SINGLE EVENT UPSET TESTING OF FLASHED BASED FIELD PROGRAMMABLE GATE ARRAYS

By

Juan-Pierre Potgieter

Submitted in fulfilment of the requirements for the degree of

Master of Engineering in Mechatronics

in the Faculty of Engineering, The Built Environmental and Information Technology at the Nelson Mandela Metropolitan University

December 2014

Supervisor: Dr Farouk Smith

Department of Mechatronics

Port Elizabeth, South Africa

December 2014

DECLARATION

I, Juan-Pierre Potgieter, student number s209003191, in accordance with rule G4.6.3, hereby declare that this Dissertation for Master of Engineering in Mechatronics is my own work and that it has not previously been submitted for assessment or completion of any postgraduate qualification to another University or for another qualification.

Juan-Pierre Potgieter

Abstract

SINGLE EVENT UPSET TESTING OF FLASHED BASED FIELD PROGRAMMABLE GATE ARRAYS

Keywords: Single Event Upset, Field Programmable Gate Arrays, Single Event Effects on Video Graphics Array Controller, NRF iThemba LABS ,Single Event Effect Mitigation techniques, 66Mev Single Event Upset testing.

In the last 50 years microelectronics have advanced at an exponential rate, causing microelectronic devices to shrink, have very low operating voltages and increased complexities; all this has made circuits more sensitive to various kinds of failures. These trends allowed soft errors, which up until recently was just a concern for space application, to become a major source of system failures of electronic products. The aim of this research paper was to investigate different mitigation techniques that prevent these soft errors in a Video Graphics Array (VGA) controller which is commonly used in projecting images captured by cameras. This controller was implemented on a Flash Based Field Programmable Gate array (FPGA). A test set-up was designed and implemented at NRF iThemba LABS, which was used to conduct the experiments necessary to evaluate the effectiveness of different mitigation techniques. The set-up was capable of handling multiple Device Under Tests (DUT) and had the ability to change the angle of incidence of each DUT. The DUTs were radiated with a 66MeV proton beam while the monitoring equipment observed any errors that had occurred. The results obtained indicated that all the implemented mitigation techniques tested on the VGA system improved the system's capability of mitigating Single Event Upsets (SEU). The most effective mitigation technique was the OR-AND Multiplexer Single Event Transient (SET) filter technique. It was thus shown that mitigation techniques are viable options to prevent SEU in a VGA controller. The permanent SEU testing set-up which was designed and manufactured and was used to conduct the experiments, proved to be a practical option for further microelectronics testing at iThemba LABS.

Acknowledgements

This dissertation could not have been completed if it was not for the help and support from people and organizations. First of all I would like to thank my supervisor Dr F. Smith for introducing me to the field of radiation effects on electronics and allowing me to be part of his research. The knowledge and experience I gained while working under him is invaluable.

I would like to thank NRF iThemba LABS for granting us the beam time required to conduct the experiments. A special thanks to Dr Retief Neveling, Dr Rudolf Nchodu and Dr Ricky Smit for always being there to give a helping hand at the iThemba Labs.

Thank you to the National Research Foundation and The South African National Space Association for their generous support during the duration of this project.

Thank you to Stefan Van Aardt for the help in constructing the mechanical system.

To my parents, for all their support and encouragement throughout my academic career and for always believing in me.

Last but not least a big thanks to Keri Plumstead, who was always there for me. For all the reading and advice to all the cups of coffees that were made. Without her moral and emotional support this project would have never been a success.

Contents

List of Figures

List of Tables

List of Equations

Nomenclature

VI	LabVIEW V file-extension
HDL	Hardware Description Language
ASIC	Application Specific Integrated Circuit
FPGA	Field Programmable Gate Arrays
GEO	Geostationary Orbits
LEO	Low Earth Orbit
LET	Linear Energy Transfer
LETlt	LET threshold
MOS	Metal Oxide Silicon
MOSFET	Metal-oxide semiconductor field effect transistor
SEB	Single Event Burnout
SEE	Single Event Effects
SEL	Single Event Latch up
SEU	Single Event Upsets
SRAM	Static Random Access Memory
TMR	Triple Modular Redundancy
DMR	Dual Modular Redundancy
VGA	Video Graphics Array
SASA	South African Space Association
GG	Guard Gate
eV	Electron Volt
HSYNC	Horizontal Synchronization
VSYNC	Vertical Synchronization
TTL	Transistor-Transistor Logic
VHDL	Very-High-Speed-Integrated-Circuit Hardware Descriptive Language

Chapter 1

1. Introduction

1.1 Overview

Flash-based Field Programmable Gate Arrays (FPGAs) have become popular in satellite and aerospace applications as opposed to the Application Specific Integrated Circuit (ASIC) alternatives. FPGAs have the advantage over ASICs of being less expensive, available off the shelf and can be programmed and reprogrammed on site as often as desired. FPGAs do, however, have some disadvantages when compared to the ASIC. These include: slower computational speeds, less complex design capabilities and higher power requirements [1]. FPGAs have the potential to be utilised in aerospace applications; however, they are sensitive to radiation exposure which could pose possible problems when implementing FPGAs in the space environment. This radiation exposure can cause permanent or non-permanent damage. This dissertation will only investigate the non-permanent damage caused by radiation, called Single Event Effects (SEEs - radiation effects caused by a single particle strike). The importance of considering SEEs in the radiation research field has increased considerably in the last 40 years. This is due to the fact that device sizes are decreasing, which causes an increase to its vulnerability to the SEEs; however, smaller device size decreases permanent damage effects [2].

SEEs were first observed during the early 1970's where it went from being an unknown entity to a major part of radiation studies. In those times scientists only saw radiation effects as large amounts of radiation, the thought that one particle could cause radiation effects was unheard of. In 1972 the history of SEE's started at a company called Hughes. Hughes lost communication with a satellite for 96 seconds. Since there were a number of communication activities in progress, Hughes didn't understand the source of the problem. His colleague, Smith, then proposed the idea that a particle can deposit charge, which caused an upset in the circuitry [3].

In 1977 Rockwell launched the first prototype GPS satellite into geosynchronous orbit. The system on the satellite was consistently having, on average, one upset per day. Rockwell then started carrying out theoretical calculations and calculated that Dynamic random-access memory (DRAM) will have about one upset per day. Even with these calculations, Rockwell was still very skeptical with the results, as the idea of a single particle causing the upset seemed absurd. As time went by, components and electrical circuits became smaller. As the size reduced, the critical charge of the components became less and the occurrence of SEEs increased. Therefore, the importance for the testing of single particle radiation on electronics has become more significant in recent times.

Fast track to the 21st century and the effects of SEEs on microelectronics have grown exponentially. It has been observed that SEEs not only affect devices in space, but also devices here on Earth. In 2005 the ASC Q cluster supercomputer showed 7170 errors after 102 days in its 81-Gb cache memory [4]. In 1998 a study reported that every day, the 1 out of 10000 SRAMs attached to pacemakers underwent bit flips. This factor increased by 300 times when the patient was on an air craft [5].

This exponential growth lead to an increase of the number of radiation facilities[1] that are dedicated facilities for SEE testing on electronics. All these facilities, however, are located in the Northern hemisphere and priced at up to R5000 per hour to use the radiation source [6]. There are no known SEE testing facilities in the Southern hemisphere. This dissertation will look to change that.

1.2 Objectives

Section 1.1 described the importance of SEE testing and the need for methods to protect circuits against this effect, however, to test for the effects that radiation has on a FPGA and how susceptible it is to SEEs, a number of elements are required, namely:

- A source of radiation
- A microelectronic device to test under this source

- A set-up to monitor the SEEs that occur

Therefore, the main objectives of this dissertation are to:

1. Design and construct a set-up that can be used to test the effect radiation has on different microelectronics.
2. Investigate applications that are used on satellites that could be susceptible to SEEs
3. Investigate different techniques that are used to increase a circuit's tolerance towards SEEs and test its viability in a practical application.

After all this has been achieved a conclusion will be made on the best technique to increase a circuit's tolerance to SEEs and if the set-up designed is a plausible set-up to use for future tests needed in the radiation field.

1.3 Dissertation Outline

This introduction forms the first chapter of this dissertation. The remainder of the dissertation is organised as follows:

Chapter 2: Background and Related Work for SEE effects

Chapter 2 presents an overview of all the important issues regarding SEEs in microelectronic components as well as related research. The chapter begins by presenting different space radiation sources that affect the electronics in a satellite system. The chapter then introduces the type of effects these radiation sources can have on electronic devices and describes the methods to quantify these effects. Different SEE testing facilities are investigated to observe the various methods that are used. Different techniques are then described to mitigate these errors that have been caused by radiation sources. The chapter concludes by describing different applications used by satellites that could be susceptible to SEE occurrences.

Chapter 3: Video Graphics Array (VGA) Implementation

In Chapter 3 the implementation of the VGA is described. The chapter commences by describing the use of VGA controllers in satellites, especially CubeSats. The fundamentals of VGA are then explained by describing the operation of the system and the signals that are processed. The

fundamentals of a VGA controller were then designed in VHDL and implemented on a FPGA. The VHDL code is explained as well as the method used to implement different mitigation techniques in the VGA system.

Chapter 4: Experimental set-up and test methodology

The test methodology that was used to conduct the SEE testing is described in this chapter. The mechanical set-up that was designed is explained and analysed. The chapter then describes the method of the tests that were performed (i.e. how errors were captured and the other electronics used.) The chapter concludes by giving a complete overview of the mechanical and electrical aspects of the testing set-up and how they are integrated together.

Chapter 5: Experimental results and Discussion

The results that were obtained using the test methodology described in chapter 4 are evaluated and discussed. The chapter starts off with important parameters and calculations that are needed to analyse the results obtained. The chapter then shows the method of each mitigation technique that was implemented. The results of each mitigation technique are then displayed and discussed. The chapter concludes by comparing and discussing the results.

Chapter 6: Conclusion and Recommendations

Chapter 6 presents a set of conclusions that were drawn from the study and improvements that could be made to the current system. Further work is mentioned which could benefit the research field of radiation effects on FPGAs

Appendices:

Appendices are attached to provide further information on the work conducted.

Chapter 2

2. Literature Study

This chapter will describe SEE's as well as look at the history of SEE's. A background of radiation sources and the effects that radiation has on electronics will be covered as well as methods to protect electronics from these effects. Different testing facilities available will also be discussed followed by a discussion of different types of FPGA's.

2.1 Radiation Sources

It is important to know in which space environment the FPGAs that are normally found on spacecraft and satellites will be operating. In the vast "emptiness" of space there are high-energy particles that can cause malfunctions or even permanent damage of electronic components. These particles can be classified into three categories, namely trapped particles in the Van Allen Belts, low level flux of ions outside our solar system and lastly, bursts of radiations emitted by the sun.

2.1.1 Trapped Particles

The Von Allen Belt as can be seen in Figure 2.1 was first discovered in 1958 when Explorer I was launched with a Geiger counter on board. When Explorer reached about a 1000km in altitude the Geiger counter experienced a high count rate. Initially it was suggested that the Geiger counter malfunctioned, however when the Explorer III indicated the same results a little later it was demonstrated that these effects were real. Van Allen who conducted these experiments on the two Explorers, realised that these high counts on the Geiger counter were caused by trapped particles inside the earth's magnetic field [7].

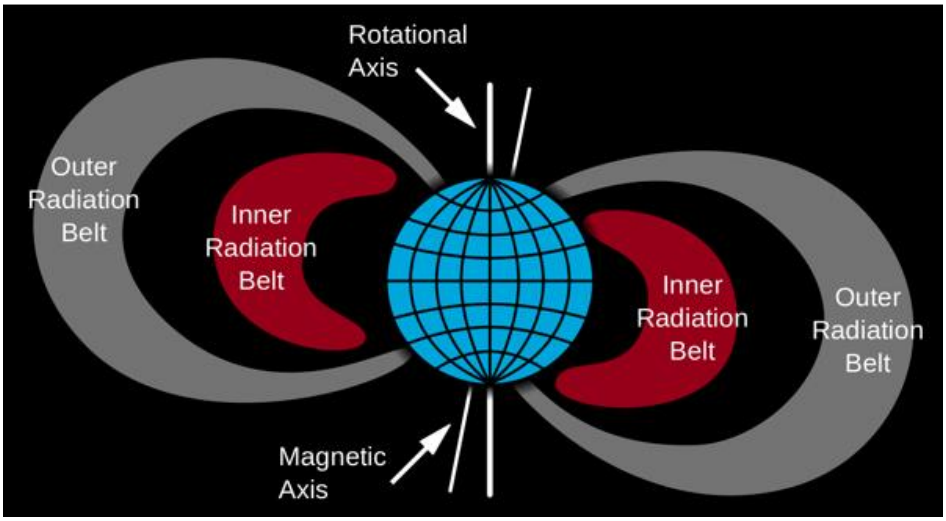


Figure 2.1 Van Allen belt [7]

The outer radiation belt of the Van Allen belt is approximately 19 000km above the Earth's surface. This belt consists of mainly protons and electrons with energies up to 100Mev. The inner radiation belt is located approximately 1000km above the Earth's surface, this belt mostly contains high energy protons [8].

2.1.2 Galactic Cosmic Rays (GCR)

Cosmic rays are situated outside our solar system, consisting of high energy radiation particles and are believed to be remnants from supernova explosions. These radiation particles consist of high energy protons and atomic nuclei. It is a common occurrence for cosmic rays to produce showers of secondary particles that penetrate the Earth's atmosphere; occasionally it might even reach the Earth's surface [9]. Single event effects are the main radiation effects caused by GCR in microelectronics and photonics [10].

Figure 2.2 illustrates the cosmic rays entering our solar system (red arrows).

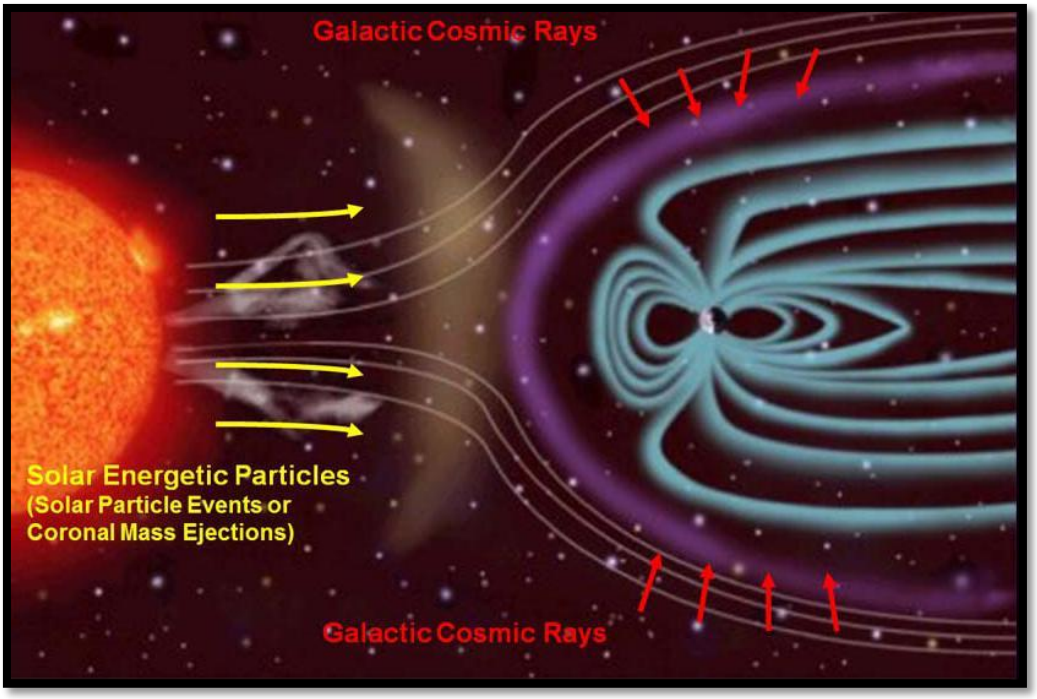


Figure 2.2 Galactic Cosmic rays [11]

2.1.3 Solar Particle Events

The plasma of charged particles is known as solar wind. This solar wind consists of electrons, protons and heavy ions that originate from the Sun's corona. Solar winds were first noticed by Ludwig Biermann from Germany. He noticed that a comet's tail always pointed away from the Sun, irrespective of the comet's direction of motion. He recommended that the reason for this phenomena was due to a stream of particles emitted by the Sun [12]. This occurrence was given the term "Solar wind" by Eugene Parker in 1958. It was given this name due to the way these particles flow [13]. As the Sun constantly heats up the particles in its corona, these particles increase in heat such that they are able to escape the Sun's gravitational pull. This high temperature causes a plasma stream that travels outward along the suns magnetic field lines. The Sun's rotation keeps the flow of the wind continuous. These winds can be seen in Figure 2.2 illustrated by the yellow arrows. The solar wind consists of 95% protons and the remaining 5% consists of helium and lesser doses of oxygen [14], these particles can have energies exceeding 10 MeV [20]. Solar winds might not be as dangerous as cosmic rays, however, consistent bombardment of with these

particles can cause failures in microelectronics. These solar storms can even result in Low Earth Orbit satellites prematurely re-entering the atmosphere.

2.2 Radiation effects

The effects radiation has on an electric component can either be permanent or just a temporary error in the system. This section will describe the different errors that occur and explain how they affect a specific system.

2.2.1 Single Event Effects

SEEs are induced when an energetic particle from the environment, be it heavy ions, protons or neutrons, hits an electronic medium through which it passes. A charged particle may generate a current pulse inside the silicone by either direct ionization or indirect ionization [15]. This ionization induces a current pulse in a p-n junction. SEEs are caused by a single particle strike and can either cause a temporary error called a soft error or a permanent error which can be fatal to electronic devices, called hard error, this is illustrated in Figure 2.3.

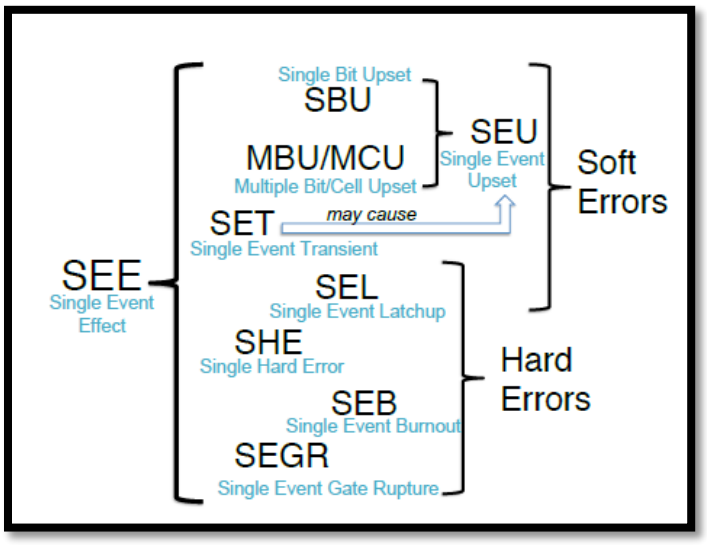


Figure 2.3 Different types of SEEs [16]

SEE's can be broken into different effects [16]:

- Single Event Upset (SEU)
- Single Event Burnout (SEB)
- Single Event Gate Rupture (SEGR)
- Single Event Latch-up (SEL)
- Single Event Transient (SET)

This dissertation only looked into effects that are relevant to the research, which are SEUs, SETs and SELs.

When a single particle collides with a circuit node in a digital system, it might result in a charge collection in that node, this can cause the circuit node voltage to change for a short period. If this voltage is large enough, a Single Event Transient (SET, also called a glitch) can occur (i.e. logic of 1 will be changed to logic level 0 and vice versa). The charge needed to be collected to cause the SET is called the Critical Charge [17].

If the SET is allowed to propagate in a digital system it can be captured by a memory element and a bit change is possible and this is known as a single event upset (SEU).

Figure 2.4 shows where an SET and SEU occurs when a particle strikes. A SET occurs when a particle strikes the combinational logic of a circuit whereas an SEU occurs when a particle strikes a memory element like a flip flop or a latch.

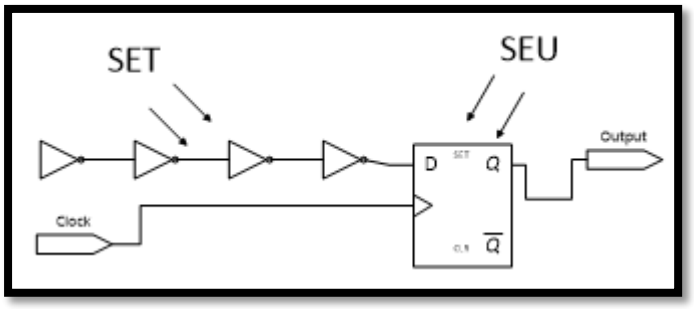


Figure 2.4 Indicating where a SET and SEU occur in a digital circuit

Each particle ionises a p-n junction in different methods. Heavy ions can create electrons and holes by direct ionisation seen in Figure 2.5 whereas protons themselves pass through the device with little effect, however, the collisions of proton in the device causes nuclear reactions known as secondary particles. This is indicated in Figure 2.5. These particles create the charge needed to cause the current pulse to create the SEU in the device.

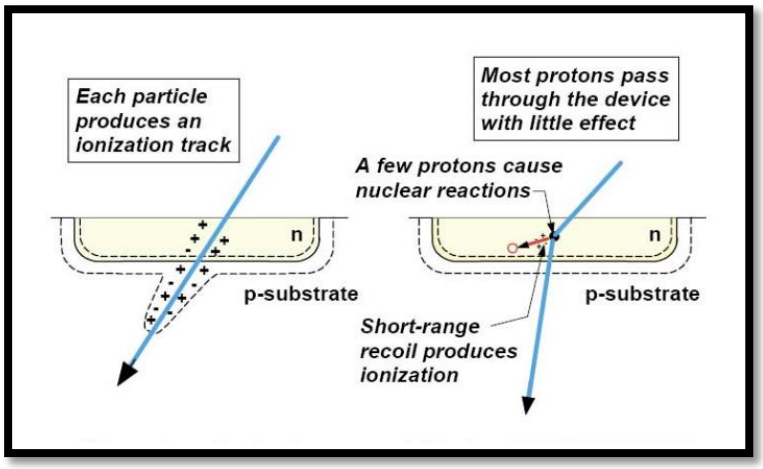


Figure 2.5 ionization effects of a heavy ion (let) and proton particle effect (right) at a p-n junction

The major concern in space applications are SEUs, however, multiple bit upsets (MBU) are becoming a serious matter to address due to the development of nano-technology. A MBU is caused when a single high energy ion passes through the silicone and energises two or more adjacent memory cells [18].

A single particle does not always result in a temporary error in the digital circuit; it can sometimes have a long lasting and damaging effect. The most common destructive error to occur in electronics is called a Single event latch-up.

