCEs. Measurement of the ad-hoc parameters would therefore be required to predict the reliability of a TMR system.

5.2.4 Conclusions

From the experimental data, it can be observed that finer granularity TMR has a clear impact on reducing the amount of L1 errors, and therefore SDE probability. Although the amount of L2 errors gathered is not enough to conclude on the effect of granularity on the DCE probability, the results obtained with the Markov model indicate that this is not the limiting factor. Indeed, it was found that uncorrectable errors in the device have a major impact on the reliability predicted by the model, as they block the scrubber and allow SDE accumulation. The reduction in SDE probability offered by finer granularity TMR does play a role in mitigating these effects. Improvement can also be attained by improving the correction capabilities of the scrubbing system, the simplest solution being to use the enhanced repair mode of the Xilinx SEM IP core, and essential bit classification. This problem can also be solved entirely be changing the scrubbing system to one that can repair any MCU, such as a golden-copy based system. As mentioned in Section 3.2, this requires some way of receiving clean bitstream frames at the OptoHybrid, one way of doing this without adding a rad-hard memory to the board could be to send them from off-detector memory through the optical link.

Finally, with regards to the operation of the OptoHybrid, although the parameters used in the model above are not likely not hold for a TMR implementation in the OptoHybrid firmware, a general idea of the time scale for the mean time to failure of the device was found to be in the order of days, though large variations are observed when increasing the SDE rate λ. If TMR is implemented on larger modules than the ones used here (see Figure 4.10, each sector contains 9 modules), then the SDE probability can be expected to be larger if only because more SEUs will occur inside it. One should also take note of the fact that no presumptions are made on the effects of the aforementioned *failure* of a device on its output. It should be assumed that in the event of a failure, the output of the device cannot be trusted for at most 30 minutes, as that is the duration between reconfiguration in CMS. To make a less restrictive prediction, the outputs of the actual OptoHybrid modules should be tested in error injection or beam testing.



(a) Mitigation system test

(b) Mitigation system test

(c) Static cross-section test

Figure 5.10: Current measurements at three points during the tests

5.3 Currents and voltages

Current measurements at three points during the tests are reproduced on Figure 5.10. Figures 5.10a and 5.10b were measured during the mitigation system test, several current levels can be observed in these figures. The lowest levels were likely during the reconfiguration of the FPGA, while the highest in 5.10a occurred in periods when the beam was shut off. Figure 5.10c was measured during the static cross-section test, it can be observed that as SEUs accumulate in the device, the current in the 1.8V line, which feeds the VCCINT, VCCAUX and VCCBRAM inputs of the FPGA, builds up. However when the device is reset, it returns to normal levels. These increases could be explained by the creation of random circuitry inside the FPGA, e.g. short-circuits created by changes in the routing matrices. Many discrete spikes in current can be observed in all the measurements and in all the lines, but from the data gathered, there is no obvious sign of latch-up. Finally, a steady increase in current can be seen in 5.10b. This could be due to temperature or radiation effects, an extensive post-irradiation analysis of the device is still to be done in order to find which.

5.4 TID

The device was exposed to a total dose of 94 krad, and no permanent damage was observed, in accordance with previous results. Based on the projected TID of the ME0 OptoHybrid (30 krad), accumulated radiation effects are not expected to be a problem in CMS operation.

Conclusion

This thesis has given an overview of the data acquisition system of the CMS Muon detector, and indicated the place of the OptoHybrid board in this chain. The possible effects of radiation in an SRAM-based FPGA have been identified in two groups: accumulated effects and single event effects. It has been established that accumulated effects will not be a problem for the ME0 and GE2/1 detectors, as the total ionizing dose over their lifetime has already been surpassed in accelerated radiation testing, without permanent damage. Within SEEs, it was said that SEUs in the configuration memory are the principal cause of system failures in FPGAs. Results were presented from existing studies of neutron/proton SEU cross-sections in the configuration memory of two families of Xilinx FPGAs, the current 7-series and the previous Virtex-6 used in GE1/1. It was shown that current scaling trends appear to reduce the SEU cross-section while increasing the proportion of multiple cell upsets.

Following the beam testing campaign, data specific to the Artix-7 was found for the proton CRAM SEU cross-section in the energy range 25.8 to 62 MeV, as well as statistical data on the proportion of MCUs using a static testing method proposed in [57]. Relations between these characteristic and the incident angle of the proton beam were also found at 30°and 45°. The cross-section data obtained was consistent with previous results in the 7-series family, but did not exhibit the same reduction at 20 MeV found in the GE1/1 results. From the MCU statistical data, it was found that energy and angle of incidence both increase the MCU proportion, and the subset of errors which are not correctable by the built-in Xilinx ECC.

Following an overview of the SEU mitigation techniques which can be applied at the firmware level, a system combining the built-in Xilinx configuration memory scrubbing system (SEM IP Core) with Triple Modular Redundancy was chosen. As this system is unable to repair the subset of MCUs which create multiple errors in the same frame of memory, and because the TMR protection can be broken in certain situations, two models for predicting the reliability of this protection were proposed. The first simple model aimed at providing a lower bound for reliability, by making the assumptions that only one SEU would be present in the system at any one time during CMS operation, and that domain-crossing errors, which break TMR, occur with any MCUs, but not with single bit upsets. Because this model does not account for the fact that the SEM IP Core will stop functioning when encountering a non-repairable upset, a second model, based on a Markov chain, was proposed. This model aimed at providing an accurate estimation of the reliability by using parameters measurable for a specific module design.