Single Event Latch-up is a condition where the parasitic PnP structure in CMOS is latched to a high current state. This can either be destructive or non-destructive. In the non-destructive case, the affected device will have to have power recycled to restore normal operation. Long term high current will eventually result in failure [19].

2. 3 Device vulnerability

To test how vulnerable a device is to SEE's, one has to take note of two parameters: Linear Transfer Energy (LET) and cross section (σ) [20].

LET, also known as stopping power, is a measure of the energy transferred to the device per unit length as an ionising particle travels through a material. The common unit is MeV-cm2/mg of material. For MOS devices the material is silicone. LET is also known as the mass stopping power of the particle. The LET threshold (LETth) is the minimum LET to cause an effect.

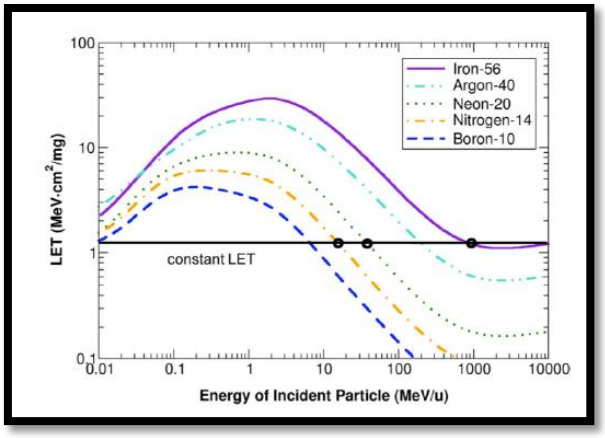


Figure 2.6 LET in silicone vs various particle energy [17]

Figure 2.6 illustrates the LET in silicone versus various particle energy. It can be noted that when the energy of the specific ion increases above 1 MeV/u the LET decreases, alternatively when the particles energy decreases to about 1 MeV/u, the LET value increases, as the particle slows down in the material it begins to lose energy more rapidly. However, as the energy of the particle decreases below 1 MeV/u the LET decreases since the particle cannot lose energy as rapidly simply because it does not have a lot of energy to begin with. This graph indicates that the higher the energy of a particle the lower the LET values are and the particle will thus deposit less charge in the materials.

This suggests that particles with high energy have a minimal possibility of causing a SEEs. This, however, does not suggest that the SEE studies at these high energies can be excluded. Protons

and neutrons can produce significant upset rates due to indirect interactions. The particles can produce secondary particles when they undergo inelastic nuclear collisions[2] with materials in and surrounding a device. These collisions produces heavier and/or less energetic secondary particles that can have a higher LET than the primary particle [21] [22] [23] [24].

To measure how sensitive a device is to SEE's, the SEE cross section (σ) formula is used as shown by equation 2.1.

Equation 2-1 – SEE cross section

$$\sigma_{dev} = \frac{Number\ of\ single\ event\ recorderd\ during\ the\ experiment}{Number\ of\ particles\ per\ Cm^2}$$

This equation yields the SEE sensitivity of the device. The cross section is calculated from experimental measurements by dividing the number of events (errors) observed, by the particle fluence and has units of cm^2/device.

When both the LET of the particle and the cross section of a device have been calculated, a specific graph can be generated which is commonly used in displaying the probability of an SEE event occurring in a specific device. This graph is shown in Figure 2.7

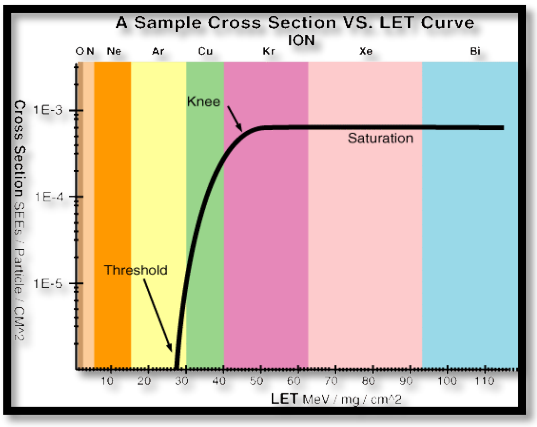


Figure 2.7 Example of cross-section vs LET [25]

[2] An inelastic collision is a collision in which kinetic energy is not conserved

This graph is constructed by plotting the SEE cross section versus the LET, more than one data point is needed to plot this graph and this is done by varying the MeV of the particle and calculating the cross section at that specific particle energy.

There are two important values that must be noticed. The first is the LET threshold; this is when the first LET will cause an event in this system. The second is the saturation cross section, which corresponds to a maximum LET, so even if the LET increases, no more events will take place.

2.4 Current testing facilities

This section will take a closer look at the testing facilities available for SEE testing of microelectronics around the world. As 90% of the cosmic rays are comprised of protons, only proton facilities will be investigated [26]. Any cyclotron facility could be used for testing of SEE; however, not all have a permanent set-up dedicated to SEE testing. The various facilities will be explained and the advantages and disadvantages will be described. These observations will be kept in mind for the designing of the SEE test set-up for iThemba Labs.

SEE testing with proton beams can be conducted in two different conditions namely open air or in a vacuum chamber, each having its own advantage. Testing electronics in open air gives the advantage of easily accessing the device under test (DUT) during testing since there is no vacuum to switch off; however, there is a significant loss of energy of the proton beam as it travels through the air towards the DUT. A vacuum on the other hand, gives a more stable proton beam. The biggest disadvantage is that it takes about 5 hours for a vacuum chamber to pump down, therefore, if anything goes wrong during testing another 5 hours will be needed repeat the test.

2.4.1 The Base Facility

The Berkeley Accelerator Space effect (BASE) facility is located in California USA. This facility makes use of a 88 inch Cyclotron and can reach energies up to 200MeV and as low as 50MeV. Only open air testing is available at this facility [27]. The set-up can be seen in Figure 2.8.

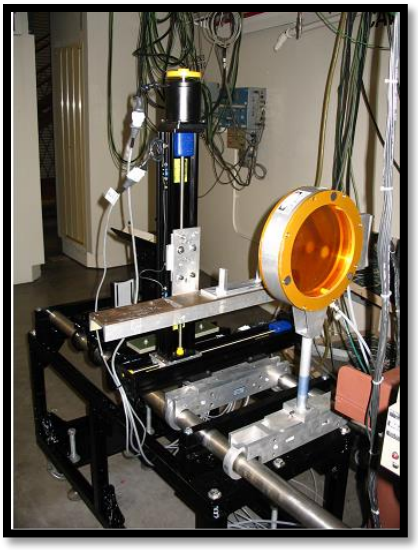


Figure 2.8 SEE testing Set-up at the base facility [27]

This mechanical set-up can only move in a vertical and horizontal direction. The problem with this set-up is that the beam size produced is about 7cm in diameter and the uniformity is less than 40%, this is due to the copper that is used between the beam and the target to degrade the beam [27]. This can make analysing results problematic.

2.4.2 Indiana University

Another facility that has a set-up dedicated to SEE testing is the Radiation Effects Research Program (RERP) at the University of Indiana. The RERP offers open air testing with a proton beam up to 200MeV. The diameter of the beam can be from 2cm to 30cm. This facility charges R8200/hour for use of the beam [28].

2.4.3 Radiation Effects Facility

The Radiation Effects Facility (REF) is located in Texas, America. This facility is a division of Texas A&M University. This facility has the option for testing microelectronics in open air or in a vacuum chamber. Different ions can be produced by this facility ranging from 15MeV/u LET to 40MeV LET.

2.4.3.1 Vacuum Chamber

Figure 2.9 shows the vacuum chamber used by REF. Call-out A is used to hold a standard PCB board of 10cm by 10 cm. This configuration has four degrees of motion, X Y Z and theta. X and Y are the horizontal and vertical axis where Z is the axis perpendicular to them. Theta is the angle of rotation about the target chamber Y axis.

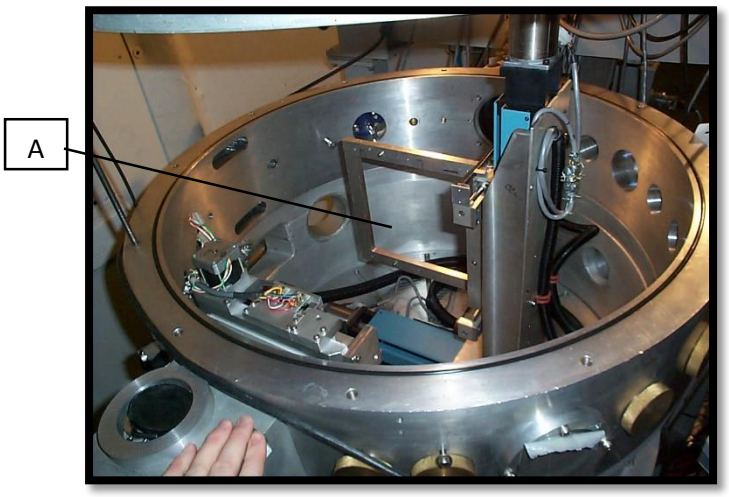


Figure 2.9 Vacuum SEE testing at REF] [29]

The advantage of this set up is the various degrees of motion it can produce, therefore, any part of the PCB board can be radiated with ease. The disadvantage is that only one PCB can be tested on a single test run and as mentioned earlier, it takes up to 5 hours to produce a vacuum.

2.4.3.2 Open Air testing

Their open air testing set-up as can be seen in Figure 2.10. It uses the same square PCB-holder and the mechanical set-up and has the same four degrees of freedom as the one in the vacuum chamber.

Figure 2.10 Open air SEE testing facility at REF [29]

The main advantage of this set-up is that boards can be changed or removed as soon as the beam is switched off. There is no 5 hour waiting period as there is with the vacuum set-up. Open air testing's biggest disadvantage is that since the proton beam is traveling through air it loses energy and it cannot be predicted, with confidence, what the energy of the proton is at the PCB.

2.4.4 Conclusion

Investigating these different testing facilities gave an indication of what should be focused on when designing a set-up that could be used at iThemba Labs. Observations that were made are:

1. All of the set-ups only catered for testing of one PCB at a time.
2. When open air testing is conducted, the diameter of the beam is quite large. When it is desired to test specific components on a PCB board this could be problematic. Since the beam is degraded by inserting copper between the beam and the PCB, the beam's uniformity is never more than 40% over a specific area, unlike vacuum testing where the uniformity is 100% [29].
3. All the set-ups are fixed.

From these observation it was concluded that a vacuum test would be the best option and that the structure should have the following features:

1. Able to test more than one PCB in a single test run.
2. Able to be easily removed from the chamber.
3. Able to have at least 3 degrees of motion.

2.5 Proton Testing Facility Available

The testing facility that will be used for this dissertation is the NRF iThemba Laboratory for Accelerator Based Sciences (LABS) situated in Cape Town. The facility has the most powerful accelerator in the southern hemisphere [30]. This accelerator is used for proton and neutron therapy for cancer patients as well as the production of isotopes. Nuclear physics research is also conducted here.

As this is the only facility of its kind in the southern hemisphere, beam time is a rare commodity. The Separated Sector Cyclotron (SSC) accelerates particles to energies up to 200MeV on Mondays and Fridays and the rest of the week the beam is kept at a constant 66 MeV for isotope production [30].

The testing for this project will be conducted in the A line scattering chamber shown in Figure 2.11, indicated by the yellow arrow.

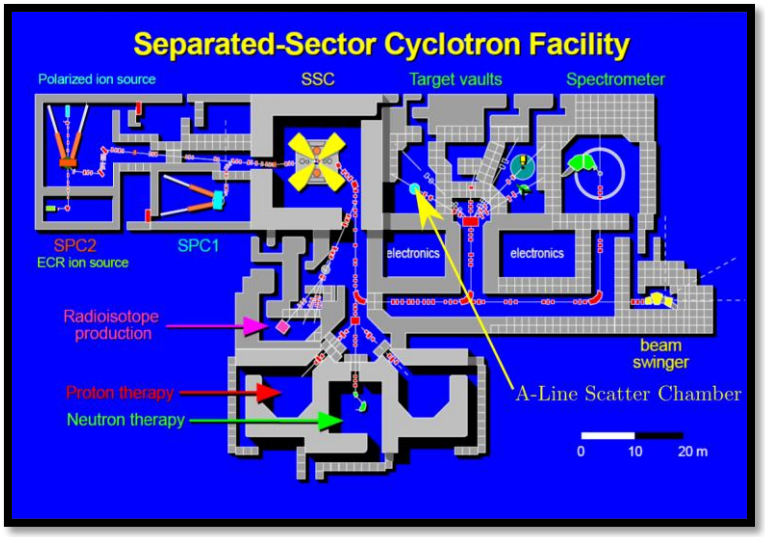


Figure 2.11 Floor plan of iThemba LABS [30]

2.6 SEU mitigation techniques

Radiation protection against SEE's can be done in many different ways. Devices can either be physically hardened during the manufacturing process or the device can be radiation-hardened by changing the circuit design at the logic level. This dissertation will investigate the different logic level techniques used to protect circuits from radiation exposure. These techniques are called mitigation techniques.

There are two different methods of mitigation, one is component redundancy and the other is SET filtering. The most common method used is redundant mitigation.

This section will explain the different mitigation techniques used in this dissertation to cope with the SEU and SET problem. These techniques will be illustrated and explained in detail. It is important to note that an overhead is always involved when attempting to mitigate SEU errors. This means that a sacrifice will need to be made in the form of either the cost, power consumption, or area.

2.6.1 Redundant Mitigation

This section will explain mitigation techniques that use redundant circuitry.

2.6.1.1 Dual and Triple Modular Redundancy

Triple Modular redundancy (TMR) is one of the most common and trustworthy mitigation techniques used to mitigate SEU errors. TMR is a widespread method and is applied in several applications aside from SEU mitigation [31]. Dual Modular Redundancy (DMR) is very similar to TMR where both make use of redundant components. DMR duplicates the circuit twice whereas TMR duplicates it thrice.

TMR replicates a circuit three times, the circuit outputs are then connected to a circuit called a majority voter. A majority voter is a circuit that accepts three inputs and the original input signal is carried through to the output as long as at least two of the circuit inputs are the same. This is illustrated in Figure 2.12.

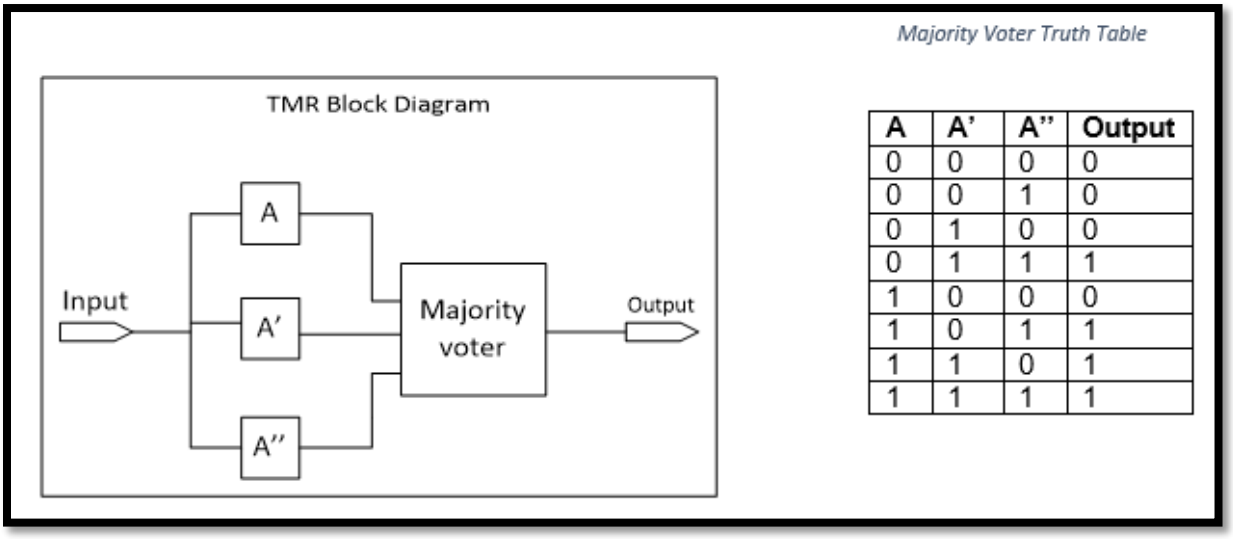


Figure 2.12 TMR Block Diagram

The method in the figure above is called local TMR as only the individual circuits have been duplicated. Another method of TMR is full global TMR, this is where the inputs, all the signals in the circuit and the majority voters with their outputs are tripled. This is the most complete form of protection against SEU errors. The downfall of full global TMR is that the area overhead and power consumption is at least three times more than the original circuit.

2.6.2 SET Filtering techniques

This section will describe the different SET filtering techniques that will be implemented in this dissertation. These methods correct the signal before it propagates to the memory elements, be it flip flops or latches. The two SET filtering techniques used in this dissertation are the Guard gate implementation and the AND-OR Multiplexer SET Filter.

2.6.2.1 Guard Gate

The first SET filtering technique makes use of the Guard Gate (GG) also known as the Muller C-gate designed by David E Muller in the 1950s [32]. The guard gate circuit consists of four Field Effect Transistors (FETs), two inputs and one output as illustrated in Figure 2.13. The inputs are connected to the upper or lower two MOSFETs, the output of the GG is connected to the connection between the two upper and lower two MOSFETs.

When the inputs are different, the output will float in a high impedance. In this case, the output voltage will maintain its value until the leakage current degrades it. When the two inputs A and B are identical, the Guard Gate acts like an inverter.

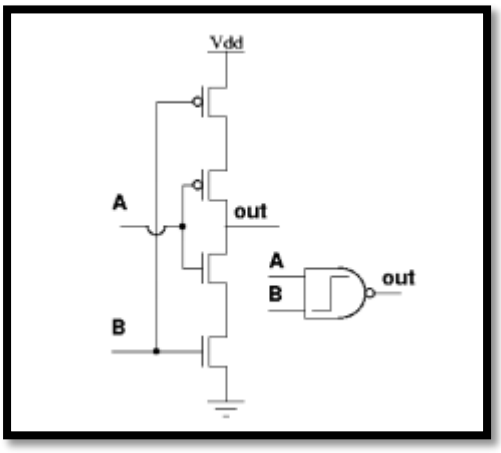
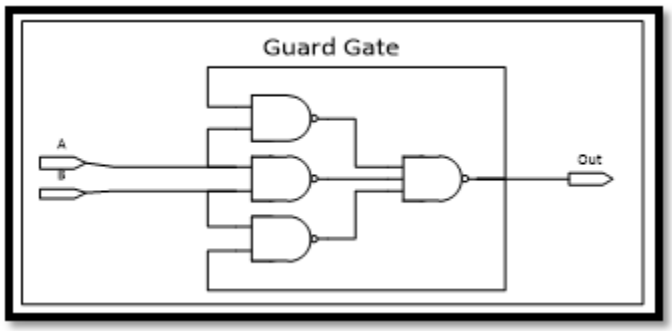


Figure 2.13 Diagram of the Guard-Gate

A different implementation on the GG cell was implemented [33] as shown in Figure 2.14. This implementation can either be used with a delay element were one of the input signals are delayed by an even number of inverters as in [34], or the combinational logic that is connected to the inputs of the GG could be duplicated as in was shown by Sana Rezgui, [33]. In this dissertation the duplication method as shown in [33] will be tested and analysed.



A	B	Y
0	0	0
0	1	Y*
1	0	Y*
1	1	1

Figure 2.14 NAND C elements filter with the Guard-Gate, with its truth table

The GG implementation consists of four NAND gates configured as shown in Figure 2.14. This is also known as the NAND C element filter [35]. The NAND C element filter idea was first thought up by Muller and Bartky in 1959. The NAND C element has a similar operation to that of the TMR with a majority voter. When one of the three combination logic elements experiences a SET error the majority voter will vote that error out the system. The NAND C element filter achieves a similar outcome when one signal experiences an error it will get filtered out, but with only two memory elements (i.e. DMR).

The truth table of the GG can explain this. When the two inputs are identical the output follows, as soon as one input differs the output will remain as it was. Therefore, an error will only be observed if both signals experience an upset. This circuit is able to handle SETs occurring either during the high or low states of a circuit line, but not both simultaneously. It would be desirable to have a single circuit that can handle a SET occurring during both the high and low states of a circuit signal.

2.6.2.2 AND-OR-Multiplexer SET filter

This is a new SET filter mitigation technique developed by Smith [36]. This mitigation method is known as the AND-OR multiplexer SET filter. This new technique involves the feedback from the output to do the voting. This filtering technique allows for one circuit that can handle a SET occurring both in the high and low states [36].

This SET filter technique can be implemented in two different ways. The first method uses a number of even inverters to cause a delay in the second signal as illustrated in Figure 2.17 and the second method makes use of DMR where the combination logic is doubled.

SET occurrences can be classified in two parts. The first part is that when the SET occurs, the signal is in a low voltage state and the second part when the voltage is in a high state. To prevent SET for both cases the SET filter has to be broken up into two separate parts. A common circuit that is used to remove glitches in a system is illustrated in Figure 2.15. A glitch[3] can also be thought

[3] A sudden surge of current

of being a SET; so hypothetically, if a specific circuit can prevent a glitch it should be able to prevent a SET of occurring.

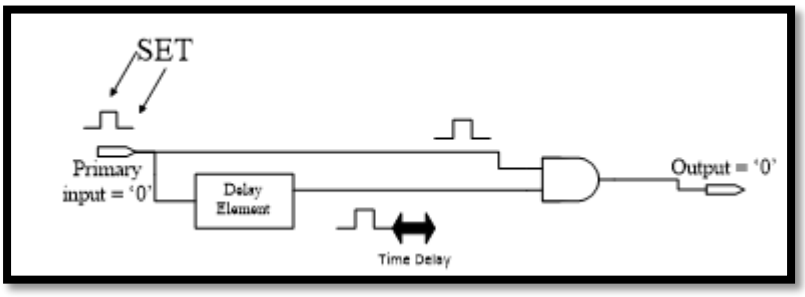


Figure 2.15 AND gate suppressor for primary inputs at logic 0

Figure 2.15 was originally designed to remove glitches on a falling edge of a clock pulse. This method can be applied to any circuit signal, being a clock signal or an internal logic signal. The circuit consists of one AND gate and one delay element consisting of an even number of inverters. The input gets branched off into two paths. One path includes a delay element and the other is clear. The delay element produces a delay signal with a greater time width than the anticipated SETs. The delayed signal and the non-delayed signal is then connected to the input of an AND gate. When the input of the AND gate suppressor is of a signal logic '0', as illustrated in Figure 2.15, a SET will cause the input to temporally change its state (i.e. '0' to '1'). However, due to the fact that the SETs have a shorter pulse width than that of the delay signal caused by the inverters in the delay element, the SET will arrive at one input of the AND gate and will be dissipated by the time the delay signal arrives at the other input of the AND gate. Therefore, the output of the AND gate will remain at logic signal '0'. It is important to note that when the input is of logic '0' the output is insensitive to SETs if and only if the pulse width of the SET is shorter than the pulse of the signal of the delay element. However, the opposite is true if the input signal is logic '1'. To remove SETs occurring when the input is a logic '1' a different circuit is needed as shown in Figure 2.16.

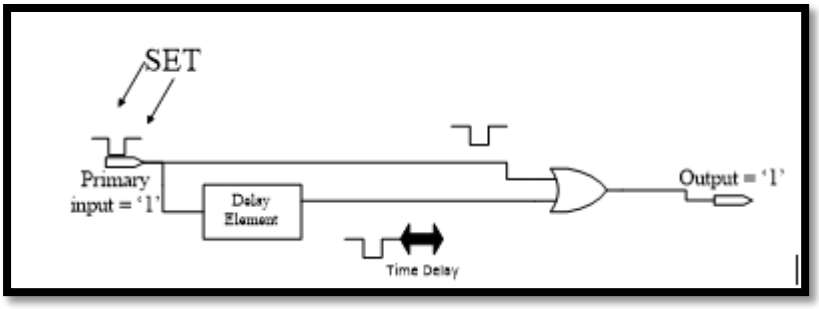


Figure 2.16 OR gate SET suppressor for logic value 1

This circuit has the same layout as the AND gate suppressor, the AND gate has just been replaced with an OR gate. When the input to this circuit is a logic '1' a SET which results in a temporary bit flip from logic '1' to '0' at the input and then propagates to the delay, will not change the state of the output because '1 OR 0 = 1'. This circuit is immune to SETs if the input is of logic '1' as long as the pulse width of the SET is shorter than that of the delayed signal. The opposite is true for this circuit when the input is of a logic signal '0'.

It is clear now that in order to protect a circuit from SETs when the input signal is a '1' or a '0' these two circuits needs to be combined in some way. A method was derived by Smith [36] where these two different methods of SET filtering are combined with a two-input multiplexer as shown in Figure 2.17.

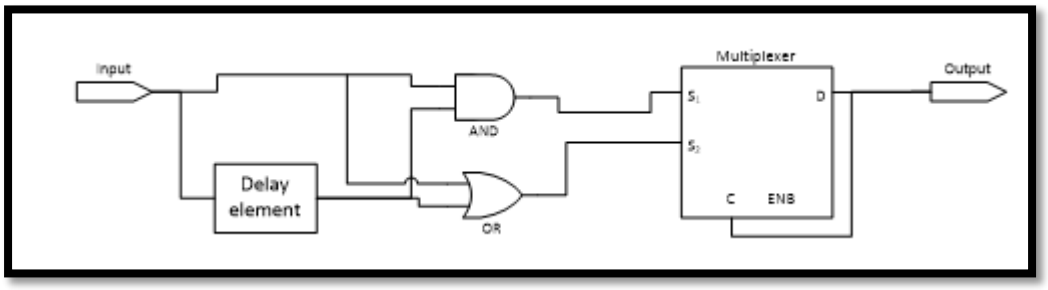


Figure 2.17 SET Suppressor Digital circuit

This circuit has the ability to be insensitive to SETs if the input signal is '0' or a '1'. A two-input multiplexer is used to select the AND gate when the output is '0' and selects the OR gate when the output is '1'.

Consider the condition where the signal input is of logic '1', both the AND gate and OR gate will output a logic '1'. Since both outputs of the gates are of logic '1' the multiplexer will also have an output of signal '1'. With the output of the multiplexer connected to its own selection line

indicated by port C in Figure 2.17, it will select either the AND gate or the OR gate circuit depending on the logic value of the multiplexer's output. In this example the output is of logic '1', therefore, it will select the OR gate circuit.

Now let's say that at some time in the future a SET occurs on the input signal and the logic gets flipped to a signal '0'. At one input of the AND and OR gates a '0' will be captured while the delay signal will filter out the SET. The AND gate will now output a signal '0', however, the OR gate will still output a signal '1', and as the previous output of the multiplexer was still a signal '1' before the SET occurred it will send a '1' to the selection line of the multiplexer and output the signal that was generated by the OR gate. This then propagates the logic signal '1' from the OR gate which is the correct signal that is required. Therefore, eliminating any SET with a pulse width that is shorter than that of the delay signal.

2.6.2.3 SET suppression circuit with DMR

One of the concepts that was discussed a lot in the previous section is the concept that the SET suppression circuit with the delay only eliminates the SET if the SET pulse width is shorter than the delay signal's pulse width. The drawback is that this reduces the speed of the system and one cannot be certain what the exact length of the SET pulse width will be, so the number of inverters needed to increase the delay is uncertain.

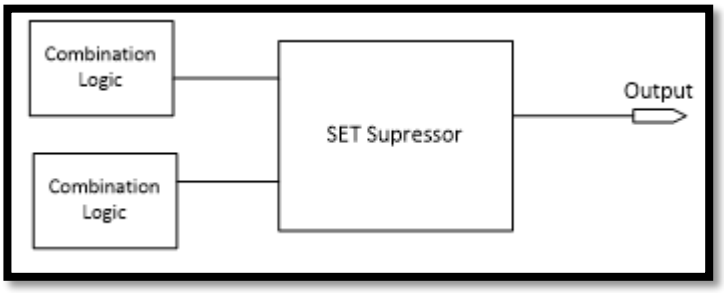


Figure 2.18 DMR Combinational logic connected to SET SUP

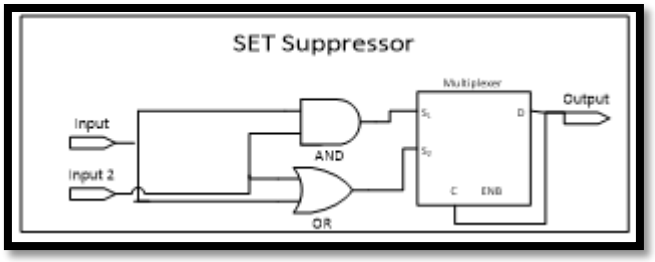


Figure 2.19 SET SUP DMR implementations

Therefore, another method was designed by Smith which uses the same principle explained previously. However, this method uses a DMR technique instead of the delay element implementation. Instead of delaying the input signal, the combination circuit is duplicated in order produce an output which is identical to the original output as shown in Figure 2.18. The reason behind this method was to eliminate the dependency on the SET pulse width. Since the combinational circuit is duplicated identically if an SET occurs in one of the combinational logic, the circuit will always be insensitive to SETs and independent of the SET pulse width. The only time a SET will be captured is when an upset occurs in both combinational logic components

2.6.2.4 Multiple Bit Upset Filter

All the various mitigation techniques explained so far in this dissertation are implemented to mitigate a single bit flip (SBU) in a circuit; however, there are occasions were more than a single node gets struck by a particle, this is known as a multiple bit upset (MBU).

To combat the MBU effect, the AND-OR multiplexer explained in section 2.6.2.2 can be extended to provide protection for multiple SETs in addition to single SETs as shown in Figure 2.20.

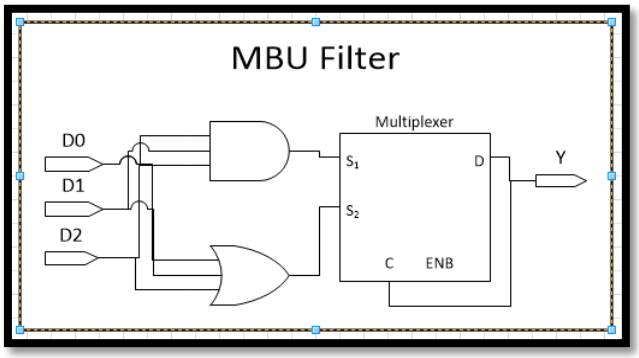


Figure 2.20 MBU Filter Design

This protection is achieved by using TMR and adding a three-input SET filter. With D0, D1 and D2 being identical redundant circuits, the three-input SET filter will suppress the SETs in any of the two adjacent circuits. Should two simultaneous SETs arrive at any two inputs of the three-input MBU filter, its operation will be identical to that of the two-input SET filter explained in section 2.6.2.2.

As this implementation has never been tested until now. All the observations and comments are theoretical. If it can be proven that this is a practical option to prevent a MBU, this implementation will be an ideal implementation to replace the voting circuit commonly used in TMR mitigation.

2.7 FPGAs

There are many different types of FPGAs that are manufactured by various companies. They consist of different technology, each with their advantages and disadvantages. Selecting the correct FPGA for the purpose of this dissertation is crucial.

2.7.1 Different types of FPGAs

There are two types of FPGAs, namely SRAM and Flash Based FPGAs. The majority of FPGAs produced today are SRAM based. Altera, Achronix and Xilinx are some of the companies which produce SRAM based FPGAs. Xilinx and Altera are the most dominant FPGA companies; these two companies comprise of approximately 90% of the FPGA market shares [37].

SRAM based FPGAs have their advantages and disadvantages. The one disadvantage is that SRAM based FPGAs are volatile components, which loses its data once the device gets turned off. This means that this type of FPGA needs to be reprogrammed on each start up [38]. The biggest disadvantage of these types of FPGAs is that they are easily affected by radiation particles that cause SEEs. An advantage of the SRAM based FPGA is that due to its architecture, its computational speed is fast when compared to the Flashed based FPGA [39].

Flashed based FPGAs make use of non-volatile memory in storing the program data; this implies that the FPGA will not lose the program that is currently on the FPGA once it has lost power. The biggest distributor of Flashed based FPGA's is Actel (now Microsemi). The flash memory that stores the configuration bits is resistant to upsets, unlike the SRAM FPGAs.

Table 2.1 Different effects SEE has on Flashed and SRAM FPGAs [40]

	0.25um FLASH	0.25um SRAM
SEU	Data Error	Data Error + Functional Interrupt
SET	CMOS	CMOS + Configuration Bits
SEL	CMOS	CMOS

Table 2.1 indicates the differences between SRAM and Flashed based FPGAs with regard to radiation effects. It can be noted that Flash FPGAs are less affected by particle radiation as it is only the CMOS in the flash based FPGA that gets affected. It is important to note that flashed based FPGAs are immune to Single Event Latch-ups (SELs) when the LET is below 96 MeV-cm^2/mg [41].

It is for this reason that a flashed based FPGA was chosen as the device under test for this dissertation. This dissertation focuses on the effects of SEU and mostly SET has on FPGAs, therefore, eliminating the possibility of SELs occurring is advantageous.

2.8 Applications of FPGA in Space

Much SEU testing has been conducted and various mitigation schemes have been tested before, however, most of these tests were conducted by using inverters as the combinational logic of the system [33] [42]. This is a great way for testing the effeteness of the mitigation schemes and calculating the cross sectional area of a device, however, when a FPGA is used for applications in space, it has a specific function to perform and it is never just a string of inverters.

This section will investigate the different applications a satellite may need to execute. An application will then be chosen to implement on a FPGA. This application will be protected with different mitigation techniques and an investigation will then take place to observe if these mitigation techniques can protect a practical application for a satellite.

Satellites can be classified into different sections depending on its purpose. The common types are:

- Earth observation satellites
- Communication satellites
- Navigation satellites
- Weather satellites

2.8.1 Earth observation satellites

As the name suggests, Earth observation satellites are used for observing. These are used not only to observe the earth but space as well. This is either done using cameras with low resolution cameras [43] or telescopes. An example of this type of satellite is the Hubble telescope used to capture images of space. An example of an image captured by the Hubble telescope can be seen in Figure 2.21.

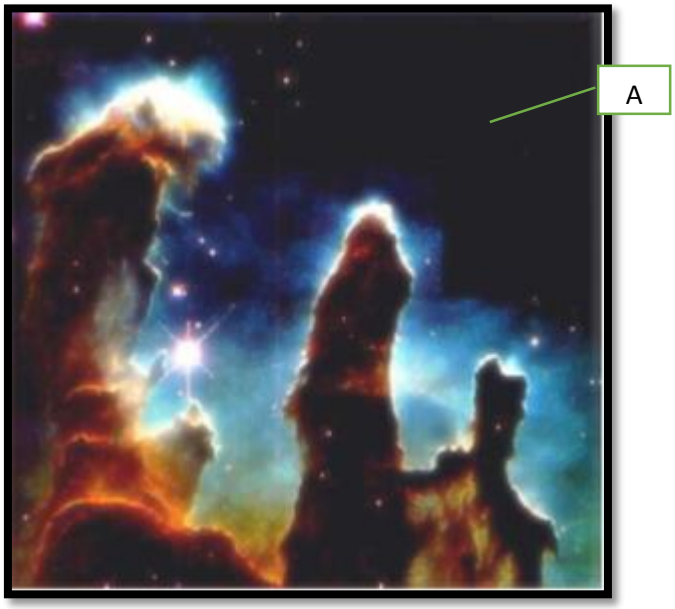


Figure 2.21 The birth of a star, picture taken by the Hubble telescope [44]

If one looks closely at this image, particularly the top right corner, an interesting phenomena can be observed. It can be seen that the top right corner is completely black; this is an example of an upset event that occurred in the system due to cosmic rays [44].

2.8.2 Communication satellites

Communication satellites are most probably the most used type of satellite in the world today. The first satellite in space called Sputnik 1 was a communication satellite [45]. The main purpose of a communication satellite is to receive signals (e.g. data, voice, TV) and relaying this signal back to the ground. With so many communication protocols used on satellites this will be a feasible application to test to investigate its operation under radiation.

2.8.3 Navigation satellites

Satellites have the ability to determine, with accuracy, geodetic position, speed and direction of a vehicle or an air craft [46]. Navigation satellites work by calculating the time it takes for its transmitted signal to be received. As the speed of light is a known constant this is easily calculated by the satellite system.

A practical example of the use of Navigation satellites is GPS. GPS is made possible by a network of 24 satellites named Navstar which orbit the earth every 12 hours. These satellites are moving in different directions, this allows the user on the ground to receive signals at different times. As soon as four satellites get in touch with the receiver, the receiver can calculate where the user is [47]. The 24 satellites used for GPS can be seen in Figure 2.22.

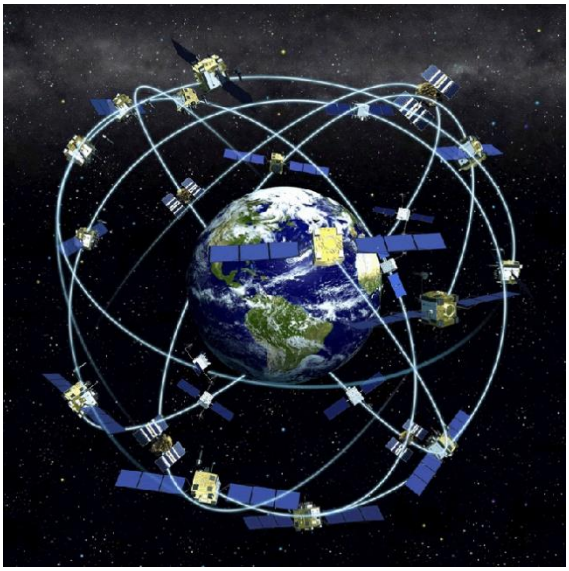


Figure 2.22 Navstar Navigation satellites [48]

Navigation satellites can also be affected by the harsh radiation environment in space. Earlier this year there was a solar flare that caused GPS accuracy to be out by $\pm$500m [49]. Therefore, GPS is a suitable application to be tested for upsets.

2.8.4 Weather satellites

Weather satellites form images by scanning the earth using instruments called radiometers. Radiometers normally consist of a telescope or antenna, a scanning device and detectors that detect visible, infrared, or microwaves for the purpose of monitoring weather systems around the world.

Electrical voltages are captured by the instruments and then digitised to be transmitted to the receiving station on the ground. This information is then sent to different weather forecast centres around the world to be analysed.

Weather satellites are launched into two different orbits, each of these orbits has their advantages and disadvantages for weather monitoring. The first orbit is called the geostationary orbit which can be seen in Figure 2.23, this orbit is at a very high altitude of about 36210 kilometres and is orbiting over the equator. This satellite orbits at the same speed as the earth [50].

The other orbit used for weather satellites is called the polar orbit, where the satellite is launched into a low altitude, around 804 kilometres. This satellite orbits the North Pole and South Pole

approximately every 100 minutes. Unlike the geostationary orbit, the polar orbit allows complete Earth coverage as the Earth turns beneath it.

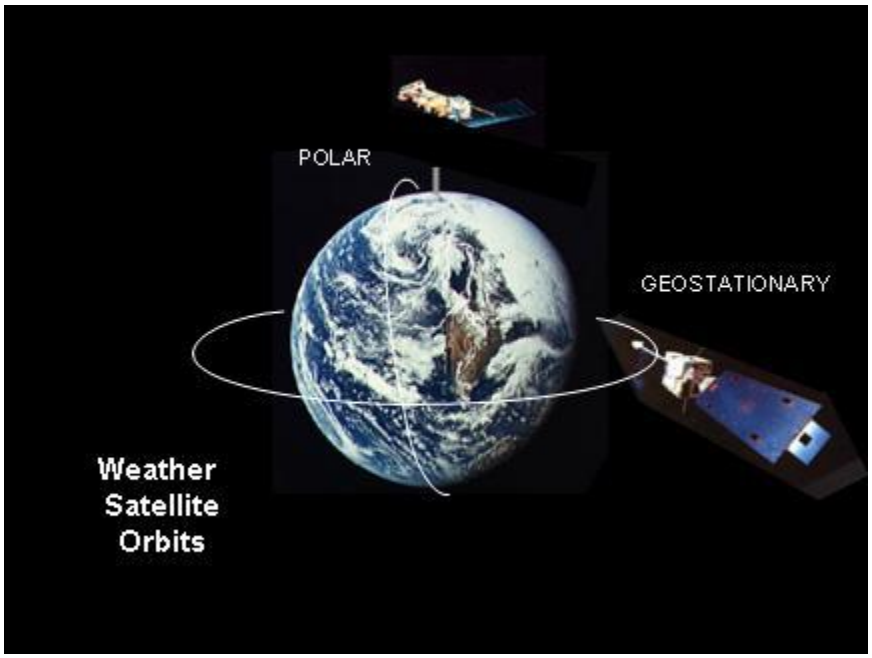


Figure 2.23 Weather satellites orbits [50]

Investigating all the possible satellite applications that are used in the world today, it was decided that an earth observation system will be implemented and tested for SEE effects. At the South African Space Association youth conference it was mentioned that South Africa wanted to invest more in the development of observation satellites.

The first observation satellite launched by South Africa was the SumbandilaSAT designed by SunSpace and CSIR. Its main function was to monitor and manage disasters such as flooding and oil spills.

In 2013 the Cape Peninsula University of Technology launched South Africa's first CubeSat. This satellite was launched as a communication satellite; however, on the CubeSat a small VGA camera was part of the pay load to capture images of earth. It was noticed that after time the images sent to Earth started to deteriorate, due to the radiation in space [51].

This information lead to the idea to create a VGA controller that is used by CubeSat's to display captured images on digital screens.

Chapter 3

3. Video Graphics Array implementation

This section will explain the fundamentals of the VGA controller on which SEE testing was conducted. The operation of a VGA will be explained and where in space a VGA controller is used.

3.1 Introduction

Video graphics array (VGA) was introduced in the late 1980's as a video display standard. This standard is still used today to display images on CRT and LED computer screens. There are three signals that send color information to a VGA monitor namely red, green and blue, which are analog signals[4]. There are also two directional signals namely horizontal and vertical synchronization which are TTL signals. By controlling these five signals, any picture can be created on a screen.

Many cameras use the VGA method to capture photographs and display them, including the cameras aboard satellites that take images from space [52]. This makes it an ideal operation to test for SEE's as it has a practical application in the space environment.

3.2 VGA cameras

As microelectronics have advanced in the last decade, the use of CubeSats have increased considerably [53]. The main purpose of CubeSats to date has been for educational purposes or for testing technology [54]. The development of CubeSats enabled satellites to use COTS camera components to capture images from space. However, these cameras are not immune to space radiation, like any other microelectronic device. They can be susceptible to permanent effects

[4] 0.7 to 1V

which may result in the image captured by the camera deteriorating over time or a soft error occurring and the wrong signal is transmitted to a computer screen.

A common camera-type that is used on CubeSats is the CMOS camera [55]. Capturing an image from a CMOS camera requires many different sub systems as is shown in Figure 3.1. The camera module sends the captured information to the capturing logic on the FPGA; this information consists of the position and colour of each pixel that constructs an image. This information is then stored in a memory buffer. The transfer of all this information between the camera module and the camera programming logic is executed by the I2C controller.

The information stored in the memory buffer is then displayed on a screen to form the image captured by the camera module. The controller which is used to take this information and correctly project it on a screen is called a VGA controller.

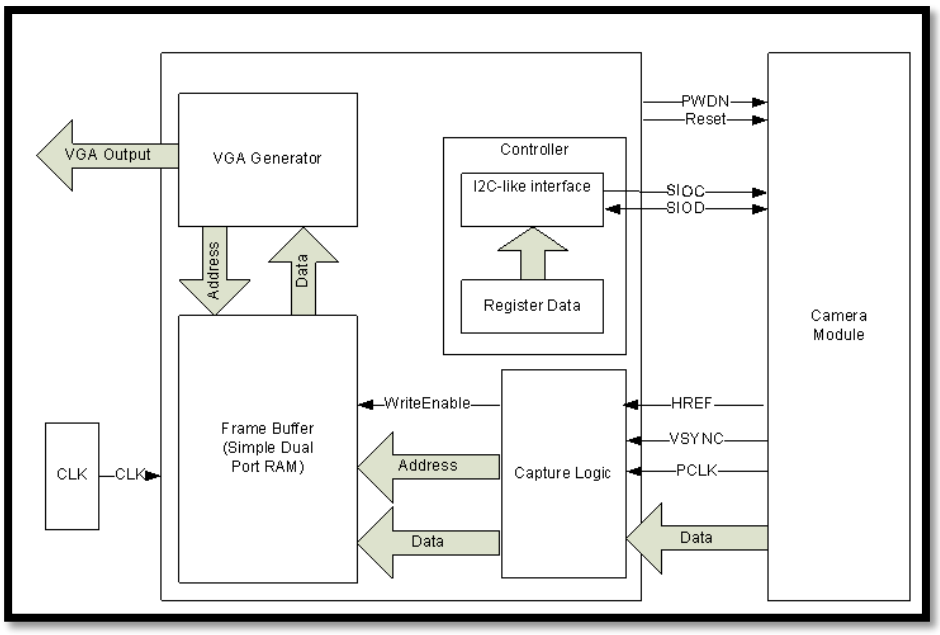


Figure 3.1 Diagram of the sub systems that is needed to capture an image from a CMOS camera [56].