Although not all the data that was hoped for was obtained, the beam testing of the mitigation system has shown that increasing TMR granularity will yield a significant improvement in error rates. Simulation with the proposed Markov chain have also revealed that, within that model, non-repairable MCUs have a major impact on reliability. Although it was established from the cross-section data that the first assumption of the simple model was likely to hold, the results obtained from the Markov chain model indicate that ignoring the effects of non-repairable MCUs is misguided. As such, measuring the ad-hoc parameters required for the Markov model, either in beam testing or error injection, is the approach recommended by this thesis for reliability estimations. The parameters measured for the experimental TMR module, which does not reflect the OptoHybrid modules, have resulted in a lower bound mean time to failure estimated in days for GE2/1, and hours for ME0. Only lower bound values could be given as the expected energy threshold below which the SEU cross-section should fall, found at 20MeV in GE1/1, was not found during these tests. Therefore the full neutron flux was used to predict the SEU rates, which is most likely extremely pessimistic.

MCUs will therefore be a major factor in the reliability of the OptoHybrid FPGA if the built-in ECC is used. Obtaining data on the relative angular intensity in CMS at the OH location would be useful for accurate predictions, and cross-section data should be found in the lower energy ranges. If with this data obtained, the estimated reliability is deemed too low for CMS, then it can be increased through the improvements proposed in Section 5.2.4.

Glossary

ASIC Application-Specific Integrated Circuit.

BRAM Block Random-Access Memory.

CLB Configurable Logic Block.

CMS Compact Muon Solenoid.

CRAM Refers to the configuration memory of an FPGA.

CRC Cyclic Redundancy Check.

critical bit A bit within the configuration memory of an FPGA which will modify the functionality of the design if flipped.

CSC Cathode Strip Chamber.

DCE Domain Crossing Error.

DT Drift Tube.

DUT Device Under Test.

ECAL Electromagnetic Calorimeter.

ECC Error Correction Code.

essential bit A bit within the configuration memory of an FPGA which will modify the circuitry of the design if flipped, without necessarily impacting the functionnality.

FPGA Field-Programmable Gate Array.

GBT GigaBit Transceiver.

GEB GEM Electronics Board.

GEM Gas Electron Multiplier.

GLIB Gigabit Link Interace Board.

hard error An error in a system which requires a power cycling of the device to repair.

HCAL Hadronic Calorimeter.

HDL Hardware Description Language.

IC Integrated Circuit.

ICAP Internal Configuration Access Port.

LET Linear Energy Transfer.

LHC Large Hadron Collider.

LUT Look-Up Table.

MBU Multiple Bit Upset.

MCU Multiple Cell Upset.

MTTF Mean Time To Failure.

MTTR Mean Time To Repair.

OH OptoHybrid.

PISO Parallel In Serial Out.

radiation hardened (rad-hard) Refers to an electronic component which has been made resistant to the effects of radiations by some method.

RPC Resistive Plate Chamber.

SBU Single Bit Upset.

SDE Single Domain Error.

SECDED Single Error Correction, Double Error Detection.

SEE Single Event Effect.

SEL Single Event Latchup.

SEM Soft Error Mitigation.

SET Single Event Transient.

SEU Single Event Upset.

SoC System on Chip.

soft error An error in a system which does not require a power cycling of the device to repair.

SRAM Static Random-Access Memory.

TMR Triple Modular Redundancy.

Bibliography

[1] *7 Series FPGAs Configurable Logic Block*. User Guide UG474. Xilinx, 2016, p. 74.

[2] *7 Series FPGAs Configuration User Guide*. User Guide UG470. Xilinx, 2018, p. 180.

[3] D. Acosta and A. Tapper. *CMS Technical Design Report for the Level-1 Trigger Upgrade*. Technical Design Report CMS-TDR-012. CERN, 18 June 2013. URL: `https://cds.cern.ch/record/1556311` (visited on 07/05/2019).

[4] David Barney. *CMS Slice*. CERN Document Server. 16 Feb. 2015. URL: `https://cds.cern.ch/record/2628641` (visited on 25/05/2019).

[5] Niccolò Battezzati, Luca Sterpone and Massimo Violante. *Reconfigurable Field Programmable Gate Arrays for Mission-Critical Applications*. New York: Springer-Verlag, 2011. ISBN: 978-1-4419-7594-2. URL: `https://www.springer.com/la/book/9781441975942` (visited on 09/05/2019).

[6] Giuseppe Battistoni, Francesco Broggi and Markus Brugger. "The application of the Monte Carlo code FLUKA in radiation protection studies for the Large Hadron Collider". In: *Progress in Nuclear Science and Technology* 2 (2011), pp. 358–364. ISSN: ISSN 2185-4823. DOI: `10.15669/pnst.2.358`. URL: `http://inis.iaea.org/Search/search.aspx?orig_q=RN:49052049` (visited on 25/05/2019).

[7] J. Blomgren et al. "Mono-energy neutron testing of Single Event Effects". In: *International Conference on Nuclear Data for Science and Technology*. International Conference on Nuclear Data for Science and Technology. EDP Sciences, 2007, pp. 1287–1290. DOI: `10.1051/ndata:07734`. URL: `http://dx.doi.org/10.1051/ndata:07734` (visited on 01/06/2019).

[8] Fredrik Brosser and Emil Milh. "SEU Mitigation Techniques for Advanced Reprogrammable FPGA in Space". MSc. Thesis. Chalmers University of Technology, 2014. 128 pp.

[9] Ken Chapman. "SEU Mitigation Techniques for SRAM based FPGAs". Xilinx, 30 Sept. 2015. URL: `https://indico.nikhef.nl/event/36/contribution/0/material/slides/0.pdf` (visited on 02/06/2019).

[10] CMS Collaboration. *The Phase-2 Upgrade of the CMS Muon Detectors*. Technical Design Report CMS-TDR-016. CERN, 12 Sept. 2017. URL: `https://cds.cern.ch/record/2283189` (visited on 06/05/2019).

[11] CMS Collaboration. *The Phase-2 Upgrade of the CMS Tracker*. Technical Design Report CMS-TDR-014. CERN, 30 June 2017. URL: `https://cds.cern.ch/record/2272264` (visited on 24/05/2019).

[12] A. Colaleo et al. *CMS Technical Design Report for the Muon Endcap GEM Upgrade*. Technical Design Report CMS-TDR-013. CERN, 3 June 2015. URL: `https://cds.cern.ch/record/2021453` (visited on 07/05/2019).