This dissertation will only investigate the effect radiation has on the VGA controller because if a complete camera-module system is implemented, it will be difficult to determine where any SEU errors originate from. Since any error in the I2C controller, camera module or the memory buffer will propagate through to the VGA output and attempting to protect each sub-section from SEU affects will be tedious and not help in the objective of this dissertation. Therefore, a VGA controller

will be designed and implemented on its own to investigate if a VGA controller can be protected against radiation effects.

The next section will explain the required fundamentals of a VGA controller to enable the design of a VGA controller on an FPGA device.

3.3 VGA Fundamentals

A colour VGA video signal consists of 5 different signals: two synchronization signals (Hsync and Vsync) and three video signals (R, G, and B).

- HSYNC (Horizontal sync), indicates that the electron beam must restart at the screen's next scan line (starts a new line).
- VSYNC (Vertical sync), Makes the electron beam restart at the first screen's scan line (starts a new frame)
- R Red intensity
- G Green intensity
- B Blue intensity

Figure 3.2 illustrates the operation of the Hsync and Vsync. When the Hsync logic signal changes state from a logic level 1 to 0 the electron beam starts the next row on the screen, known as the scan line. This procedure is continued until the whole screen has been scanned. The Vsync signal changes state to indicate the procedure should restart at the top left corner of the screen.

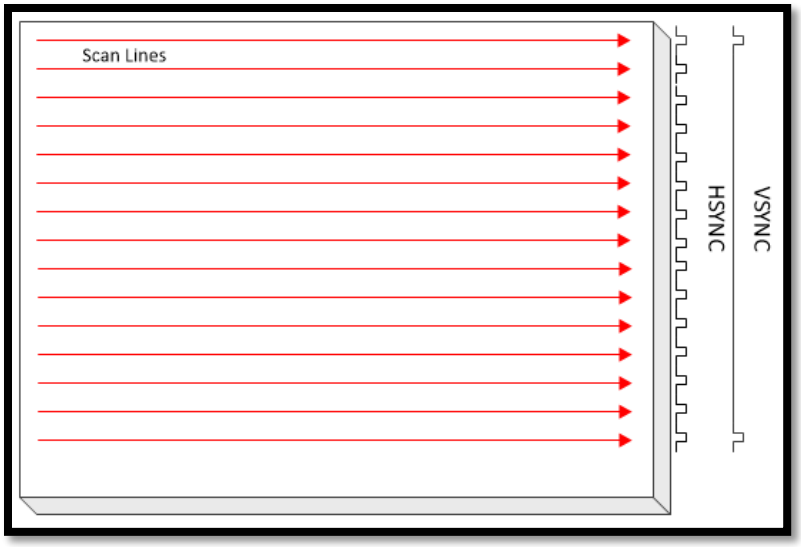


Figure 3.2 Figure illustrating the Hsync and Vsync process

The resolution of the screen is determined by the horizontal and vertical synchronization signals[5], whereas the colour of every pixel is determined by the red, green and blue video signals. Hsync and Vsync are TTL signals whereas the RGB signals are a voltage range, ranging from +0.7V (complete darkness) to +1V (maximum brightness). These three colour signals control an electron gun inside a cathode ray tube which makes the screen's phosphor produce a basic colour in a pixel. Any colour on the visual spectrum is the visual fusion of different intensities of brightness of these three primary colours.

In a 640 X 480 display for example, a frame of VGA video has 480 lines and each contains 640 pixels. The standard VGA controller operates at a 25 MHz clock.

3.3.1 Synchronization Signals

The Hsync and Vsync signals are used to control the timing of the scan rate. As mentioned previously, these two sync signals are digital signals, unlike the RGB signals which are analog. In other words, the Hsync and Vsync signals are either logic 0 or logic 1. The Hsync signal determines the time it takes to scan a row on the screen. The Vsync signal determines how long

[5] Example 640 X480 pixels

it will take to scan the entire screen. Manipulating these two digital signals and the RGB signals, images are formed on the monitor screen.

3.3.2 Synchronization Timing

The horizontal synchronization signal timing diagram is show in Figure 3.3 When the Hsync signal is inactive the signal is at a logic 1 value. A row scan begins when the horizontal synchronization signals goes low for $3.77\mu s$. A $1.79\mu s$ high on the Hsync signal follows this. The next step involves sending the data for the three colour signals, one pixel at a time, for 640 columns, which takes $25.17\mu s$. After the last column pixel (when the electron beam is in the bottom right corner of the screen), there is a $0.79\mu s$ of inactivity on the RGB signal before the next row scan will commence. The total time for one complete row-scan is $31.77\mu s$.

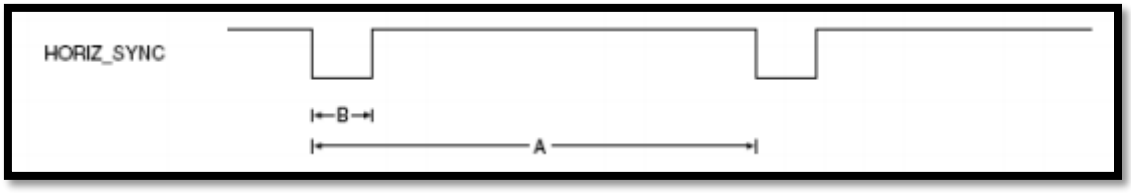


Figure 3.3 VGA Horizontal Sync signal construction

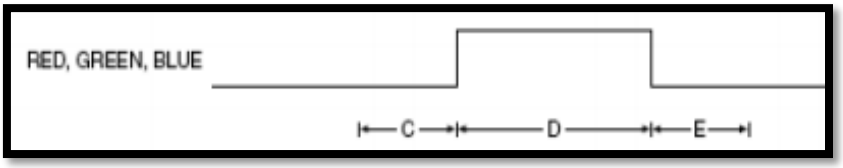


Figure 3.4 VGA colour signal construction

Table 3.1 Timing of the colour and Hsync VGA signals

Parameter	A(Total)	B	C	D	E
Time	$31.77\mu s$	$3.77\mu s$	$1.77\mu s$	$25.17\mu s$	$0.79\ \mu s$

The construction for the Vsync signal is identical to the Hsync signal, the only difference between these two signals is the timing of them. The $64\mu s$ low vertical sync signal resets the scan to the top left corner of the screen. Next, the 480 Horizontal sync row scans commence.

After the last row on the screen has been scanned, there is another $450\mu s$ delay before the Vsync signal goes low again to start another complete screen scan in the top left corner. To successfully display images on a monitor, simply get the Hsync and Vsync timing correct.

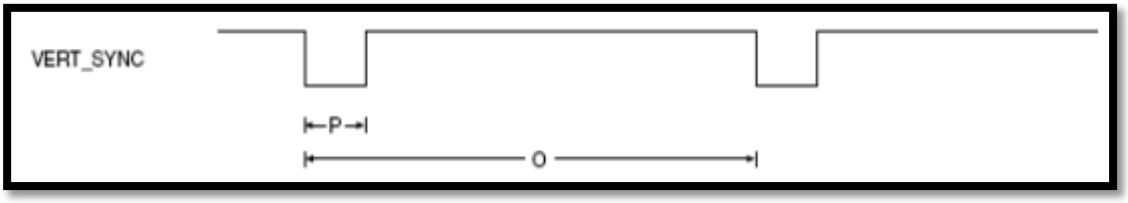


Figure 3.5 VGA vertical Sync signal construction

Table 3.2 Timing of the Vsync VGA signal

Parameter	O(Total)	P	O-P
Time	$16.6ms$	$64\mu s$	480 Horizontal scans

3.3.3 Clock cycles

The correct timing of the two synchronous signals is the most important aspect of a VGA controller. It can be achieved by using a suitable clock frequency. To achieve the standard 640 X 480 screen resolution, a clock frequency of 25 MHz is used. The period of the clock frequency is an important factor for when the VHDL code is written. This will be used to calculate the number iterations needed to correctly make the 5 VGA signals go high and low at the correct time. This is calculated to be:

$$T = \frac{1}{VGA_{Freq}}$$

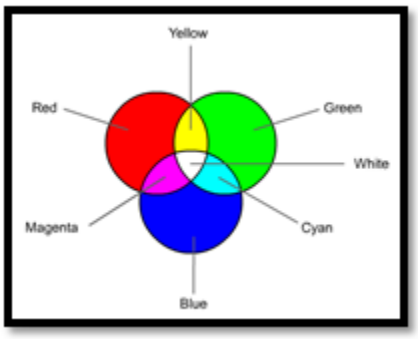
$$T = \frac{1}{25 \times 10^6}$$

$$T = 0.04\mu s \; per \; clock \; cycle$$

3.4 VGA Controller VHDL

Knowing all the fundamentals of the VGA controller, the VHDL code can now be written for the FPGA. Since there are many different VGA applications that are possible, ranging from writing letters on a computer screen [60] to writing a complex ping pong game [30]. It was decided to program a simple colour changing VGA VHDL program because it contains all of the fundaments

of a VGA controller needed for all VGA applications [8]. The program accepts three signals as the three colour inputs red, green and blue and outputs the desired colour on a computer. By varying the densities of those three primary colors, most colours are able to be displayed on the computer screen. The VGA controller that was developed in this dissertation makes use of a 3 bit[6] colour spectrum, therefore, only 8 colours are able to be displayed on the computer screen. Figure 3.6 illustrates the different colours that various combinations of red, green and blue can produce.



Colour	Red	Green	Blue
Black	0	0	0
Blue	0	0	1
Green	0	1	0
Red	1	0	0
Cyan	0	1	1
Magenta	1	0	1
Yellow	0	1	1
White	1	1	1

Figure 3.6 Colour representation of the 3 bit input

3.4.1 VGA Code

Figure 3.7 is a code[7] abstract from the VGAVHDL code. Using the period calculated in section 3.3.3 of $0.04\mu s$ the timing constants used in the VGA VHDL code can be calculated.

```
architecture rtl of vga_sync is
signal horiz_sync, vert_sync, pixel_clock_int,CLK25 : std_logic:='0';
signal video_on_int, video_on_v, video_on_h : std_logic:='0';
-- Horizontal and vertical counters
signal h_count, v_count : unsigned(9 downto 0):= (others => '0');
-- Horizontal Timing Constants
constant h_pixels_across : natural := 640;
constant h_sync_low : natural := 664;
constant h_sync_high : natural := 760;
constant h_end_count : natural := 800;
-- Vertical Timing Constants
constant v_pixels_down : natural := 480;
constant v_sync_low : natural := 491;
constant v_sync_high : natural := 493;
constant v_end_count : natural := 526;
```

Figure 3.7 VHDL VGA Timing constant

[6] $2^3 = 8\ possibilities\ of\ colour$

[7] Full code can be found on the CD provided in Folder VHDL code

These timing constants are used to program the operation of the Vertical Sync and the Horizontal Sync. These timing constants controls the signal construction, indicating when the digital signal should rise and fall. This can be seen in Figure 3.8.

```
--Generate Horizontal Sync Signal using H_count
if (h_count <= h_sync_high) and (h_count >= h_sync_low) then
horiz_sync <= '0';
else
horiz_sync <= '1';
end if;

-- Generate Vertical Sync Signal using V_count
if (v_count <= v_sync_high) and (v_count >= v_sync_low) then
vert_sync <= '0';
else
vert_sync <= '1';
end if;
```

Figure 3.8 VHDL HSYNC and VSYNC signal construction

The VHDL code for VGA controller was initially programmed on the Altera DE2 Development FPGA board. The FPGA board has an onboard VGA port which outputs the required signals described earlier, to a computer screen. These colours were programmed to change by toggling the three switches on the development board that represent the three primary colours. Once it was clear that the VHDL VGA controller was operational it was then programmed onto the Proasic3E development board as this is the board that will be used during SEE testing.

Once the VGA VHDL code had been successfully programmed and uploaded to the Proasic3E FPGA. An oscilloscope was used to observe if the signals were outputting the correct signals. It can be observed in Figure 3.9 that the desired results were achieved with the RGB and Horizontal sync signals. The oscilloscope signal is identical to the desired signal explained in section 3.3.2.

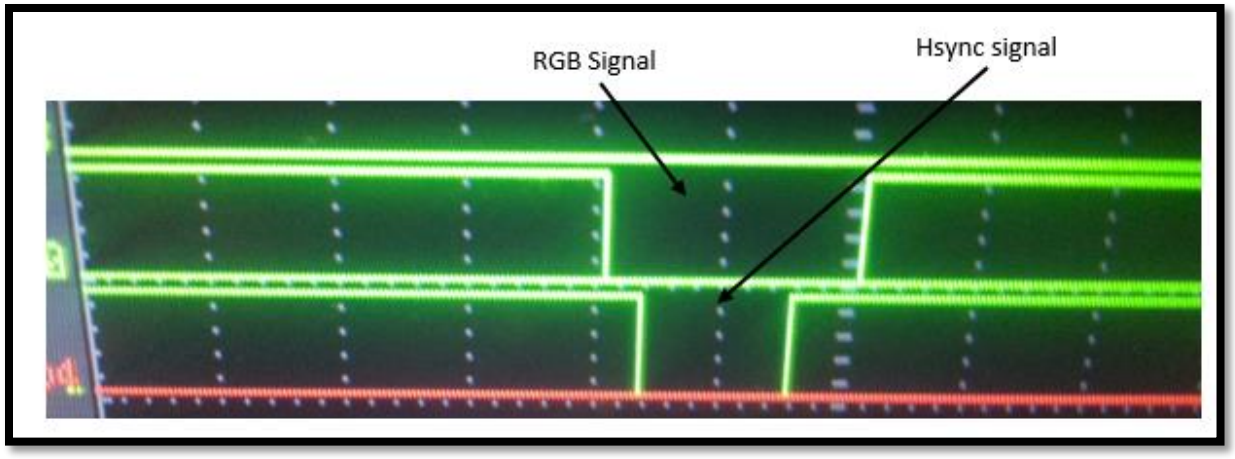


Figure 3.9 Oscilloscope digital signals RGB and Hsync

The vertical sync signal is the slowest signal of the three signals. Its logic only changes every 524 horizontal cycles. Figure 3.10 indicates that the FPGA gave the desired vertical sync signal.

Figure 3.10 Oscilloscope digital signals of Vsync (bottom) and Hsync (top) signal

3.4.2 VGA NETLIST

Once the basic VGA VHDL code was written, different mitigation schemes were implemented into the VGA code to test their effectiveness in preventing SEE errors in the system. Since the VGA VHDL code was written in behavioral code, as seen in the code extract in Figure 3.11, it is

not possible to insert any mitigation in the VGA behavioral code. The behavioral code had to be converted to its net list form, the net list broke the VGA VHDL program into its most basic components (i.e. AND gates). This gave the ability to add components to the code and to duplicate them. This is a requirement to mitigate the VGA system. To convert the behavioral code to the required net list FPGA Compiler II 3.8 had to be used.

The conversion can be seen in the code snippets represented in Figure 3.11 and 3.12.

```
wait until(pixel_clock_int'event) and (pixel_clock_int='1');

    if(rst = '1') then
      h_count <= (others => '0');
      v_count <= (others => '0');
      video_on_h <= '0';
      video_on_v <='0';
          else
```

Figure 3.11 VHDL behavioral code abstract

```
red_out_reg : DFC22 port map (reset, Q => N_red_out , C
vert_sync_reg : DFC22 port map (reset, Q => vert_sync ,
horiz_sync_reg : DFC22 port map (reset, Q => horiz_sync
green_out_reg : DFC22 port map (reset, Q => N_green_out
```

Figure 3.12 VHDL net list code abstract

The VHDL net list is used to implement all the mitigation implementations that will be tested. These mitigation methods were described in Chapter 2. In order to implement the necessary mitigations schemes i.e. TMR, DMR etc. The VHDL net list code has to change accordingly. For TMR all combinational logic and flip flops need to be tripled and then connected to their respective majority gates. Doing this manually would take a long time because every component and signal will need to be tripled or doubled for TMR and DMR respectively. A C# program was written to generate the necessary VHDL code for the DMR and TMR duplications. The user interface can be seen in Figure 3.13.

Figure 3.13 C# interface for generating DMR and TMR

The C# code triples or doubles all the components and their respective signals depending on what is required by the user. The user will enter either a 1 or a 2, depending if TMR or DMR[8] is needed to be generated. Once a choice is made then the generated VHDL code will be written to a text file.

The DMR does not only double the components and their respective signals, but also inserts the AND-OR multiplexer in front of the flip flops. Figure 3.14 illustrates an abstract of the code when DMR is selected.

```
Setsup_00 : SET_SUP Port map (vert_synca_0,vert_synca_1,SET_Signal0_0),
vert_sync_out_reg_0  :DFC22 port map (  reset, Q => N_vert_sync_out_0
Setsup_01 : SET_SUP Port map (vert_synca_0,vert_synca_1,SET_Signal0_1),
vert_sync_out_reg_1  :DFC22 port map (  reset, Q => N_vert_sync_out_1
Setsup_02 : SET_SUP Port map (vert_synca_0,vert_synca_1,SET_Signal0_2),
vert_sync_out_reg_2  :DFC22 port map (  reset, Q => N_vert_sync_out_2
Outgate_0 : MAJ_GATE port map (N_vert_sync_out_0,N_vert_sync_out_1 ,N_v
Outgate_1 : MAJ_GATE port map (N_vert_sync_out_0,N_vert_sync_out_1 ,N_v
```

Figure 3.14 Abstract of VHDL DMR code

The flip flops are tripled along with the SET suppressors. The combinational logic that was doubled is connected to the inputs of the set suppressor as can be seen by components Setsup_00, Setsup_01 and Setsup_02 in the DMR code extract. This C# code was also used for the Guard Gate implementation since the only change that was needed was changing the SET_SUP port map to a GG port map as can be seen by Figure 3.15.

[8] Full code can be found on CD under C# program

```
Setsup_240 : GG Port map (C119_N40_0,C119_N40_1,SET_Signal26_0);
h_count_reg_7_0  :DFC22 port map ( reset, Q => h_count_7_0  ,CLK =>
Setsup_241 : GG Port map (C119_N40_0,C119_N40_1,SET_Signal26_1);
h_count_reg_7_1  :DFC22 port map ( reset, Q => h_count_7_1  ,CLK =>
Setsup_242 : GG Port map (C119_N40_0,C119_N40_1,SET_Signal26_2);
h_count_reg_7_2  :DFC22 port map ( reset, Q => h_count_7_2  ,CLK =>
Outgate23_0 : MAJ_GATE port map (  h_count_7_0, h_count_7_1 ,  h_cou
Outgate23_1 : MAJ_GATE port map (  h_count_7_0, h_count_7_1 ,  h_cou
```

Figure 3.15 Abstract of DMR with Guard gate implementation

An abstract of the TMR code can be seen in Figure 3.16.

```
h_count_reg_6_0  :DFC22 port map ( rst, Q => h_count_6_0  ,CLK
h_count_reg_6_1  :DFC22 port map ( rst, Q => h_count_6_1  ,CLK
h_count_reg_6_2  :DFC22 port map ( rst, Q => h_count_6_2  ,CLK
Outgate22_0 : MAJ_GATE port map (  h_count_6_0,  h_count_6_1 ,
Outgate22_1 : MAJ_GATE port map (  h_count_6_0,  h_count_6_1 ,
Outgate22_2 : MAJ_GATE port map (  h_count_6_0,  h_count_6_1 ,
```

Figure 3.16 Abstract of VHDL TMR code

For TMR the combinational logic, flip flops and the majority gate are tripled as required.

Once all the different VHDL code for the different implementations were generated, it was required to fill up the memory of the FPGA to increase the probability of experiencing an SEE upset. The method used to fill up the FPGA memory with the VGA program involved first making the VGA a component and then duplicating the VGA component as many times as necessary to achieve the required chip percentage. Once duplicated, the output of each VGA component were ANDed together such that all the signals are outputted to one output as illustrated in Figure 3.17.

Another C# program was written to achieve this duplication process. The user interface can be seen in Figure 3.17.

```
Please enter the name of the file
VGA
File created
Enter the number of insatnces for VGA default
10
Enter the number of insatnces for VGA DMR
20
Enter the number of insatnces for VGA TMR
3
```

Figure 3.17 C# interface to fill FPGA memory

The user must enter how many instances of a certain implementation are required. Once entered, the C# program will generate the required number of instances for each implementation that is required to fill the FPGA chip to a certain percentage.

This C# program duplicates the same instance twice to use for comparing two signals. This is explained in Figure 3.18.

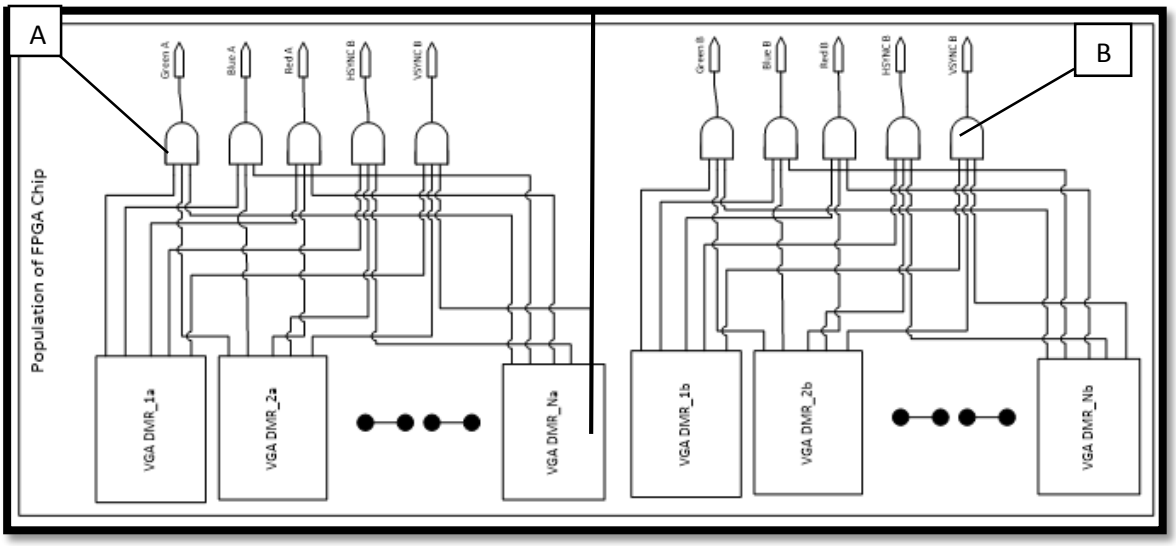


Figure 3.18 comparing two duplications of VGA implementations

VGA DMR is duplicated N times on one part of the FPGA as indicated by call-out A, the exact same duplication is done on another part of the FPGA as indicated by call-out B in Figure 3.20. Each duplication outputs the five VGA signals: red, green, blue, HSYNC and VSYNC. The same signal of each component is ANDed together so that there is only one output for each signal-type on half of the FPGA (i.e. Callout A and Callout B). Therefore, there will be a green signal outputted in call out A and another green signal in call out B, these signals will then be compared to observe any errors. This is done for each of the five signals. A code abstract can be seen in Figure 3.19 of how this was implemented in VHDL code. Further details of how the errors were counted will be explained in Chapter 4 and Chapter 5.

```
TVGA0: vga_sync_TMR port map  (Tclock,'1',
TVGA1: vga_sync_TMR port map  (Tclock,'1',
TVGA2: vga_sync_TMR port map  (Tclock,'1',
TVGA3: vga_sync_TMR port map  (Tclock,'1',
TVGA4: vga_sync_TMR port map  (Tclock,'1',
TVGA5: vga_sync_TMR port map  (Tclock,'1',


TVGA2000: vga_sync_TMR port map  (Tclock2,
TVGA2001: vga_sync_TMR port map  (Tclock2,
TVGA2002: vga_sync_TMR port map  (Tclock2,
TVGA2003: vga_sync_TMR port map  (Tclock2,
TVGA2004: vga_sync_TMR port map  (Tclock2,
TVGA2005: vga_sync_TMR port map  (Tclock2,
```

Figure 3.19 Duplicating VGA components

Figure 3.19 indicates how the VGA implementation was duplicated. TVGA0 to TVGA5 is the implementation that is implemented in call out A and TVGA2000 to TVGA2005 is the VGA implementation on call out B, it can be noticed that this VHDL extract duplicates the VGA TMR implementation. All 6 VGA implementations are duplicated and compared in this way.

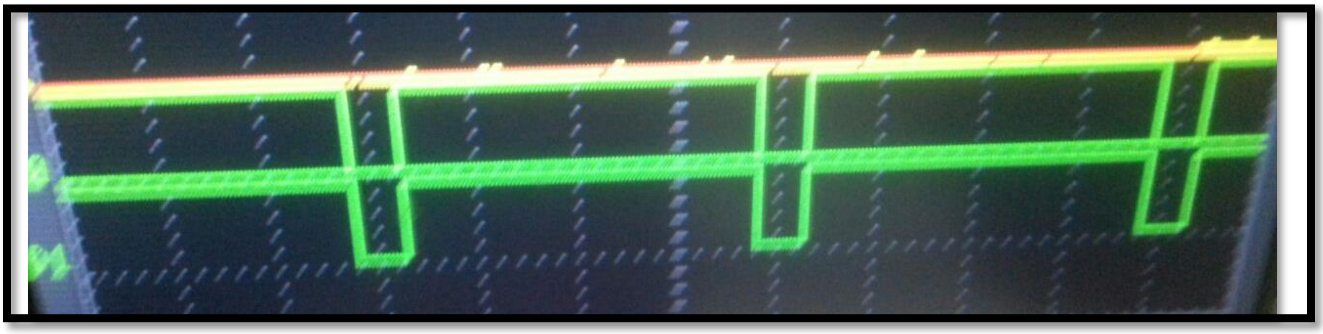


Figure 3.20 Two identical VGA Hsync signals used to observer errors

Figure 3.20 indicates the two identical signals that is produced by the code extract. These two signals are compared to one another and if there is any change in one of these signals an error will be counted.

Chapter 4

4. Experimental set up and Test Methodology

In this chapter the desired mechanical set-up and the test methodology will be discussed. First the testing facility will be described and an explanation will be given about what mechanical requirements are needed to design and build a successful SEU testing facility. Finally the completed set-up will be illustrated and explained.

4.1 SEE testing facility

The tests were performed at NRF iThemba labs at their separated-sector cyclotron facility using the A-Line scattering chamber. The facility can give a wide range of proton energies up to 220MeV. Unfortunately due to the congested scheduling at iThemba labs only 66Mev could be used for testing of SEE in the FPGA chip.

Figure 4.1 The A-Line scattering chamber at iThemba labs

4.1.1 Mechanical.

One of the aims of this dissertation was to design and build a mechanical structure that can be used regularly by researchers intending to test different electronics radiated by a proton beam. In this section this mechanical structure will be described and the requirements that the mechanical structure has to meet will be explained. To simulate the ideal testing conditions a mechanical structure needed to be designed so that accurate and precise movements of each DUT can be possible. Due to the complexity of the mechanical design, it was split up into two sections. This dissertation will concentrate on one section and give a brief overview of the whole system. The second section will be covered by another Masters student, Stefan Van Aardt.

4.1.1.1 Requirements

Through analysing different research facilities in Chapter 2 and communicating with different researchers from Stellenbosch University and iThemba, a set of requirements were established based on their suggestions.

The requirements are summarized as follows:

1. **The mechanical structure must be able to test multiple DUT for a single vacuum set up**

   Before any tests can commence at the A-chamber the vacuum must first be created. It takes approximately 5 hours for the vacuum to be created. Therefore, it will be ideal to cater for multiple DUTs under a single vacuum.

2. **It must be possible to select a single DUT for irradiation.**

   As described in requirement 1, many DUTs must be able to be tested in a single vacuum. A method must be created that any one of them can be irradiated at any time, by placing them in line with the proton beam.

3. **The angle of incidence of the selected DUT must be adjustable.**

   According to [57] the change of the angle of incidence of the DUT increases or decreases the probability of an SEE to occur. The mechanical set-up must be designed and build such that it can change the angle of the DUT.

4. **The system must be able to move in a vertical direction**

   The idea of this set up is not only to create a testing facility for SEE testing, but also for any other electronics testing. A method to irradiate any part of a PCB will be ideal (therefore being able to move the DUT up or down will be desired).

5. **All components and materials must be selected to handle vacuum and radiation environments**

   It is important to take special care of choosing the mechanical and electrical components to be able to operate inside a vacuum chamber.

All these requirements were met and will be indicated and explained in the next section.

4.1.1.2 Mechanical set up available

The mechanical system that was designed had to be interoperable with the mechanical system which is already in place at iThemba Labs.

Inside the vacuum chamber, which can be seen in Figure 4.2, there are two rotatable arms. These arms are a fixed structure inside the chamber which iThemba uses for their tests when required. Since iThemba lab's researchers use this chamber for their own experiments, the mechanical design must be done in such a way that it was possible to easily assemble and disassemble the structure.

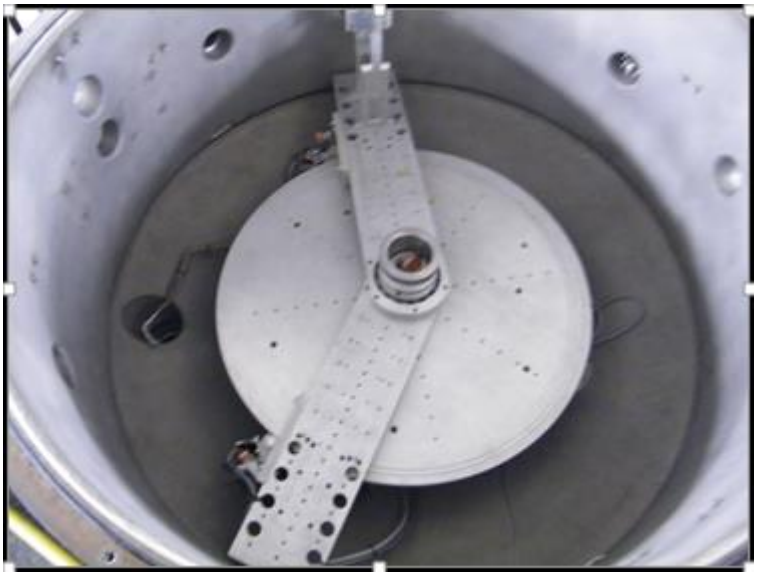


Figure 4.2 Inside the vacuum chamber

4.2 Mechanical System design

4.2.1 Complete System

This section will describe the steps taken in the design of the mechanical set-up. The mechanical system consists of three main aspects:

- The top section will grip the DUT's and has the ability to change the angle of incidence of each DUT independently.
- The top section will be able to move vertically, such that the particle beam can be aimed at any part of the DUT.
- The bottom part will rotate the whole system, such that each DUT can be moved into the path of the beam.

Figure 4.3 illustrates the full view of the drawing of the mechanical structure. The three aspects described above can be seen in callout A, B and C respectively.

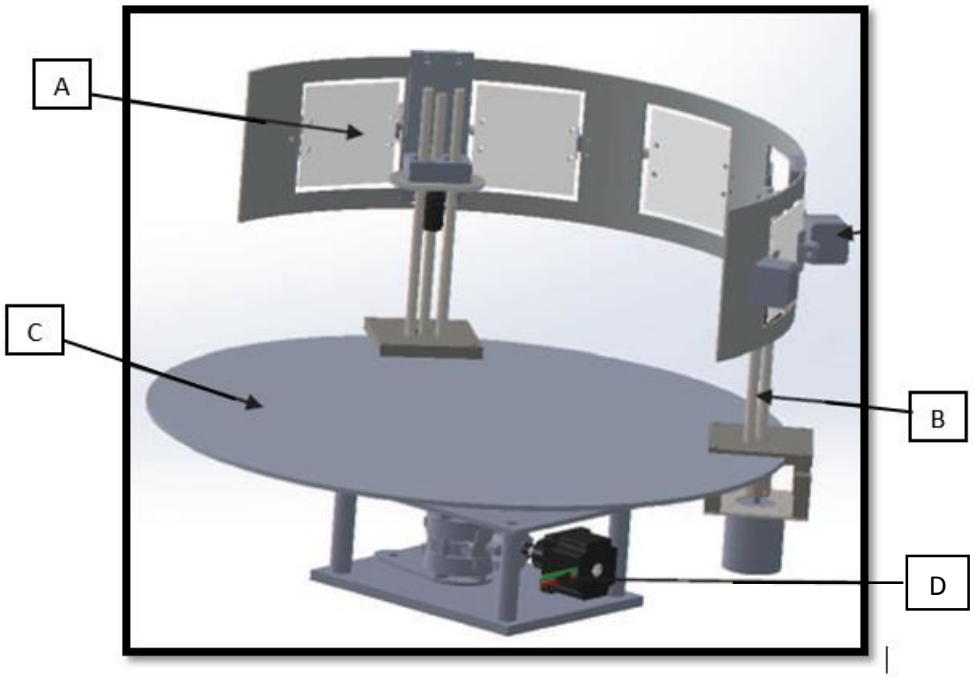


Figure 4.3 Complete mechanical set-up

The top section of the structure is where the DUT will be mounted ready to be irradiated (callout A). This section is connected to two lead screws that allow the top structure to move in a vertical direction which allows the proton beam to irradiate the top or bottom of any PCB (Callout B). The complete top section is then connected to the bottom disk that allows the top section to rotate either clockwise or anti-clockwise. This allows the mechanical structure to have the capability to radiate any DUT attached in the vacuum test (Callout C). The whole structure is rotated by a specially designed worm gear powered by a stepper motor (Callout D). A stepper motor was chosen as the choice of motor as the increments of rotation can be precisely controlled. As mentioned earlier in the dissertation, the top section was designed and analysed by Stefan van Aardt. This dissertation will discuss and analyse the rotating section in more detail below.

4.2.2 Bottom Section

The bottom section of the design is responsible for the rotation of the entire system, which enables each DUT to be radiated. This section is broken into different aspects to achieve the best design. The first aspect is having a solid base which would be able to support the weight of the

entire system, second is a method to easily rotate the structure with precision and lastly the stepper motor that will be used to rotate the system.

The base

The base that supports the structure is a simple mechanical structure that consists of two aluminum plates and 4 aluminum rods as is indicated by Callout A and Callout B in Figure 4.4.

This base is fastened to the rotary arm[9] that is already present inside the vacuum chamber as was shown in Figure 4.2

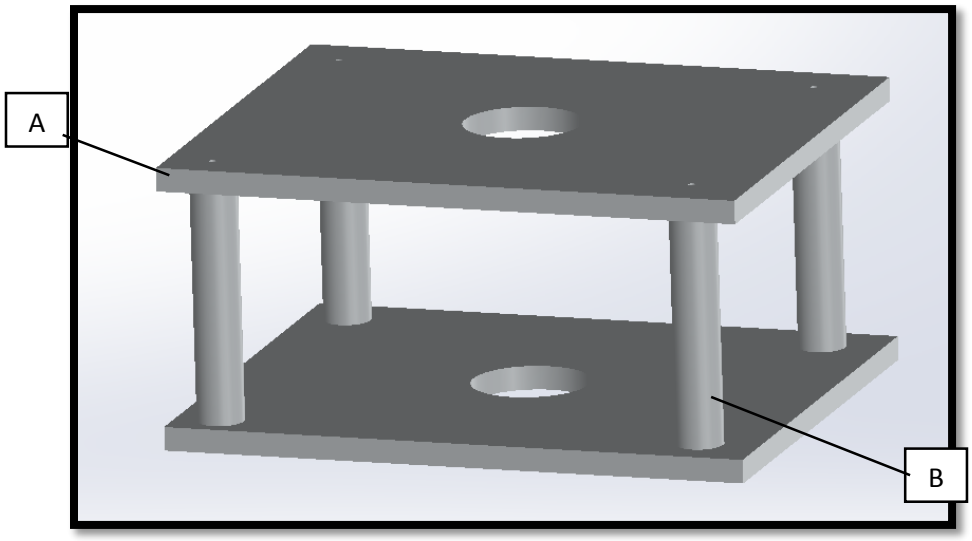


Figure 4.4 Mechanical base

The entire top section rests on top of this base. Therefore, the correct strength analysis is required to make sure that this base can support the weight of the top section. This was achieved by conducting force analysis on the top piece of the structure by using Solidworks 2012.

Figure 4.5 illustrates the von mises stress of the part under evaluation. The weight of the top structure was calculated to be no more than 12kg when aluminium was selected as the material.

Since the force from a 12kg is equal to 117.6N[10] a load of 200N was applied to the part to incorporate a safety factor. As indicated by Figure 4.5 the highest von misses stress is 77 630N/m^2

[9] Images of full construction onto the arm can be found on the CD that was provided.
[10] Gravity constant 9.8X12kg

which is much lower than the yield stress of 680 000 N/m^2 of aluminum. Therefore, this structure is robust enough to handle the weight of the structure.

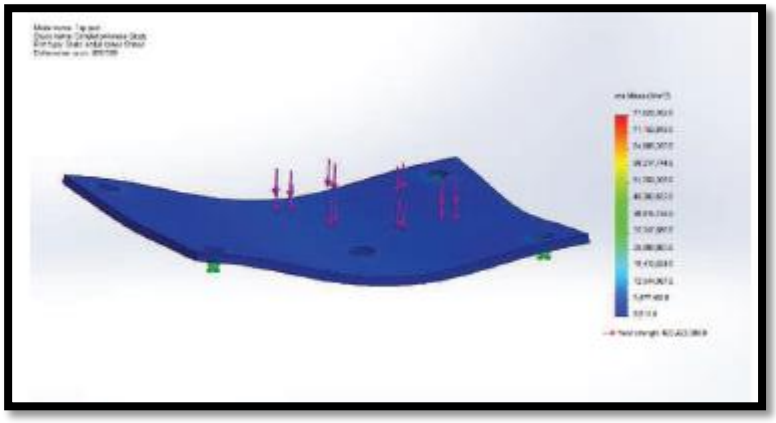


Figure 4.5 Von Mises stress analysis on the top plate of the base

Using Solidworks 2012 the deflection of the mechanical structure was calculated as well. Figure 4.6 illustrates the deflection of the part. A quick look at Figure 4.6 gives the illusion that the center of the structure has significant deflection, however when the part was examined in more detail, the deflection was only $4.24X10^{-2}$mm in the centre. This concludes that this system will not have any substantial deflection that will cause problems in the future.

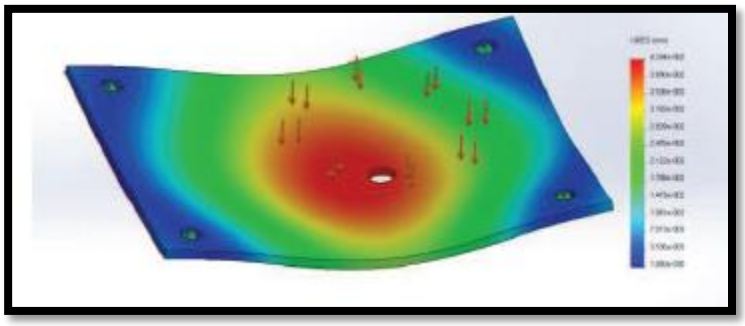


Figure 4.6 Deflection of the top plate of the base

4.2.2.1 Rotation Mechanism

As mentioned in section 4.2.1, one of the objectives of this mechanical system is to rotate the top section. Three different components were used to make the rotation of the system accurate and simple. The first component that was used was a thrust bearing. A thrust bearing is a particular

type of rotary rolling-element bearing. Like other bearings they permit rotation between parts, but they are designed to support a predominately axial load. This can be seen in Figure 4.7

Figure 4.7 SKF Thrust bearings that were used to ease the rotation of the top section

The thrust bearing that was chosen was the SKF51108[11]. This thrust bearing is small yet robust, measuring at 60mm in diameter and 13mm in height. It has the capability to handle up to 63kN of force which is more than adequate to handle the force required.

This thrust bearing was paced in-between two specially machined "holders" that allows the top section to rotate independently from the bottom section while still being connected to one another. The bottom holder shown in Figure 4.8 was designed to hold the thrust bearing while connected to the bottom base. The top holder shown in Figure 4.8 is rested on top of the thrust bearing, free to rotate. The top holder also has a shaft connected that goes through the centre the thrust bearing. This shaft is connected to the key shaft that is connected to the worm gear drive, as this shaft rotates the top section will rotate.

[11] Data sheet can be found on the CD in the Folder Data sheets

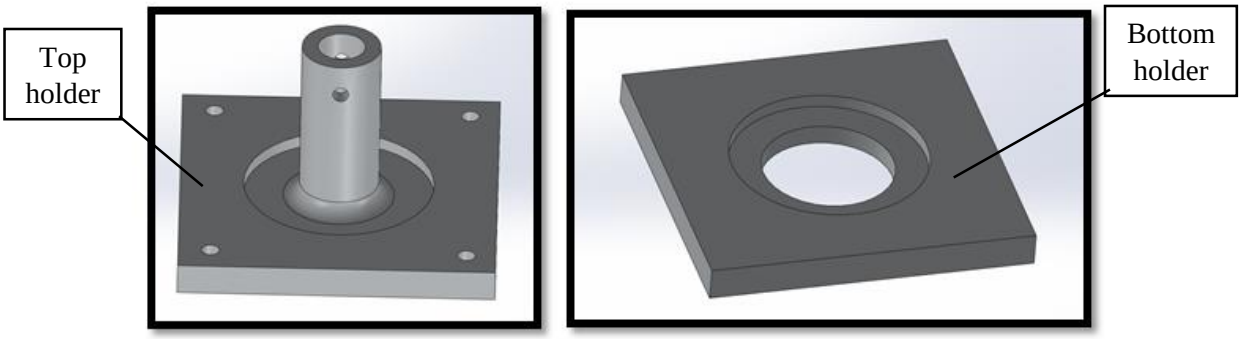


Figure 4.8 Top and bottom holder that rotates with the thrust bearing

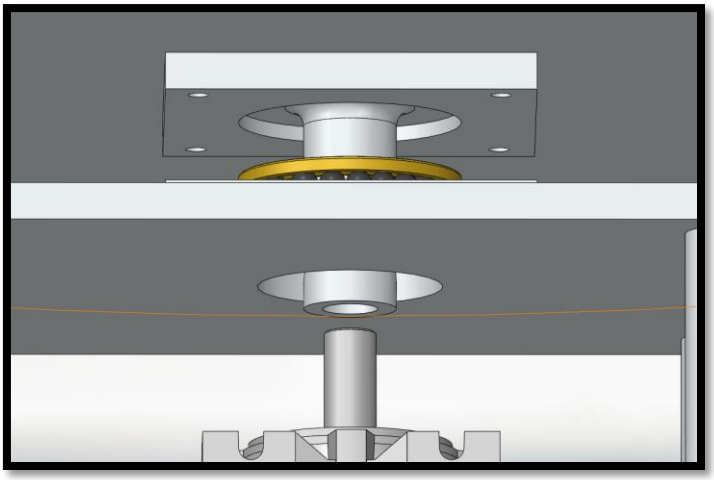


Figure 4.9 Thrust bearing and its holder

The worm gear drive needs to output a certain torque that is required for the top section to rotate. To calculate the torque the moment of inertia of the system must be calculated first.

The moment of inertia of the system is given by:

Equation 4-1 Torque equation

$$\tau = I\alpha$$

Where:

τ is the torque required.

α is the angular acceleration of the system.

I is the moment of inertia of the system

Using Solidworks 2012 the moment of inertia was calculated to be:

$$\bar{I} = \begin{bmatrix} 2.39 & 0 & 0.98 \\ 0 & 2.67 & -0.1 \\ 0.98 & -0.01 & 1 \end{bmatrix}$$

Using the right hand rule on the disk, it can be seen that the angular acceleration is only in the x direction. Therefore, the angular acceleration matrix only have an x component.

$$\alpha = \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}$$

The angular acceleration chosen for the disk is 2rad/s^2, therefore, the torque is equal to:

$$\tau = \begin{bmatrix} 2.39 & 0 & 0.98 \\ 0 & 2.67 & -0.1 \\ 0.98 & -0.01 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

Multiplying these matrixes the torque vector is equal to:

$$\tau = \begin{bmatrix} 4.78 \\ 0 \\ 1.96 \end{bmatrix} N.m$$

Therefore, more than 4.78N.m of torque is needed. Since there are few motors with this amount of torque, a worm geared drive will be used to obtain the required torque. The angular acceleration can be made to be any value. A lower torque would be adequate, however, the system would rotate at a slower rate. Before a worm gear drive can be chosen, the stepper motor which will rotate the worm gear drive must be chosen. The reason for a stepper motor was that the incrementation of the motor is easily controlled, therefore, an encoder is not needed to calculate the position of the stepper motor because each pulse that is sent to the motor rotates it by a certain number of degrees.

It was decided that all stepper motors chosen must have a 12V power supply because the voltage source comes from outside the vacuum chamber. Therefore, it makes sense to only have one voltage source, since more voltage sources means more wires are needed from outside the chamber to inside the chamber.