[13] M. Desogus, L. Sterpone and D. M. Codinachs. "Validation of a tool for estimating the effects of soft-errors on modern SRAM-based FPGAs". In: *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*. 2014 IEEE 20th International On-Line Testing Symposium (IOLTS). July 2014, pp. 111–115. DOI: `10.1109/IOLTS.2014.6873681`.

[14] *Device Reliability Report*. User Guide (UG116). Xilinx, 2018, p. 106.

[15] *Dictionary: JESD88 | JEDEC*. URL: `https://www.jedec.org/standards-documents/dictionary/` (visited on 19/05/2019).

[16] Elena Dubrova. *Fault tolerant design*. Springer, 2013. 147 pp.

[17] Apollinari G et al., eds. *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1.* CERN Yellow Reports: Monographs 4/2017. Geneva: CERN, 2017. 599 pp. ISBN: 978-92-9083-470-0 978-92-9083-471-7.

[18] Ivan Griffin. *Gajski-Kuhn Y-chart.* URL: `http://texample.net/tikz/examples/gajski-kuhn-y-chart/` (visited on 26/05/2019).

[19] Jinhua Han and Gang Guo. "Characteristics of energy deposition from 1-1000 MeV proton and neutron induced nuclear reactions in silicon". In: *AIP Advances* 7.11 (1 Nov. 2017), p. 115220. DOI: `10.1063/1.4995529`. URL: `https://aip.scitation.org/doi/10.1063/1.4995529` (visited on 01/06/2019).

[20] Werner Herr and Bruno Muratori. "Concept of luminosity". CERN, 2006. URL: `http://cds.cern.ch/record/941318/files/p361.pdf`.

[21] Jameel Hussein and Gary Swift. *Mitigating Single-Event Upsets.* White Paper. Xilinx, 2015, p. 11.

[22] Allan H Johnston. *The Effect of Device Scaling on Single-Event Effects in Advance CMOS Devices.* Jet Propulsion Laboratory - California Institute of Technology, 3 June 2015, p. 12.

[23] Andrew Keys et al. "Developments in Radiation-Hardened Electronics Applicable to the Vision for Space Exploration". In: *AIAA SPACE 2007 Conference & Exposition.* 0 vols. AIAA SPACE Forum. American Institute of Aeronautics and Astronautics, 18 Sept. 2007. DOI: `10.2514/6.2007-6269`. URL: `https://arc.aiaa.org/doi/10.2514/6.2007-6269` (visited on 02/06/2019).

[24] C. Ghabrous Larrea et al. "IPbus: a flexible Ethernet-based control system for xTCA hardware". In: *Journal of Instrumentation* 10.2 (Feb. 2015), pp. C02019–C02019. ISSN: 1748-0221. DOI: `10.1088/1748-0221/10/02/C02019`. URL: `https://doi.org/10.1088%2F1748-0221%2F10%2F02%2Fc02019` (visited on 29/05/2019).

[25] D. S. Lee et al. "Single-Event Characterization of the 28 nm Xilinx Kintex-7 Field-Programmable Gate Array under Heavy Ion Irradiation". In: *2014 IEEE Radiation Effects Data Workshop (REDW).* 2014 IEEE Radiation Effects Data Workshop (REDW). July 2014, pp. 1–5. DOI: `10.1109/REDW.2014.7004595`.

[26] David S. Lee et al. "Addressing Angular Single-Event Effects in the Estimation of On-Orbit Error Rates". In: *IEEE Transactions on Nuclear Science* 62.6 (Dec. 2015), pp. 2563–2569. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/TNS.2015.2498641`. URL: `http://ieeexplore.ieee.org/document/7348778/` (visited on 27/12/2018).

[27] David S Lee et al. "Heavy-lon Irradiation of the Xilinx Kintex UltraScale+ FPGA". 2018. URL: `https://www.osti.gov/servlets/purl/1508451` (visited on 02/06/2019).

[28] Jérôme Lemaire. "Exploring FPGA hardening solutions at the detector level for the future high luminosity phase of the CMS experiment at the LHC". MSc. Thesis. Brussels: Université Libre de Bruxelles, 2016.

[29] Thomas Lenzi. "Development of the DAQ System of Triple-GEM Detectors for the CMS Muon Spectrometer Upgrade at LHC". PhD thesis. Brussels: Université Libre de Bruxelles, 2016.

[30] Linfei Wang et al. "Experimental study of single event upset and single event latch-up in SOI SRAM". In: *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT).* 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT). Oct. 2016, pp. 1506–1508. DOI: `10.1109/ICSICT.2016.7998784`.

[31] Daniel McMurtrey et al. "Estimating TMR Reliability on FPGAs Using Markov Models". 2008.

[32] P. Nsengiyumva et al. "A Comparison of the SEU Response of Planar and FinFET D Flip-Flops at Advanced Technology Nodes". In: *IEEE Transactions on Nuclear Science* 63.1 (Feb. 2016), pp. 266–272. ISSN: 0018-9499. DOI: `10.1109/TNS.2015.2508981`.

[33] Vlad-Mihai Placinta. "First results on KINTEX-7 FPGA testing in mixed field radiation at CHARM facility". 2017.

[34] Vlad-Mihai Placinta and Lucian N. Cojocariu. "Radiation Hardness Studies and Evaluation of SRAM-Based FPGAs for High Energy Physics Experiments". In: *Proceedings of Topical Workshop on Electronics for Particle Physics — PoS(TWEPP-17).* Topical Workshop on Electronics for Particle Physics. Santa Cruz, California: Sissa Medialab, 20 Mar. 2018, p. 085. DOI: `10.22323/1.313.0085`. URL: `https://pos.sissa.it/313/085` (visited on 16/11/2018).