The stepper motor chosen was a Hybrid stepping motor[12] with the following specifications:

- Step Angle: 1.8 degrees
- Voltage: 12V
- Current: 0.68A
- Dimensions: 57mm X 76mm
- Holding torque: 9kg-cm

The stepper motor has a holding torque of 9kg-cm which translates to 0.88N.m. As the calculations showed, at least 4.78N.m is required to rotate the structure. With the help and advice of Bearing Man Group (BMG) a VARVEL worm gear drive was chosen. VARVEL SPA is an Italian manufacturer of gear boxes and motor gear boxes, revolution motor controllers, designed especially for the textile, milling and pastry industry, glass industry, bottling equipments, wood processing, cigarette manufacturing equipments, conveyer belts, power generators, lifting devices, etc. [58].

The desired torque was provided to VARVEL and they manufactured the worm gear to meet the torque requirement. One problem that was encountered was that worm gears are not normally manufactured to be compatible with stepper motors. An external connector had to be designed and manufactured to make the stepper motor compatible with the worm gear drive. The external connector needed to be able to clamp against the stepper motor and be fitted tightly to the worm gear to prevent any slipping of the shaft. The specially designed connector can be seen in Figure 4.10 (Callout A).

[12] Data sheet can be found on the CD provided

Figure 4.10 Worm gear drive with stepper motor connector

The connector is a flexible module that can clamp closed by tightening a screw. While tightening, the gap between in the connector (Callout B) gets closed and the stepper motor is tightly fastened. As can be seen by Figure 4.10, the connector fits perfectly into the worm gear drive (Callout C). When this connector was rotated the shaft that rotates the top section of the mechanical set-up rotated without any slip

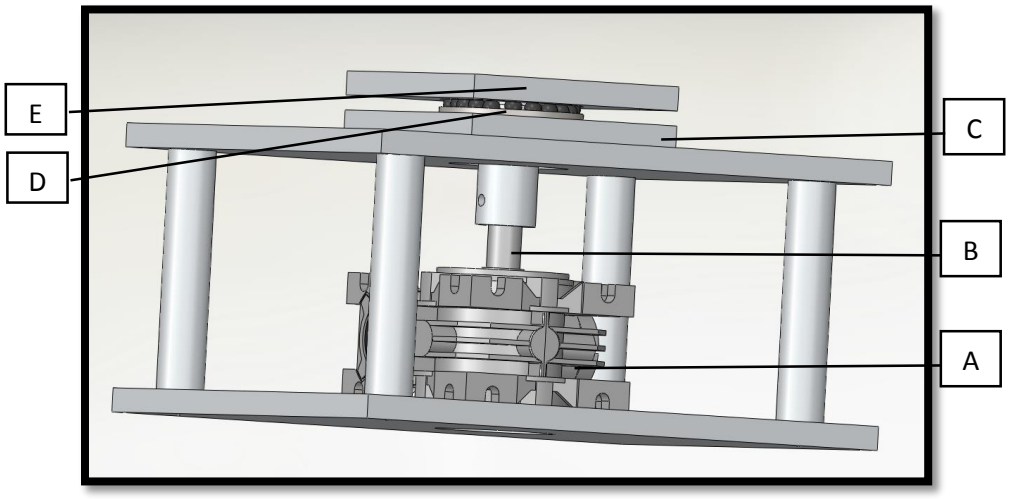


Figure 4.11 Complete bottom section

The different sections of the complete bottom section which are illustrated in Figure 4.11 are:

- Callout A - The worm gear drive
- Callout B - The key shaft
- Callout C - The thrust bearing bottom holder
- Callout D - The thrust bearing
- Callout E - The thrust bearing top holder

4.3 Completed Mechanical Design

After all CAD drawings and calculations were completed, the CAD drawings[13] were sent to be manufactured. After each component was separately manufactured, the mechanical structure was constructed as can be seen by Figure 4.12 and 4.13.

Figure 4.12 Complete mechanical structure

Figure 4.12 illustrates the stepper motor and the worm gear drive connected to one another. The bottom piece will be connected to the rotating arm inside the vacuum chamber. The worm gear rotates a shaft that is connected to the disk. The disk is supported on a set of thrust bearings, which allows the system to rotate with ease.

[13] All cad drawings are attached in Appendix B

Figure 4.13 Worm gear that rotates the mechanical structure

4.3.1 Control of Mechanical Design

This section will describe the methods that were used to rotate the mechanical structure with Labview.

4.3.1.1 H-bridge

The movement of the mechanical design needed to be programmed to rotate so that the requirement of radiating each DUT in a single test can be met. To rotate the mechanical structure an H-bridge was required to control the stepper motors' direction. A L298 Dual H-Bridge Motor Driver[14] was chosen as the H-bridge to control the stepper motor. The H-bridge uses a L298N dual full-bridge driver. It is a high voltage and high current dual bridge driver. It is designed to accept standard TTL logic and drive inductive loads such as relays, solenoids, DC and stepper motors.

Two "enable" inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

[14] The Data sheet for the H-bridge can be found on the CD provided

4.3.1.2 Rotation Program

To control the H-bridge that controls the operation of the stepper motor which translates to the rotation of the mechanical structure, a program needed to be developed. Labview was chosen as the programming language. Labview does not only offer a sufficient GUI interface and Data acquisition capabilities, it can also be used to output the required digital signals needed to operate the stepper motor. To program the stepper motor to rotate in the correct direction, one must first understand the operation of a stepper motor. As with all motors, a stepper motor consists of a stator and a rotor. The rotor carries a set of permanent magnets and the stator has the coils. The very basic design of a stepper motor can be seen in Figure 4.14:

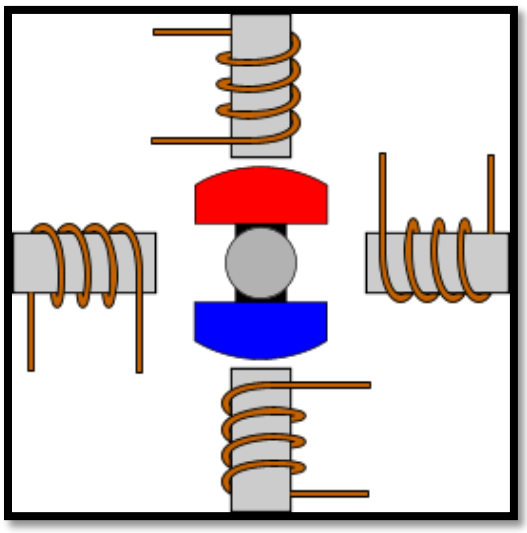


Figure 4.14 Illustration of a stepper motor [59]

There are 4 coils with a 90 degree angle between one another fixed on the stator. The above motor has a 90 degree rotation step. The coils are activated in a repeated order, one by one. The rotation direction of the shaft is determined by the order that the coils are activated.

The stepper motor that was chosen has a step angle of 1.8 degrees, this means when the sequence of the coil get activated the stepper motor rotates 1.8 degrees. Knowing the operation of a stepper motor, the labview code could be written to activate the coils in the correct sequence. The labview program created was able to meet all the requirements set out at the beginning of this chapter.

The GUI interface can be broken up in three section as can be seen in Figure 4.15, section A, B and C.

Section A - In this section the increments to each DUT is typed in. These increments are the number of rotations needed to be completed by the stepper motor to rotate to a specific DUT. These incrementation are calculated by manually moving the DUT in section C until it is in the line of the beam, as can be seen by Figure 4.15. This increment value is then recorded and stored.

Section B – Once the increments have been calculated for section A, section B is used to rotate to any required DUT needing to be irradiated by the proton beam.

Section C – This section of the GUI interface rotates the disc manually in a clockwise or anti-clockwise direction.

The speed of the stepper motor is adjusted below section B.

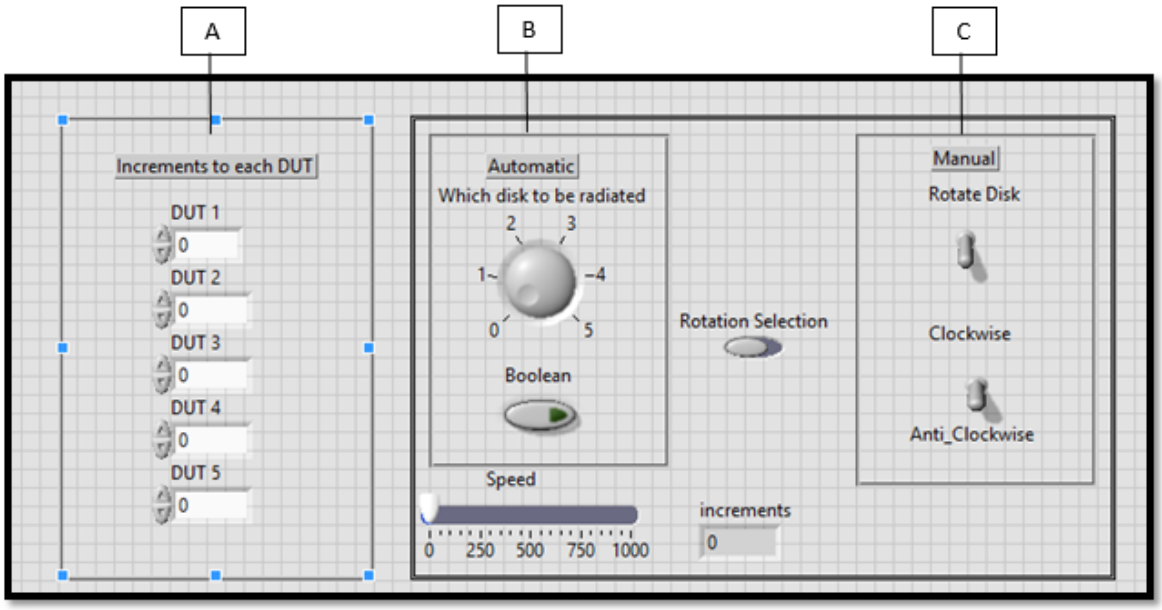


Figure 4.15 LabView GUI interface for stepper motor control

The code for the stepper motor control can be found on the provided CD. The code is broken up into two parts one for the automatic rotation and the other for manual rotation. The program to change the angle of incidence of the DUTs was completed by Stefan van Aardt.

4.4 Device under Test

The Proasic3 AP3P1000 is a low cost, low power FPGA. It is both nonvolatile and reprogrammable. The AP3P1000 FPGA chip has 1 million system gates and 24576 Versatile (D-flip-flops) as explained in Chapter 2. The chip is implemented on the ProASIC3E starter kit board as illustrated in Figure 4.16. This starter kit was an ideal DUT as it can be socketed with any device in the Proasic3E family in the PQ208 package. This an important feature for radiation testing purposes, as a large number of tested FPGA chips can simply be replaced with a new one without the need to replace the entire development board.

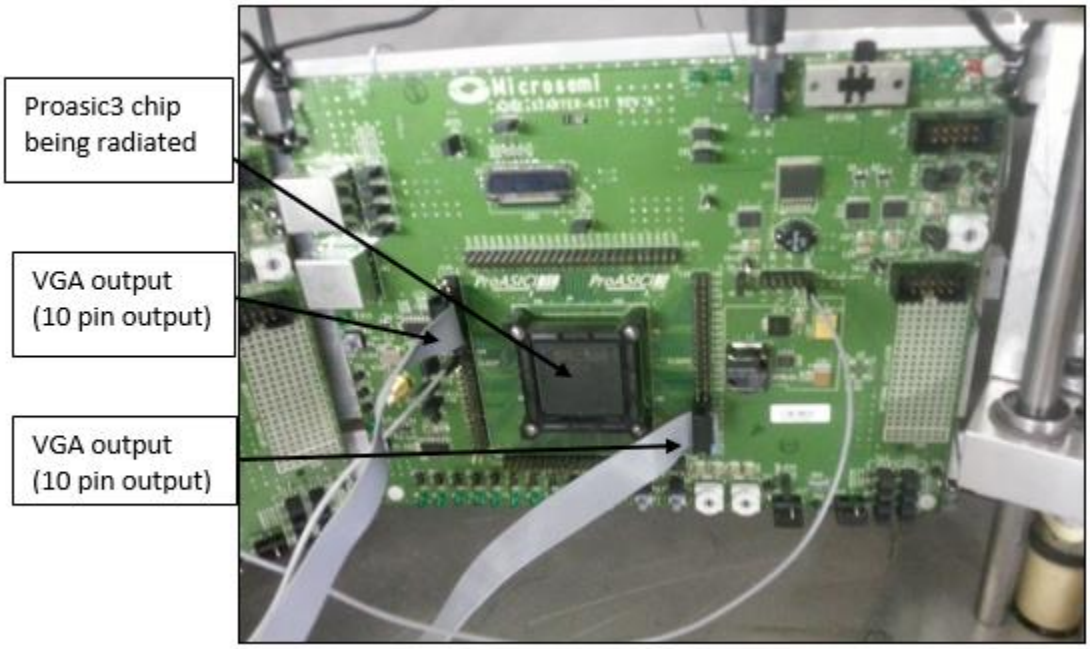


Figure 4.16 DUT that was radiated with 66Mev

Several different versions of VGA code were implemented on the DUT. The different implementations that were implemented on the DUT for radiation testing was:

Day 1:

1. VGA Default – VGA controller with no mitigation
2. VGA DMR – DMR with an SET suppressor to mitigate the SET errors

Day 2:

3. VGA TMR – Normal TMR as discussed in Chapter 2
4. VGA GG – Same configuration as VGA DMR except the guard gate mitigation is inserted instead of the SET suppressor
5. VGA MBU – In this VGA one attempted to force a multiple bit upset. MBU mitigation technique was inserted to attempt to eliminate the MBU.
6. VGA SET Delay – Mitigation technique that uses a delayed signal to attempt to eliminate SEU's

These 6 different VGA implementations will be explained in more detail later in Chapter 5.

4.4.1 VGA output and input signal

Input

A VGA controller needs 5 input signals to operate correctly, namely red, green, blue, VGA clock and reset as illustrated in Figure 4.17. For testing, the RGB signal was kept at a logic level '1'. The clock of the VGA controller has to be at 25 MHz to operate correctly. The proasic3 development board has a 40 MHz onboard oscillator. Therefore, a Phase lock loop is required to achieve a 25 MHz frequency. This PLL was generated on the Altera DE2 control board using the Altera MegaWizard manager. The reason for not using a PLL on the DUT was to only have the VGA controller program on the DUT and to keep the clock source on a different board. The reset was used in the event that an error occurred then the whole operation could be reset so that the old error would not be latched inside the programmer continuously. This, however, should not occur because the VGA controller uses flip flops instead of latches, but precautions were taken anyway.

Output

As discussed in Chapter 3 a VGA controller has five signals: Horizontal sync, Vertical Sync and the three RGB signal. Each output of each VGA implantation was clustered together in groups of ten IO pins, each on different IO Banks where it was possible.

Each implementation was duplicated on the DUT i.e. the RGB, Horizontal sync and vertical sync signal were duplicated such that two pins outputted the exact same signal. This is illustrated in Figure 4.17.

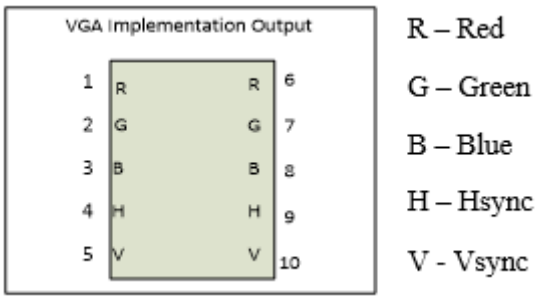


Figure 4.17 VGA implementation Outputs

Pin 1 and pin 6 in Figure 4.17 outputs the exact same VGA red signal and the same for pin 2 and 7 etc. The reason for this was so that two signals could be compared to one another by the control board to observe if any errors occurred with any of the signals, this will indicate if an SEU occurred.

4.5 Control board

The Altera DE2 Development board, shown in Figure 4.18, was used as the control board during testing. The cyclone FPGA chip makes use of 33216 Logic elements and 475 user IO-pins. Of the 475 pins only 72 are connected to the two 40-pin headers. This FPGA development board was used as the connection between the DUT and the data acquisition device. The main purpose for the control board was to count any errors that might occur on the DUT during the experiment.

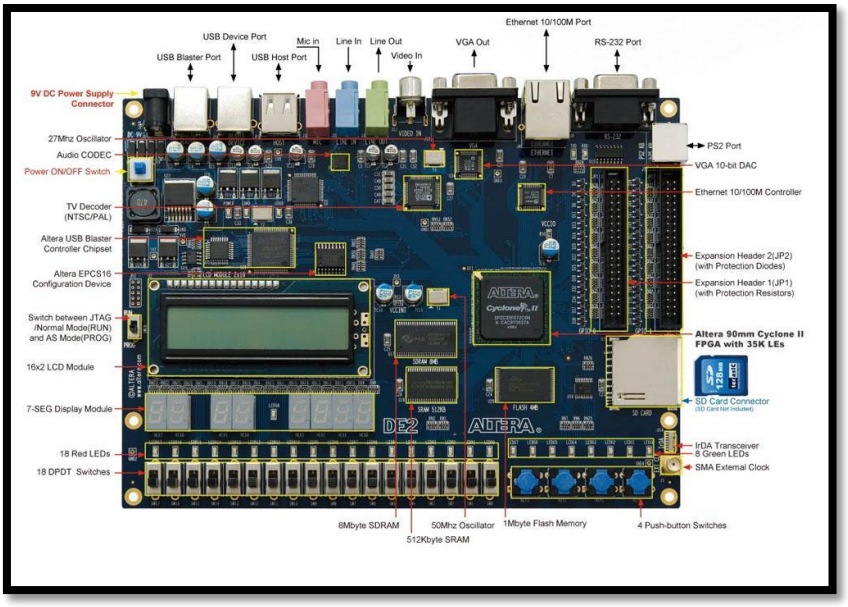
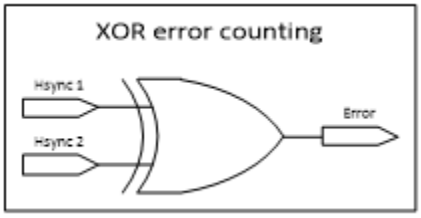


Figure 4.18 Altera DE2 development board *[60]*

The control board was programmed to accept 30 inputs and generate 30 outputs. The 30 inputs are the different VGA signals that are generated by the DUT, the three colours, horizontal sync and vertical sync. These signals will be compared to one another[15] as explained in Section 3.4.2. Once there is any difference between the two signals that are compared the control board will generate a high signal as an output. The output signals are the errors that are captured by the control board. This error will be acknowledged by the DAQ device and displayed on a computer screen. The two identical signals are sent from the DUT to the control board. These identical signals are compared by an XOR gate and the output of the XOR gate is connected to a Latch. The output is latched due to the fact that the data acquisition device runs at a slower frequency compared to the control board. Therefore, if the output is not latched the error might not be observed. A XOR gate logic is explained in Figure 4.19 below, when the two signals are not identical the output will be a '1' or a high and an error will be counted. This error is then sent to the data acquisition device.

[15] A full electrical schematic can be found in Appendix D. Illustrating the complete connections between the electronic devices.



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Figure 4.19 XOR schematic with truth table

4.6 Error Counting

To capture and display the errors that were counted by the Altera control board. The NI 9403 Ethernet data acquisition device was used, illustrated in Figure 4.20. The NI 9403 is a 32-channel, 7 µs bidirectional digital I/O module for any NI CompactDAQ or CompactRIO chassis. The direction of each digital line on the DAQ can be configured for input or output. The channels are compatible with 5V/TTL signals and source up to 2mA output current per channel.

Figure 4.20 NI 9403 and CompactRIO Chassis

This DAQ was ideal as it has Ethernet capabilities. The control room at the NRF iThemba labs was approximately 80m away from where the DUT was being radiated, therefore, Ethernet was an ideal communication method to capture the errors at the vacuum chamber and displaying them at the control room.

Labview was used as an interface to display the SEUs that occurred on the DUT. LabVIEW is a graphical programming platform which is commonly used for data acquisition and instrument control. To program in labview, graphics are used instead of the usual command line coding. The program extension in labview is called a virtual instrument (VI). This VI includes front panels as

a GUI interface for the user and a block diagram panel where the program is written. A GUI interface was created to show when an error occurs and where that error occurred. Figure 4.21 illustrates how the interfaced was set out.

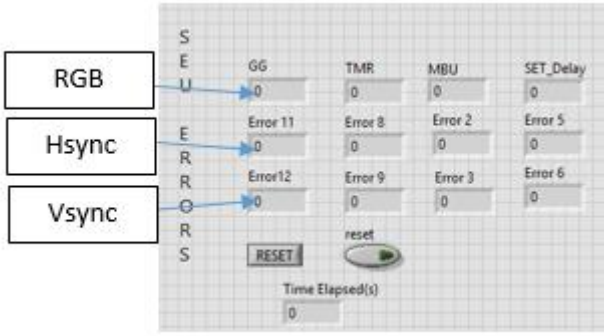


Figure 4.21 GUI interface for counting SEU errors

Each implementation had 3 numeric indicators to display any errors that occurred during testing. The three numerical indicators were broken into the three main VGA signals: RGB, Hsync and Vsync. If an error occurs in the first row, then a SEU occurred in the colour signal. If an error occurred in the second row an error occurred in the Hsync signal and the last row represented errors in the Vsync signal. The RGB signal is the three colours ANDed together. This implies if there is an error in any of the colours it will be seen in that one signal.

There are two resets buttons on the GUI interface, one is used to reset all the numerical indicators on the interface to 0 and the other reset was used to reset the VGA program on the DUT.

4.6.1 Block Diagram LabView program[16]

This section will explain the program that was written to capture the errors from the control board. As the control board was already programmed to count any SEUs that occurred, the DAQ only had to monitor for any high pulses that occurred from the control board's output. This was successfully achieved by the VI illustrated in Figure 4.22 and Figure 4.23.