[35] Marc Poizat. *Total Ionizing Dose : Radiation Environment and its Effects in EEE Components and Hardness Assurance for Space Applications.* 10 May 2017. URL: `https://indico.cern.ch/event/635099/contributions/2570674/attachments/1456398/2248961/Radiation_Effects_and_RHA_ESA_Course_9-10_May_2017_TID_MP_FINAL.pdf` (visited on 02/06/2019).

[36] Brian Hogan Pratt. "Analysis and Mitigation of SEU-induced Noise in FPGA-based DSP Systems". PhD thesis. Brigham Young University, 11 Feb. 2011.

[37] H. Quinn et al. "Radiation-induced multi-bit upsets in SRAM-based FPGAs". In: *IEEE Transactions on Nuclear Science* 52.6 (Dec. 2005), pp. 2455–2461. ISSN: 0018-9499. DOI: `10.1109/TNS.2005.860742`.

[38] Heather Quinn, Paul Graham and Brian Pratt. "An Automated Approach to Estimating Hardness Assurance Issues in Triple-Modular Redundancy Circuits in Xilinx FPGAs". In: *IEEE Transactions on Nuclear Science* 55.6 (Dec. 2008), pp. 3070–3076. ISSN: 0018-9499. DOI: `10.1109/TNS.2008.2006750`. URL: `http://ieeexplore.ieee.org/document/4723751/` (visited on 06/01/2019).

[39] Heather Quinn et al. "Domain Crossing Errors: Limitations on Single Device Triple-Modular Redundancy Circuits in Xilinx FPGAs". In: *IEEE Transactions on Nuclear Science* 54.6 (Dec. 2007), pp. 2037–2043. ISSN: 0018-9499. DOI: `10.1109/TNS.2007.910870`. URL: `http://ieeexplore.ieee.org/document/4395073/` (visited on 06/01/2019).

[40] P. M. B. Rao et al. "Protecting SRAM-based FPGAs against multiple bit upsets using erasure codes". In: *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC).* 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC). June 2014, pp. 1–6.

[41] Tai Sakuma. *Cutaway diagrams of CMS detector.* 2019.

[42] *Soft Error Mitigation Controller.* Product Guide PG036. Xilinx, 4 Aug. 2018.

[43] H. Soltani, M. Dolatshahi and M. Sadeghi. "Comparing the reliability in systems with triple and five modular redundancy". In: *2016 5th International Conference on Computer Science and Network Technology (ICCSNT).* 2016 5th International Conference on Computer Science and Network Technology (ICCSNT). Dec. 2016, pp. 437–442. DOI: `10.1109/ICCSNT.2016.8070196`.

[44] L. Sterpone and M. Violante. "A new analytical approach to estimate the effects of SEUs in TMR architectures implemented through SRAM-based FPGAs". In: *IEEE Transactions on Nuclear Science* 52.6 (Dec. 2005), pp. 2217–2223. ISSN: 0018-9499. DOI: `10.1109/TNS.2005.860745`.

[45] L. Sterpone et al. "On the design of tunable fault tolerant circuits on SRAM-based FPGAs for safety critical applications". In: *2008 Design, Automation and Test in Europe.* 2008 Design, Automation and Test in Europe. Munich, Germany: IEEE, Mar. 2008, pp. 336–341. ISBN: 978-3-9810801-3-1 978-3-9810801-4-8. DOI: `10.1109/DATE.2008.4484702`. URL: `http://ieeexplore.ieee.org/document/4484702/` (visited on 06/01/2019).

[46] Luca Sterpone, Boyang Du and Luca Boragno. "SRAM-based FPGAs: tools and methods for analysis and mitigation of radiation effects". 28 Oct. 2016. URL: `https://indico.cern.ch/event/574635/attachments/1364014/2065417/VERI_Place_Polito_Online.pdf` (visited on 02/06/2019).

[47] Aaron Gerald Stoddard. "Configuration Scrubbing Architectures for High-Reliability FPGA Systems". MSc. Thesis. Brigham Young University, 2015.

[48] A. Svetek et al. "The Calorimeter Trigger Processor Card: the next generation of high speed algorithmic data processing at CMS". In: *Journal of Instrumentation* 11.2 (Feb. 2016), pp. C02011–C02011. ISSN: 1748-0221. DOI: `10.1088/1748-0221/11/02/C02011`. URL: `https://doi.org/10.1088%2F1748-0221%2F11%2F02%2Fc02011` (visited on 29/05/2019).

[49] *The Large Hadron Collider | CERN.* URL: `https://home.cern/science/accelerators/large-hadron-collider` (visited on 02/05/2019).

[50] Jorge Tonfat et al. "Analyzing the Effectiveness of a Frame-Level Redundancy Scrubbing Technique for SRAM-based FPGAs". In: *IEEE Transactions on Nuclear Science* 62.6 (Dec. 2015). Experimental scrubbing, pp. 3080–3087. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/TNS.2015.2489601`. URL: `http://ieeexplore.ieee.org/document/7348832/` (visited on 06/01/2019).

[51] J. M. Trippe et al. "Predicting Muon-Induced SEU Rates for a 28-nm SRAM Using Protons and Heavy Ions to Calibrate the Sensitive Volume Model". In: *IEEE Transactions on Nuclear Science* 65.2 (Feb. 2018), pp. 712–718. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/TNS.2017.2786585`. URL: `http://ieeexplore.ieee.org/document/8234670/` (visited on 29/11/2018).

[52] T. Vaňát et al. "Comparing proton and neutron induced SEU cross section in FPGA". In: *2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*. 2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS). Apr. 2016, pp. 1–4. DOI: 10.1109/DDECS.2016.7482480.

[53] Erik Verhagen. "Development of the new trigger and data acquisition system for the CMS forward muon spectrometer upgrade". PhD thesis. Brussels: Université Libre de Bruxelles, 26 Feb. 2015.

[54] Michael P Vittitow. "Effects of Device Scaling on Angular Single-Event Effects". 2016.

[55] James Jixian Wang. "Radiation effects in FPGAs". In: 9th Workshop on Electronics for LHC Experiments. 2003. DOI: 10.5170/CERN-2003-006.34.

[56] M J Wirthlin, H Takai and A Harding. "Soft error rate estimations of the Kintex-7 FPGA within the ATLAS Liquid Argon (LAr) Calorimeter". In: *Journal of Instrumentation* 9.1 (16 Jan. 2014), pp. C01025–C01025. ISSN: 1748-0221. DOI: 10.1088/1748-0221/9/01/C01025. URL: http://stacks.iop.org/1748-0221/9/i=01/a=C01025?key=crossref.77ef575bfef2b0871987484ea5aa005d (visited on 22/11/2018).

[57] M. Wirthlin et al. "A Method and Case Study on Identifying Physically Adjacent Multiple-Cell Upsets Using 28-nm, Interleaved and SECDED-Protected Arrays". In: *IEEE Transactions on Nuclear Science* 61.6 (Dec. 2014), pp. 3080–3087. ISSN: 0018-9499. DOI: 10.1109/TNS.2014.2366913.

[58] K. Wyllie et al. "A Gigabit Transceiver for Data Transmission in Future High Energy Physics Experiments". In: *Physics Procedia*. Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011) 37 (1 Jan. 2012), pp. 1561–1568. ISSN: 1875-3892. DOI: 10.1016/j.phpro.2012.02.487. URL: http://www.sciencedirect.com/science/article/pii/S1875389212018706 (visited on 29/05/2019).

Appendix A

Device specifications

Board	OptoHybrid GE2/1
FPGA Model name	Artix7 (XC7A75T-2FGG484C)
Technology	28 nm
LC	75,520
Slices (2 per CLB)	11,800
CLB FlipFlops	94,400
DSP Slices	180
BRAM blocks (36 Kb ECC)	105
Total BRAM (Kb)	3780
Package dimensions (mm)	23x23
Speed grade	-2
F_max (MHz)	550
Bitstream size (bits)	30,606,404
Approximate CRAM size (bits)	24,007,668