[16] The Full labview code can be found on the CD

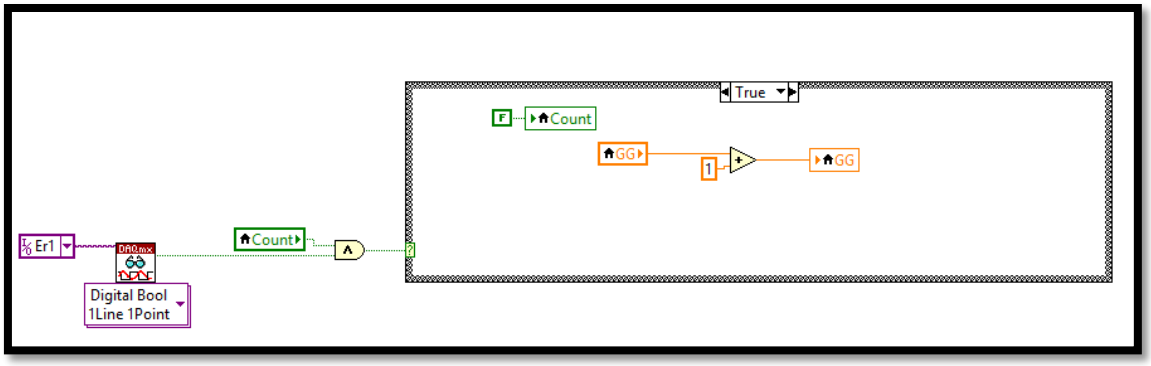


Figure 4.22 VI True Statement

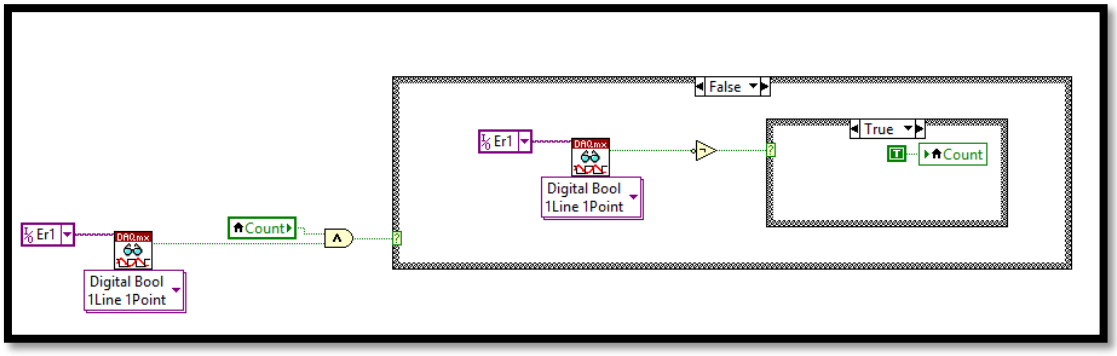


Figure 4.23 VI False Statement

The Digital Bool block diagram is used to capture the signals on the pins of the DAQ card. Once captured it is compared with a Boolean "count". Count is used to assure that a single error is only counted once. This is achieved by forcing it false as soon as an error occurs, as illustrated in Figure 4.22. Count is then only made true once the input pin from the DAQ receives a low signal, this indicates that there are no more errors and the error that was counted has been shifted out the VGA control program. This method of error capturing was duplicated for each implementation.

4.7 Completed Set-Up

Once all the different components of the experiment had been programmed and connected, they all had to be connected to one another to complete the test set-up. The DUT was connected to the mechanical set-up as shown in Figure 4.24 (callout B). The ribbon cable from the DUT is connected to the control board (callout A), it was important to keep the length of the ribbon cable as short as possible to prevent noise in the cable. Once the control board has captured an error from the DUT, the output is then connected to a bridge connector between the control board and DAQ which is outside the vacuum chamber (callout C). All the necessary signals that need to go to the

outside of the chamber are connected to special connecters that are designed so that no air can escape the vacuum (callout E). A closer look at these connectors can be seen in Figure 4.25.

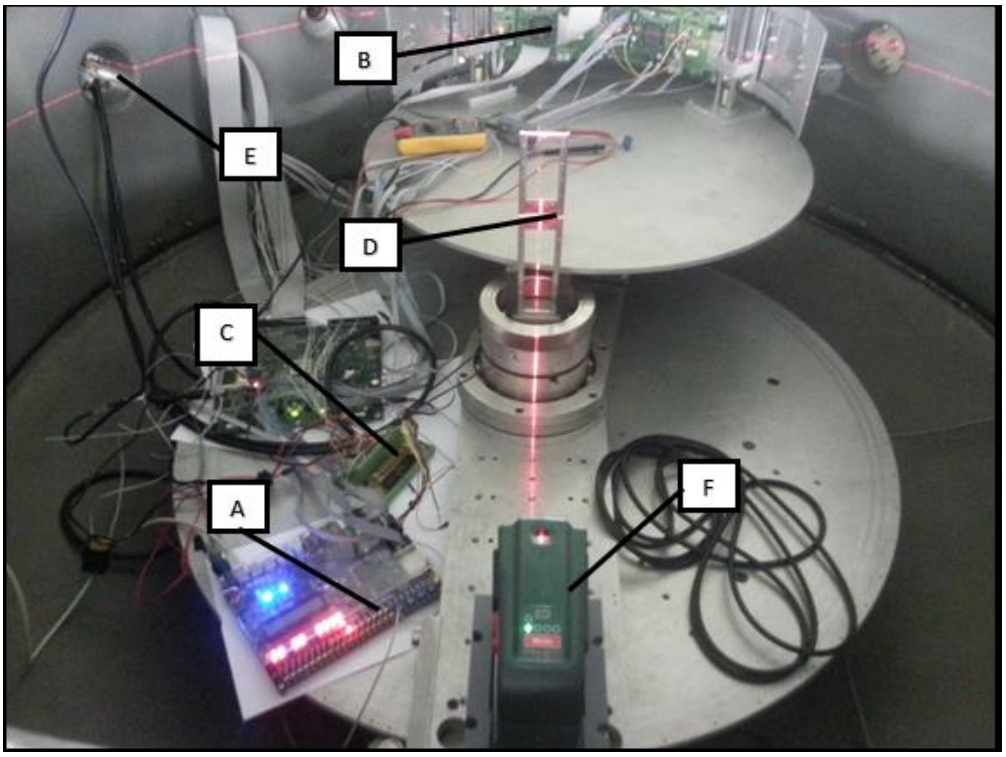


Figure 4.24 Compete Set-Up

The connectors and outer shell were manufactured and constructed by NRF iThemba labs. These were specially designed such that when the vacuum gets created no air escapes or gets into the vacuum chamber. Special wires were purchased to fit the connectors so that the connection from inside the chamber to outside was seamless. Each of these connectors allows for 20 signal pins from the inside to the outside.

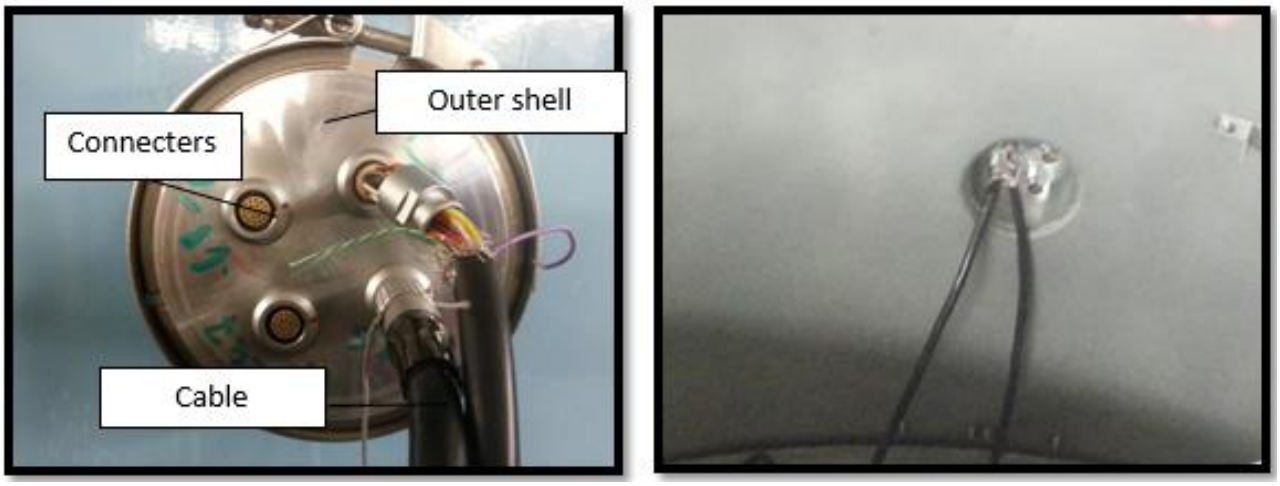


Figure 4.25 Connector outside chamber

The outside wires are directly connected to the NI 9403 Ethernet DAQ as can be seen by Figure 4.26. These wires send the errors that are generated by the control board when an error is captured.

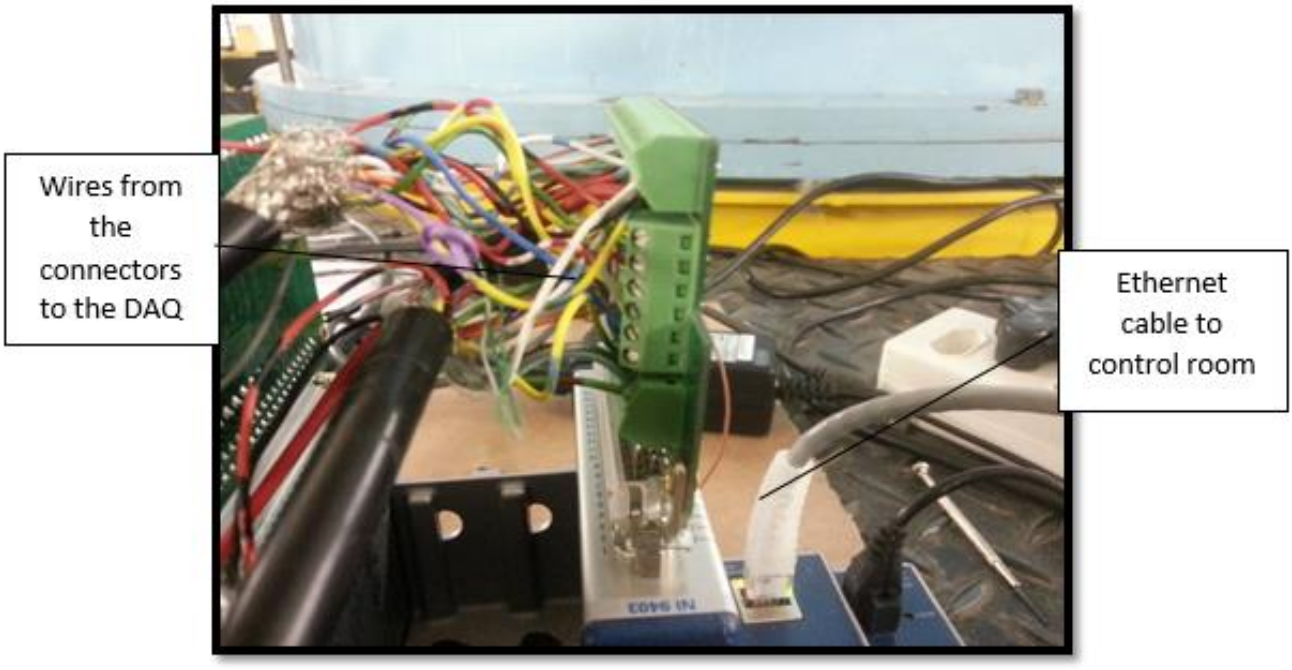


Figure 4.26 NI DAQ connector

4.8 Testing procedure

This section will explain the testing procedure from the DUT being radiated in the vacuum chamber to the errors being captured in the control room. Firstly, the irradiating technique will be explained and then all the connections required to transfer the errors captured from the vacuum chamber to the control room 80m away will be described.

4.8.1 Irradiation Technique

The irradiation procedure consists of the following steps:

1. The structure is rotated in the "open" position were no DUT or mechanical structure is in line of the beam
2. The beam is then switched on to be calibrated using detectors already installed by iThemba labs. With these detectors the control room can calculate the particles' fluency and flux density.
3. The beam is switched off
4. The DUT is moved in front of the beam using the Labview program created.
5. The beam is kept activated for a predetermined time.
6. Measurements are taken while the DUT is radiated, and beam adjustments are made to achieve the best possible measurements.
7. The beam is turned off and the DUT is rotated to change the angle of incidence

After the DUT has been irradiated the SEUs have to be counted and sent to the control room. The process in Figure 4.26 describes the complete testing procedure. While the DUT gets irradiated by the proton beam the control board waits for an SEU to occur. Once an SEU occurs on the DUT the control board captures that error and sends it through the linking connector between the control and monitoring board and the vacuum connectors via ribbon cable seen in Figure 4.27 callout C. The linking connector is then connected to the inside vacuum chamber connectors, these connectors are the only manner of connecting wires inside the chamber to wires on the outside of the chamber. The outside vacuum connector is then connected to the NI9403 DAQ card. This DAQ card is the connection between the vacuum chamber and the control room 80m away. The DAQ is connected via an Ethernet cable to an Ethernet port inside the scattering chamber. This port is connected to a main switch inside the control room were all the different Ethernet lines at iThemba

labs are connected. The laptop with the Labview program is connected to the Ethernet switch inside the control room. This enables Labview to monitor and control everything that is occurring inside the vacuum chamber. This whole process can be seen in Figure 4.27 below.

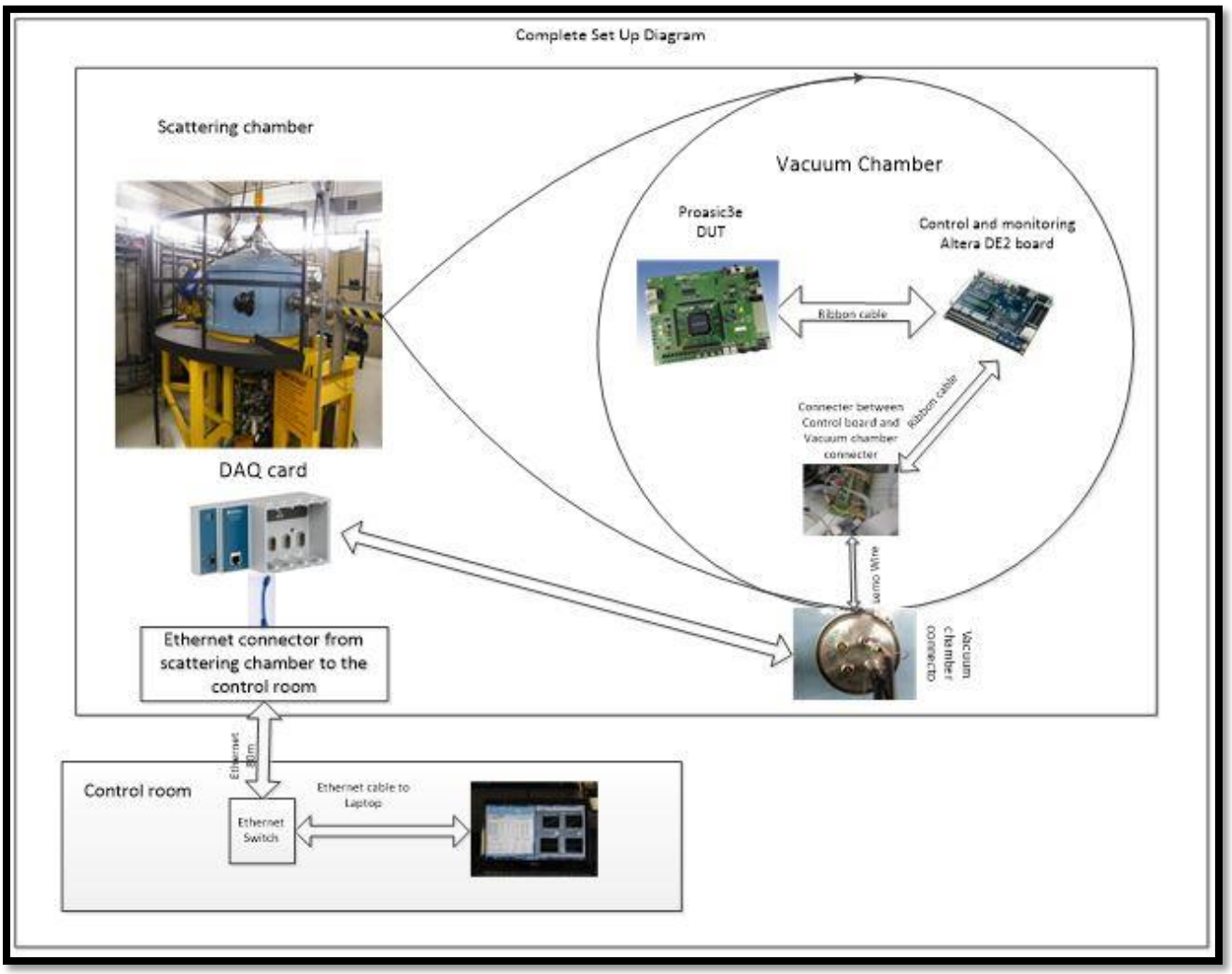


Figure 4.27 Complete Set-up Diagram

Chapter 5

5. Experimental Results and Discussion

This chapter will describe and analyse the results that were obtained by the experiment conducted as described in Chapter 4.

5.1 Beam characteristics

Before any testing can commence at iThemba labs, the beam diameter and characteristics must first be set up in the control room. The control room is the centre point of all testing conducted at the lab. Here they control the energy, flux and diameter of the beam. Magnets are used to focus and defocus the beam. Figure 5.1 shows the computer used to view the size of the proton beam. The light seen on Figure 5.1 is a thin film of Aluminum oxide (AL2O3) that glows as soon as the beam penetrates that material. The material can be seen in Figure 5.1 illustrated by Callout A. This mechanical structure is called the viewer. The viewer is used as a tool by the controller to observe if the beam is heading in the correct direction.

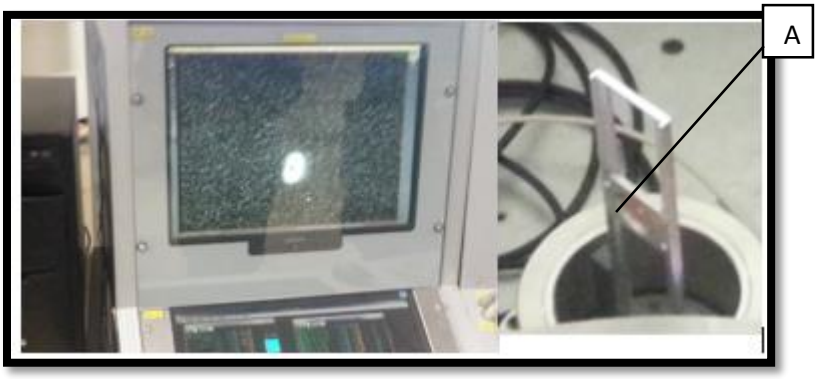


Figure 5.1 Beam illustrated on a computer screen and viewer

A closer image of the screen was taken to estimate the area of the beam as it is an important parameter in the SEE testing calculations. A closer view of Figure 5.2 gave a clear indication that the beam was actually not a perfect circle and was in fact more of an eclipsed shape. The control

room did attempt to give the beam a more circular shape, however, this was the best they could provide. The area of the beam needed to be calculated as it will be used later to calculate the SEU cross section of specific implementations. The centre of the viewer was measured to be 3mm in diameter, lengths A and B, as shown in Figure 5.2, were measured on the computer screen in the control room.

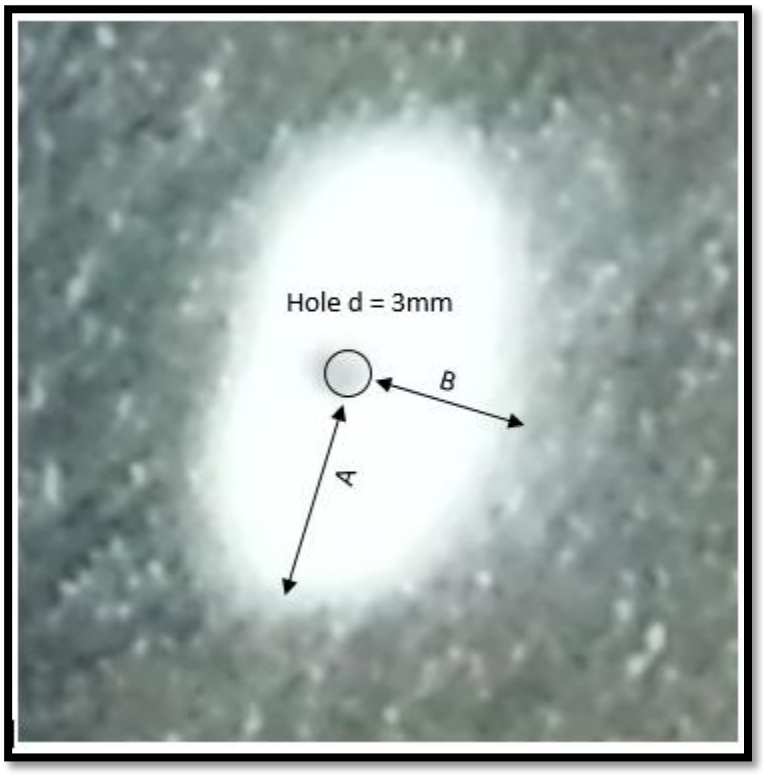


Figure 5.2 The shape of the proton beam

The area for an eclipse can be given as:

$$Area = \pi \times A \times B$$

$$Area = 13.5mm \times \pi \times 9mm$$

$$Area = 3.82cm^2$$

To calculate the amount of protons that radiated the chip during testing, the number of protons per seconds traveling towards the DUT needs to be calculated. This is achieved by the equation below.

$$\mathrm{P} = \frac{\text{Current of beam}}{\text{Charge of a proton}}$$

The number of protons per second for 1nA is given by:

$$P = \frac{1\text{ nA}}{1.602\text{ X }10^{-19}\text{C}} = 6.24\text{X}10^{9}\text{ s}^{-1}$$

The number of protons per second for 2nA is given by:

$$P = \frac{2\text{ nA}}{1.602\text{ X }10^{-19}\text{C}} = 1.25\text{X}10^{10}\text{ s}^{-1}$$

Each test run on Day 1 lasted for 120s, therefore, the amount of protons the beam radiates during a test run is:

$$P_{\text{Beam tot}_{1nA}} = 6.24\text{X}10^{9}\text{ s}^{-1} \times 120s$$

$$P_{\text{Beam tot}_{1nA}} = 7.49\text{X}10^{11}\ Protons$$

$$P_{\text{Beam tot}_{2nA}} = 1.25\text{X}10^{10}\text{ s}^{-1} \times 120s$$

$$P_{\text{Beam tot}_{2nA}} = 1.5\text{X}10^{12}\ Protons$$

To calculate the number of protons that penetrated the FPGA chip, the area of the silicone area of the chip needed to be calculated.

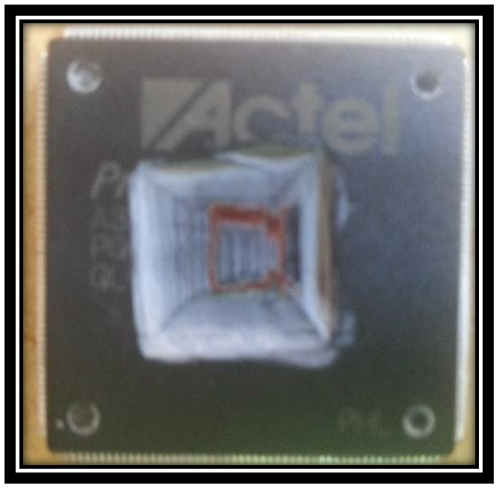


Figure 5.3 - FPGA chip decapsulated

To calculate the area of the silicone inside the chip, a Dremel tool was used to decapsulate the chip. This process grinds down the outer shell until the actual fpga chip is exposed. Once

decapsulated, the silicone area was measured to be 0.36cm^2. With this, the ratio between the beams area and the chips area could be calculated.

$$\text{Ratio} = \frac{\text{Beam Area}}{\text{Dia Area}}$$

$$\text{Ratio} = \frac{3.82\text{cm}^2}{0.36\text{cm}^2}$$

$$\text{Ratio} = 10.61$$

Therefore, the number of protons covering the die area of the silicone of the chip is 1.41 x 10^{11} protons[17] for Day 1.