Table A.1: Device under test

Appendix B

Experimental dataset

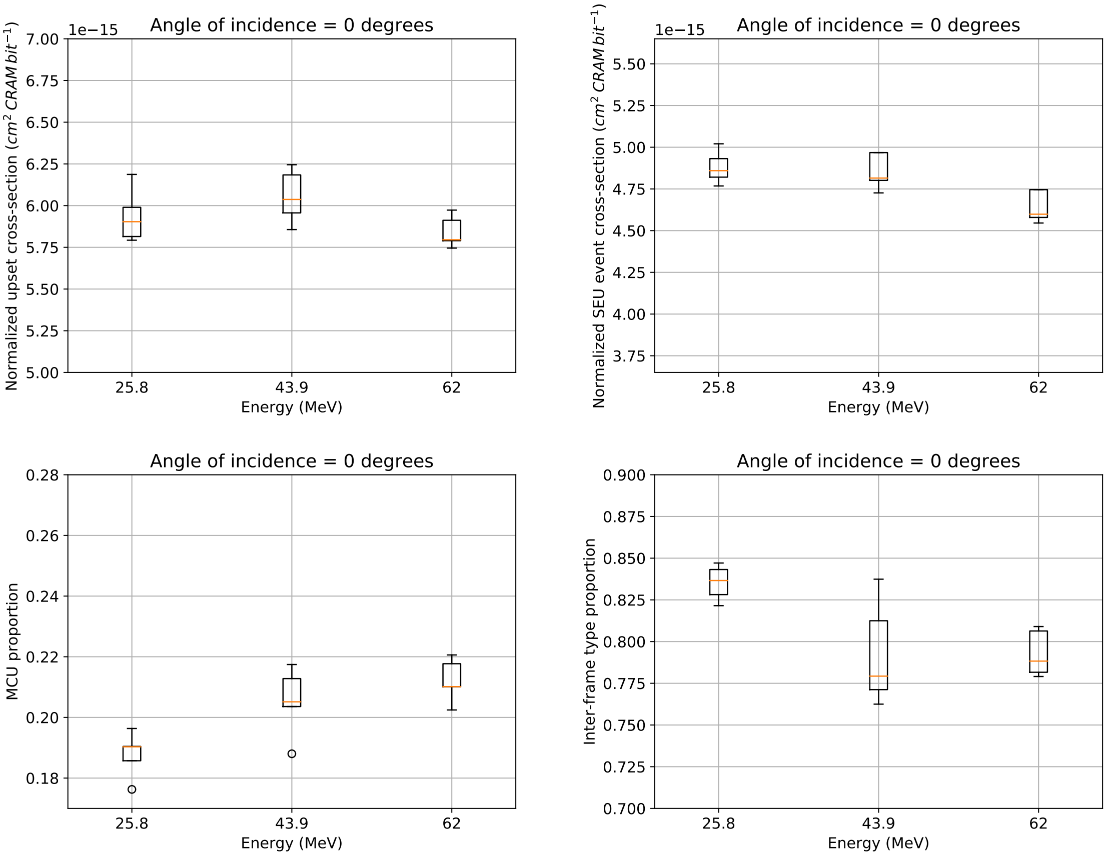


Figure B.1: Artix-7 CRAM proton irradiation data relative to energy

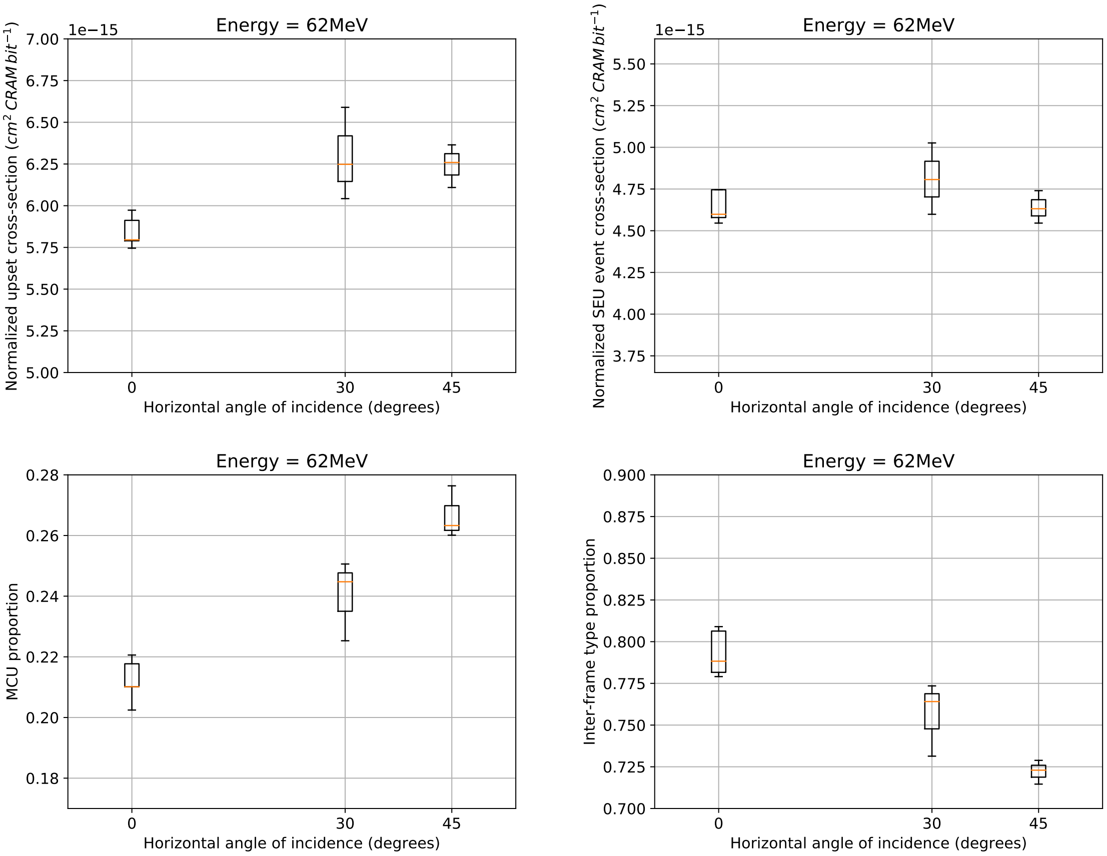


Figure B.2: Artix-7 CRAM proton irradiation data relative to horizontal device tilt

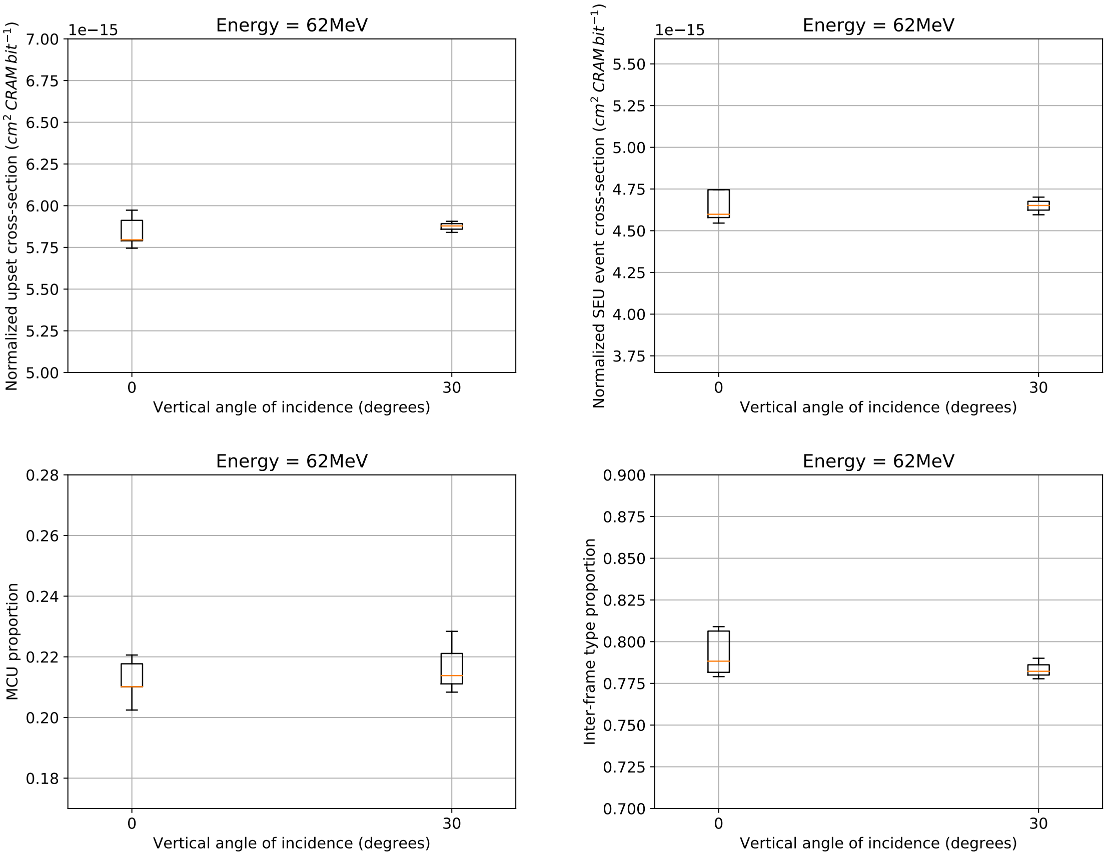
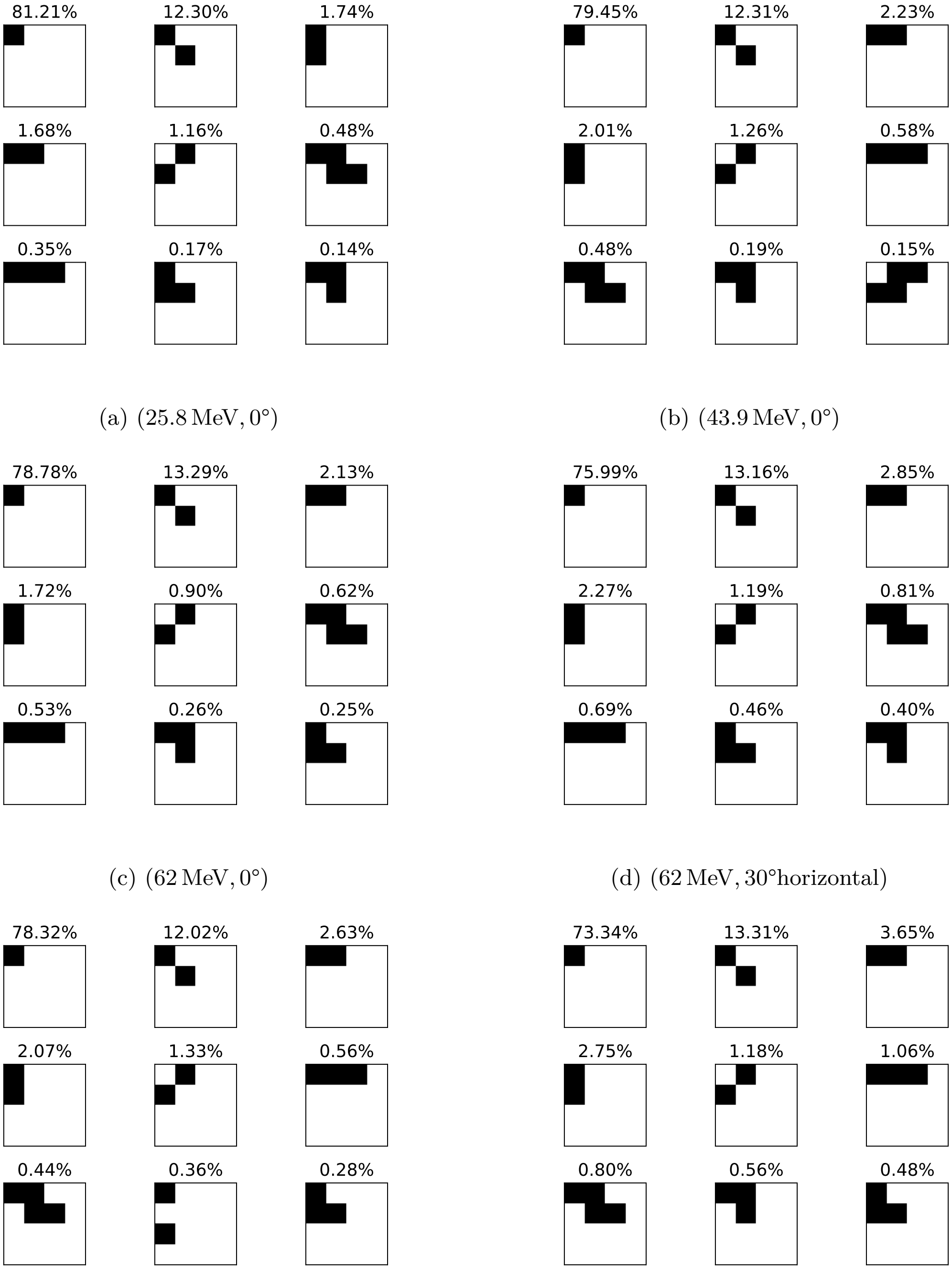


Figure B.3: Artix-7 CRAM proton irradiation data relative to vertical device tilt



(a) $(25.8\,\mathrm{MeV}, 0°)$

(b) $(43.9\,\mathrm{MeV}, 0°)$

(c) $(62\,\mathrm{MeV}, 0°)$

(d) $(62\,\mathrm{MeV}, 30°\mathrm{horizontal})$

(e) $(62\,\mathrm{MeV}, 30°\mathrm{vertical})$

(f) $(62\,\mathrm{MeV}, 45°\mathrm{horizontal})$

Figure B.4: MCU shape histograms from the irradiated bitstream

Appendix C

TMR Module - VHDL Example

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity tmr_module_basic is
   port (
      clk      : in  std_logic;
      din      : in  std_logic_vector(5 downto 0);
      dout    : out std_logic_vector(7 downto 0);
      err_flag : out std_logic
   ) ;
end entity ; -- tmr_module_basic

architecture arch of tmr_module_basic is
   signal comb1_o, comb2_o, comb3_o : std_logic_vector(7 downto 0);

   attribute DONT_TOUCH : string;
   attribute DONT_TOUCH of comb_module_i1  : label is  "TRUE";
   attribute DONT_TOUCH of comb_module_i2  : label is  "TRUE";
   attribute DONT_TOUCH of comb_module_i3  : label is  "TRUE";
   attribute DONT_TOUCH of comb1_o  : signal is "TRUE";
   attribute DONT_TOUCH of comb2_o  : signal is "TRUE";
   attribute DONT_TOUCH of comb3_o  : signal is "TRUE";

begin

   ------------------------------
   -- Comb module instantiation --
   ------------------------------
   comb_module_i1 : entity work.comb_module
      port map (
         clk  => clk,
         din  => din,
         dout => comb1_o
      );

   comb_module_i2 : entity work.comb_module
      port map (
         clk  => clk,
         din  => din,
         dout => comb2_o
      );
```

```
    comb_module_i3 : entity work.comb_module
       port map (
          clk  => clk,
          din  => din,
          dout => comb3_o
       );
    -----------------------------

    -----------------------------
    -- TMR Voter instantiation --
    -----------------------------
    tmr_voter_l1 : entity work.tmr_voter
       generic map (
          OUTPUT_SIZE => 8,
       )
       port map (
          clk => clk,
          i1  => comb1_o,
          i2  => comb2_o,
          i3  => comb3_o,
          o   => dout,
          err => err_flag
       );
    ------------------------------

 end architecture ; -- arch
```

Appendix D

Markov chain model solution

From the Markov chain in Figure 3.16, one can find the following set of equations:

$$\begin{aligned}
p_0(t+\Delta t) &= (1-(3\lambda+\kappa+\epsilon)\Delta t)\cdot p_0(t) + \mu_2\Delta t\cdot p_1(t) + \mu_1\Delta t\cdot p_2(t) + \mu_2\Delta t\cdot p_3(t)\\
p_1(t+\Delta t) &= \kappa\Delta t\cdot p_0(t) + (1-(3\lambda+\mu_2+\epsilon)\Delta t)\cdot p_1(t)\\
p_2(t+\Delta t) &= 3\lambda_1\Delta t\cdot p_0(t) + (1-(\mu_1+2\lambda+\epsilon+\kappa)\Delta t)\cdot p_2(t)\\
p_3(t+\Delta t) &= 3\lambda_2\Delta t\cdot p_0(t) + 3\lambda\Delta t\cdot p_1(t) + \kappa\Delta t\cdot p_2(t) + (1-(\mu_2+2\lambda+\epsilon)\Delta t)\cdot p_3(t)\\
p_4(t+\Delta t) &= \epsilon\Delta t\cdot p_0(t) + \epsilon\Delta t\cdot p_1(t) + (2\lambda+\epsilon)\Delta t\cdot p_2(t) + (2\lambda+\epsilon)\Delta t\cdot p_3(t)
\end{aligned}$$

By reorganizing these equations we get

$$\begin{aligned}
\frac{p_0(t+\Delta t)-p_0(t)}{\Delta t} &= -(3\lambda+\kappa+\epsilon)\cdot p_0(t) + \mu_2\cdot p_1(t) + \mu_1\cdot p_2(t) + \mu_2\cdot p_3(t)\\
\frac{p_1(t+\Delta t)-p_1(t)}{\Delta t} &= \kappa\cdot p_0(t) - (3\lambda+\mu_2+\epsilon)\cdot p_1(t)\\
\frac{p_2(t+\Delta t)-p_2(t)}{\Delta t} &= 3\lambda_1\cdot p_0(t) - (\mu_1+2\lambda+\epsilon+\kappa)\cdot p_2(t)\\
\frac{p_3(t+\Delta t)-p_3(t)}{\Delta t} &= 3\lambda_2\cdot p_0(t) + 3\lambda\cdot p_1(t) + \kappa\cdot p_2(t) - (\mu_2+2\lambda+\epsilon)\cdot p_3(t)\\
\frac{p_4(t+\Delta t)-p_4(t)}{\Delta t} &= \epsilon\cdot p_0(t) + \epsilon\cdot p_1(t) + (2\lambda+\epsilon)\cdot p_2(t) + (2\lambda+\epsilon)\cdot p_3(t)
\end{aligned}$$

Taking the limit as $\Delta t \to 0$ yields the continuous-time domain equations

$$\begin{aligned}
p_0'(t) &= -(3\lambda+\kappa+\epsilon)\cdot p_0(t) + \mu_2\cdot p_1(t) + \mu_1\cdot p_2(t) + \mu_2\cdot p_3(t)\\
p_1'(t) &= \kappa\cdot p_0(t) - (3\lambda+\mu_2+\epsilon)\cdot p_1(t)\\
p_2'(t) &= 3\lambda_1\cdot p_0(t) - (\mu_1+2\lambda+\epsilon+\kappa)\cdot p_2(t)\\
p_3'(t) &= 3\lambda_2\cdot p_0(t) + 3\lambda\cdot p_1(t) + \kappa\cdot p_2(t) - (\mu_2+2\lambda+\epsilon)\cdot p_3(t)\\
p_4'(t) &= \epsilon\cdot p_0(t) + \epsilon\cdot p_1(t) + (2\lambda+\epsilon)\cdot p_2(t) + (2\lambda+\epsilon)\cdot p_3(t)
\end{aligned}$$

The vector form of which is Eq. 3.13, which can be used to find the solution computationally. For the analytical solution, we can take the Laplace transform, which is

$$\begin{aligned}
sP_0(s)-p_0(0) &= -(3\lambda+\kappa+\epsilon)\cdot P_0(s) + \mu_2\cdot P_1(s) + \mu_1\cdot P_2(s) + \mu_2\cdot P_3(s)\\
sP_1(s)-p_1(0) &= \kappa\cdot P_0(s) - (3\lambda+\mu_2+\epsilon)\cdot P_1(s)\\
sP_2(s)-p_2(0) &= 3\lambda_1\cdot P_0(s) - (\mu_1+2\lambda+\epsilon+\kappa)\cdot P_2(s)\\
sP_3(s)-p_3(0) &= 3\lambda_2\cdot P_0(s) + 3\lambda\cdot P_1(s) + \kappa\cdot P_2(s) - (\mu_2+2\lambda+\epsilon)\cdot P_3(s)\\
sP_4(s)-p_4(0) &= \epsilon\cdot P_0(s) + \epsilon\cdot P_1(s) + (2\lambda+\epsilon)\cdot P_2(s) + (2\lambda+\epsilon)\cdot P_3(s)
\end{aligned}$$

Or in vector form:

$$\begin{pmatrix} p_0(0) \\ p_1(0) \\ p_2(0) \\ p_3(0) \\ p_4(0) \end{pmatrix} = A \cdot \begin{pmatrix} P_0(s) \\ P_1(s) \\ P_2(s) \\ P_3(s) \\ P_4(s) \end{pmatrix}$$

And from this the analytical solution is given by the inverse Laplace transform of

$$\begin{pmatrix} P_0(s) \\ P_1(s) \\ P_2(s) \\ P_3(s) \\ P_4(s) \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

And the reliability is given by

$$R(t) = 1 - p_4(t)$$