As explained in Chapter 2, the cross section is an important calculation that indicates how susceptible a system is to SEU's. The higher the value of the cross section of a device, the higher the probability of an SEU occurring and vice versa i.e. the lower the value of the cross section of a device is the more capable the system is to handle SEE's occurring. The equation for the cross section of a device is given below:

$$\sigma_{\text{SEU}} = \frac{\text{N}}{\text{P}^{18}\text{ X FF}}$$

Where:

σ_{SEU} = $Cross\ section$

N=$Number\ of\ SE$Es $counted$

P=$Number\ of\ particles\ passing\ through\ the\ circuit$

FF=$Number\ of\ registers\ in\ the\ circuit$

This equation was used to compare all the various implantations that were tested. It will give an indication as to which implementation method gave the best protection against SEE's.

[17] $\frac{1.5X10^{12}\ total\ protons\ for\ 120s}{10.61} = 1.41X10^{11}$

[18] Also known as the fluence

5.2 Test Results

It was initially agreed upon by iThemba labs that two weeks of testing would be available at their facility. Due to unforeseen circumstances on their behalf, only two days of testing was granted. The amount of implementations tested on each day had to be altered to obtain maximum results.

Since the DAQ card only has 32 IO pins and these pins had to be shared with Stefan Van Aardt, it was decided that for Day 1, the tests involving the stepper motors which control the angle of incidence and the stepper motors that control the lead screw to move the system in a vertical direction will not be used. This allowed for more IO pins to be used for the extra mitigation implementations. Each implementation that was tested over the two days will now be explained in more detail and the results for each implementation will be indicated using tables.

Each table will show the amount of errors[19] that occurred for each VGA signal during testing and at which flux these errors occurred at. With these errors and flux values, the cross section is calculated for each VGA implementation to give a clear indication as to which implementation was the most successful at minimizing the effects of SEE's.

5.2.1 Day 1

Day 1 of testing was conducted on the 9th of October 2014 at iThemba labs. Two implementations were tested on Day 1, namely the default VGA that had no form of mitigation and the VGA DMR with a SET filter designed by Smith [36].

The two programs were programmed together on a single chip. Table 5.1 indicates the percentage each program occupied on the chip and the number of flip flops in each design. The amount of flip flops are important to calculate the cross sectional area per bit. As explained in chapter 2, SEUs occur on memory elements and in this instance it is the flip flop, therefore, the more flip flops an implementation has the higher the chance of an SEU occurring.

[19] The Raw labview Data can be found in Appendix C

Table 5.1 Indicating chip percentage and the number of FF of Implementations Day 1

	Chip percentage	**Number of Flip Flops**
VGA Default	48%	2240
VGA DMR	50%	2900

Day 1's test was split into two parts. The first part of the test was conducted with the viewer out of the path of the beam, so that the scattering of the proton beam would be minimal. The second part of the test the viewer was put in the path of the beam in an attempt to cause scattering of the beam and hence cause secondary particles. As research has shown (Chapter 2), protons themselves do not have a LET high enough to cause SEUs on a regular basis, however, the secondary particles they produce might have a high enough LET to produce the required upsets.

5.2.1.1 VGA Default – No mitigation

The VGA Default implementation is the benchmark implementation. This implementation has no form of SEU mitigation implemented into it. Therefore, this implementation is suspected to have the most errors compared to the mitigated VGA implementations. Due to the beam only being available for 30 min for testing, only three different currents were used on Day 1 as indicated in Table 5.2.

Table 5.2 indicates the current used and if it was with or without the viewer in the path of the beam. The errors for each VGA signal are also indicated in Table 5.2 and Table 5.3

Table 5.2 Errors of VGA Default with no viewer

Signals	**1 nA no viewer**	**2 nA no viewer**	**4 nA no viewer**
Colour	0	0	1[20]
Horizontal Sync	0	0	0
Vertical Sync	0	0	0
Total Errors	0	0	1
Time	120s	120s	120

[20] Error occurred at approximately at about 90 seconds

Table 5.3 Errors of VGA Default with the viewer

Signals	1 nA with viewer	2 nA with viewer
Colour	48	114
Horizontal Sync	15	39
Vertical Sync	0	5
Total Errors	63	158
Time	120	120

It was noticed from Table 5.2 that with no viewer in the path of the beam, there were no significant errors. Only when the viewer was brought into the path of the beam to cause secondary particles, errors started occurring as indicated in Table 5.2.

From these Tables, the SEU cross section for the VGA Default was calculated to be:

1nA:

$$\sigma_{SEU} = \frac{N}{P \text{ X } FF}$$

$$\sigma_{SEU} = \frac{63}{7.06 \times 10^{10} \times 48\% \text{ X } FF}$$

$$\sigma_{SEU} = \frac{63}{3.39 \text{ X } 10^{10} \text{ X } 2240}$$

$$\sigma_{SEU} = 83 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

2nA:

$$\sigma_{SEU} = \frac{158}{1.41 \times 10^{11} \times 48\% \text{ X } FF}$$

$$\sigma_{SEU} = \frac{158}{6.77 \text{ X } 10^{10} \text{ X } 2240}$$

$$\sigma_{SEU} = 104.19 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

5.2.1.2 VGA DMR

The second implementation tested on Day 1 was the VGA controller with DMR with a SET suppressor as illustrated in Figure 5.4. Since this implementation was running on the same FPGA chip as the VGA default, the same parameters and testing conditions apply to the VGA DMR implementation as was described for the VGA default.

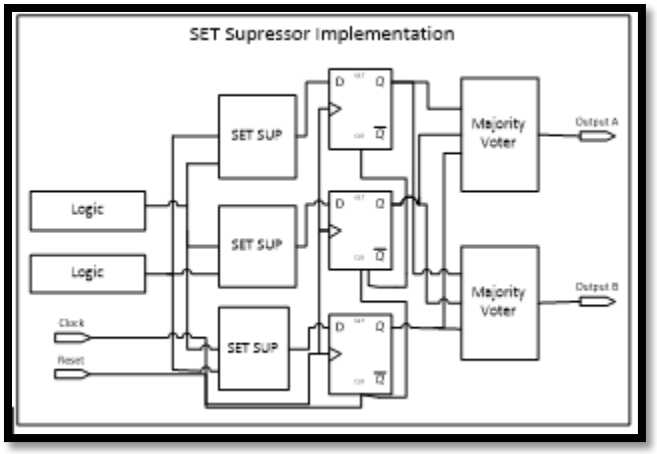


Figure 5.4 Implementation of VGA DMR

The VGA DMR implementation makes use of Dual Modular Redundancy. The combinational logic is doubled and connected to the SET Suppressor seen in Figure 5.4. The flip flops were tripled while the majority voters were added to increase the protection against SEU's.

Table 5.4 indicates the current used and if the testing that was conducted was conducted with or without the viewer in the path of the beam. The errors for each VGA signal are also indicated in Table 5.4.

Table 5.4 VGA DMR errors with no viewer in place

Signals	**1 nA no viewer**	**2 nA no viewer**	**4 nA no viewer**
Colour	0	0	0
Horizontal Sync	0	0	0
Vertical Sync	0	0	0
Total	0	0	0
Time(s)	120s	120s	120

Table 5.5 VGA DMR errors with viewer in place

Signals	**1 nA with viewer**	**2 nA with viewer**
Colour	0	10
Horizontal Sync	0	0
Vertical Sync	0	0
Total Errors	0	10
Time(s)	120	120

Looking at the results in Table 5.4, it can be seen that similar events occurred with the DMR as with the VGA default implementation. These similarities are that no errors were observed when

the viewer was not in the path of the proton beam. When the flux was increased to 3nA the FPGA chip became unresponsive and no new programs could be uploaded.

The cross section for 2nA for VGA DMR was calculated to be:

$$\sigma_{SEU} = \frac{N}{P \text{ X } FF}$$

$$\sigma_{SEU} = \frac{10}{6.627 \text{ X } 10^{10} \text{ X } 2900}$$

$$\sigma_{SEU} = 5.2 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

5.2.2 Day 1 Discussion of Results

This section will discuss the results that were obtained during testing on Day 1. Initially with the viewer in the path of the proton beam, no errors were observed for any of the implementations. This corresponds to tests that were conducted earlier this year at iThemba LABS [42]. It was then decided that for all further tests conducted after Day 1 at iThemba labs, the viewer will be placed in the path of the proton beam to create the secondary particles that are required to cause upsets. In space, FPGA's are enclosed inside a satellite of some sort [61] which causes secondary particles. Therefore, inserting the viewer will allow the testing to yield more practical results.

Comparing the two SEU cross-sections of the VGA default and VGA DMR, it can be noted that the VGA DMR with the AND-OR Multiplexer mitigation technique inserted significantly improved the system's ability to combat SEU's by a factor of 20 at 2nA.

At 1nA the SEU cross section for VGA default was 83 X 10^{-14}cm2/bit, this is just a bit less than the SEU cross section calculated at 2nA. This is an expected result because as the current increases, the number of protons that penetrated the DUT increases, therefore, increasing the probability of an error to occur. There were no errors observed for the VGA DMR at 1 Na

As mentioned in Chapter 2, an upset can occur in many different parts of the VGA system namely: the combination logic, IO banks, memory elements[21], clock, reset lines and the majority gate. As can be noted by Figure 5.4, the DMR implementation's clock and reset pins were not protected so it is assumed that the errors that occurred in the VGA DMR implementations were from these signals and from the IO pins that were not protected.

It is also important to note that the colour signal errors were always more than the Horizontal synchronise signal. This is due to the fact that the colour signal consists of the three RGB signals, whereas the horizontal signal consist only of one signal. Therefore, it makes sense that the colour signal would have more errors than the Vsync and Hsync signals. The vertical synchronise signal has the least errors in the VGA system, since its frequency of occurrence is the least. It takes 800 horizontal synchronization signal pulses per one vertical synchronization pulses and an error can only occur when it gets latched by the flip flop, so one would expect the vertical signal to have the least errors of the three VGA signals.

The beam was increased to 3nA for 120 seconds, however, no errors were observed. When the DUT was examined after testing it was noticed that the DUT had stopped working. An attempt to upload a new program to the DUT proved unsuccessful[22]. After a new FPGA chip was inserted into the Proasic3E development board, operation of the board was back to normal and ready for Day 2's testing.

5.2.3 Day 2 Test

Test run 2 was conducted on the 14th of October 2014. It was decided that due to the unreliable beam time available at NRF iThemba labs that as much code as possible must be tested on one chip. It was decided to test, VGA GG, VGA TMR, VGA MBU and VGA SET Delay. These different implementations were uploaded to the chip so that they each cover 25% of the memory. This allowed all implementations to have the same chance for a SEU to occur. Experiences from Test 1 indicated that without the viewer in place to cause secondary particles, the chance of a SEU

[21] Flip Flops
[22] Flash pro indicated an error -24

to occur is minimal. Therefore, for the test conducted on Day 2 the viewer was in the path of the proton for the entire duration of the test.

The test was split up into four test runs. The first run the 66 MeV proton beam was set to 1nA, the 2nd test set to 2nA and so on. Previous testing also indicated that the FPGA chip starts to deteriorate when the flux gets increased to about 3nA to 4 nA. Keeping that in mind, the amount of time that each test run was conducted for was left up to discretion, if it was felt that enough errors where captured to get significant results the test run was stopped and the current was increased. This method was used to keep the FPGA chip in a working condition for as long as possible.

The beam characteristics were kept the same as they were for Day 1's testing. Therefore, the area and the energy of the beam was approximately identical to testing conducted on Day 1. Table 5.6 indicates the percentage that each implementation is allocated on the FPGA and the number of flip flops each implementation has.

Table 5.6 - Chip percentage and Number of FF Day 2

	Chip percentage	**Total Number of Flip Flops**
VGA TMR	25%	3500
VGA GG	25%	2391
VGA Delay	25%	1237
VGA MBU	25%	3351

5.2.3.1 VGA TMR

One of the most common methods to mitigate SEU in FPGA's is called Triple Modular Redundancy as explained in Chapter 2. In this sub section the VGA TMR implementation will be explained and results that were obtained will be discussed and analysed.

Figure 5.5 below illustrates the structure of the TMR that was implemented in the VGA net list.

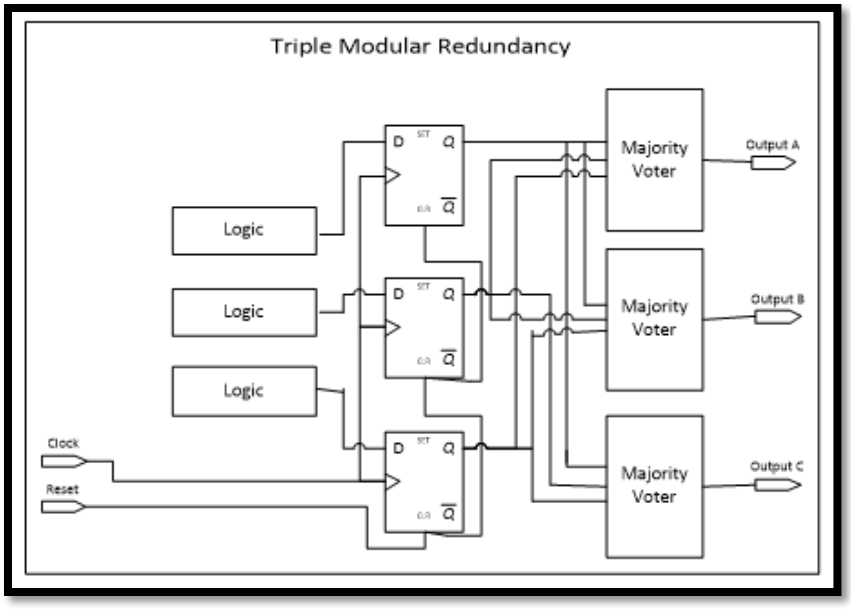


Figure 5.5 VGA TMR Implementation

The combinational logic of the VGA and the flip flops that correspond with that logic was tripled. The output of each flip flop was then connected to the input of the respective majority voter as illustrated above. It is important to note that due to the fact that the clock and reset signals are not tripled, this is not full global TMR. Therefore, single event upsets are expected.

Table 5.7 below indicates the errors occurring and at which flux they occurred at.

Table 5.7 - VGA TMR results

Signals	**1 nA**	**2 nA**	**3 nA**	**4 nA**
Colour	10	25	35	6
Horizontal Sync	3	23	18	0
Vertical Sync	0	0	0	0
Total	13	48	53	0
Time(s)	90	104	120	62

The cross sectional for VGA TMR at 1nA:

$$\sigma_{SEU} = \frac{13 \text{ errors}}{1.32 \text{ X } 10^{10} \text{ X } 3500}$$

$$\sigma_{SEU} = 28.13 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

The cross sectional for VGA TMR at 2nA:

$$\sigma_{SEU} = \frac{48 \text{ errors}}{3.06 \text{ X } 10^{10} \text{ X } 3500}$$

$$\sigma_{SEU} = 44.82 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

The TMR mitigation technique that was inserted into the VGA TMR implementation is well-known and is a common mitigation scheme used for correcting SEU errors in FPGA's. As mentioned in Chapter 2, the main disadvantage with the TMR is the extra area overhead it creates on the FPGA. Tripling of the circuit causes the design to have three times more area and power consumption than the original design, which is the VGA Default.

With the circuit tripled it is expected for the SEU cross section to decrease, which it did, since the VGA Default was $83 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$ at 1nA and $104.19 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$ at 2nA. The TMR decreased these cross sections to $28.13 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$ at 1nA and $44.82 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$ at 2na. With full TMR one does not expect any errors to occur except on the IO banks. However, a closer look at the VGA TMR implementation in Figure 5.5 shows that the reset and clock signals were not tripled, so any upsets occurring on these two signal lines will not be protected. This is the same for all the implementations tested on Day 2 because none of the reset and clock signals were protected. The IO banks were also not protected from any upsets. The reason for this was that, initially, more than two days of beam time was expected and full global TMR was one of the implementations that was planned to be tested if more beam time was granted.

The TMR implementation improved the SEU cross section of the non-mitigated VGA default by a factor of 2.95 at 1nA and 2.32 at 2nA. It was interesting to note that even though the colour signal consisted of three signals, it had about the same amount of errors as the horizontal signal for the test at 2nA. One would expect to have around three times more errors, which was the case for the 1nA, however, this was not the case. This indicates the randomness of the proton particles that cause the upsets. The viewer in place causes scattering of the particle[23]. The proton beam will hit different parts of the DUT with each different test runs, as explained in Chapter 4. Even though

[23] The Simulation of the scattering is illustrated in Appendix A

there was scattering of the proton beam, it was stated by iThemba labs that the fluence and the uniformity of the beam can be given at a ∓10% accuracy. Therefore, most of the FPGA chip should receive the same amount of protons. However, that 10% tolerance can explain the increase and decrease of errors for some signals.

Similarly to the VGA DMR results discussed earlier in the chapter, the vertical signal gave no errors. This could be due to the frequency at which the data of the vertical sync gets captured.

Errors were observed in the signals at 3nA, however, it was at 3nA where the chip started deteriorating and the chip ceased to work. Therefore, one cannot be certain at what exact point the chip ceased to work and the results for 3nA was discarded and therefor, not analysed with the other results.

The beam was increased to 4nA to confirm the assumption of the chip not being operational anymore. The assumption was found to be correct when the errors were very fewer or non-existent at 4nA than at the other nano amp values.

It can be concluded that the TMR mitigation technique did improve the design's ability to handle SEU occurrences.

5.2.3.2 VGA Guard Gate

The 2nd implementation that was tested on the 2nd day at iThemba labs was the VGA controller with the Guard Gate mitigation scheme. The guard gate mitigation scheme was explained in more detail in Chapter 2.

In this sub-section the VGA Guard gate implementation will be explained and results that were obtained will be discussed and analysed. Figure 5.6 illustrates the schematics of the guard gate mitigation used in the GG implementation. As with the SET Suppressor tested on Day 1, the guard gate was implemented in the same way with the combinational logic doubled and flip flops tripled.

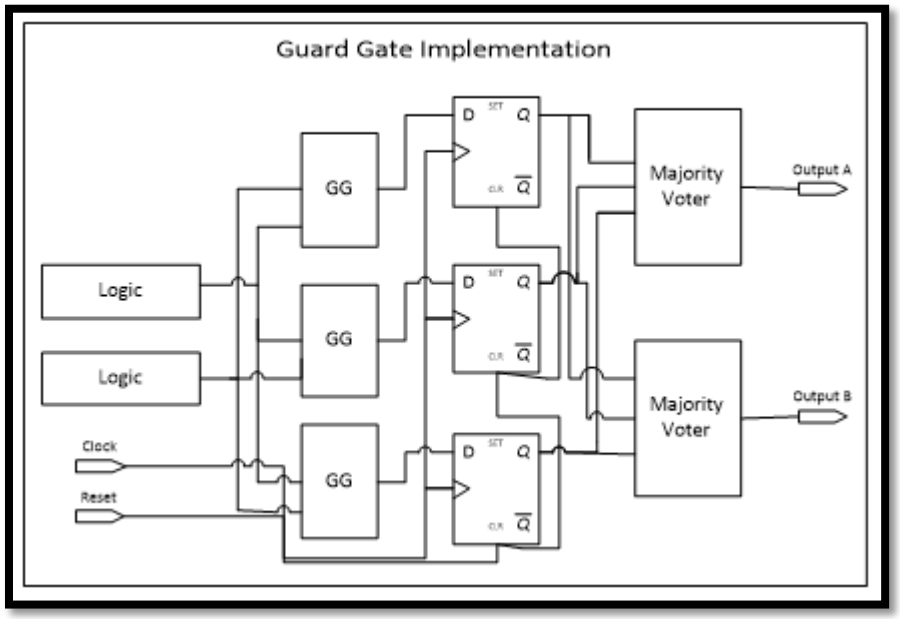


Figure 5.6 Guard gate Implementation

Table 5.8 below indicates the errors occurring and at which flux they occurred at.

Table 5.8 VGA GG Results

Signals	**1 nA**	**2 nA**	**3 nA**	**4 nA**
Colour	0	28	31	0
Horizontal Sync	0	0	10	0
Vertical Sync	0	0	0	0
Total	0	28	41	0
Time(s)	90	104	120	62

The cross sectional for VGA GG at 2nA:

$$\sigma_{SEU} = \frac{N}{P \text{ X } FF}$$

$$\sigma_{SEU} = \frac{28}{3.06 \text{ X } 10^{10} \text{ X } 2391}$$

$$\sigma_{SEU} = \ 38.27X10^{-14} cm^2/bit$$

As mentioned in Chapter 2, the Guard Gate is a circuit designed by Muller. The guard gate was designed to decrease the SEU effect in a digital design while keeping the area overhead and power consumption low.

There were no errors that occurred during 1nA testing. Errors only started occurring at the 2nA mark. When comparing the VGA GG to the original VGA Default it can be noticed that the Guard Gate implementation improved the system SEU cross section by a factor of 2.72. The cross section went down from $104.19 \text{ X } 10^{-14}\text{cm}^2/\text{bit}$ to $38.27\text{X}10^{-14}\text{cm}^2/\text{bit}$ at 2nA. This is a small improvement from the TMR implementation. There were only errors observed on the three RGB signals and none on the horizontal and vertical synchronisation signals. This is just another indication of the randomness that the $\pm$10% fluence tolerance causes.

There were errors observed at 3nA, however, as mentioned previously it is unclear when exactly at 3nA the FPGA DUT stopped working correctly, so the errors cannot be analysed with confidence.

It can be concluded that the Guard gate mitigation scheme is a viable method to reduce the SEU errors that occur in a system because it decreases the SEU cross section while not increasing the area overhead as much as the TMR implementation does.

5.2.3.3 SET Suppressor Delay

The next implementation of the 2nd day was an adaption of the SET Suppressor implemented on Day 1. The SET suppressor Delay uses the same AND-OR Multiplexer mitigation scheme as the SET suppressor used in VGA DMR. The SET suppressor makes use of a delay element inside the SET suppressor Delay logic. The SET Suppressor with delay can be seen in Figure 5.7.

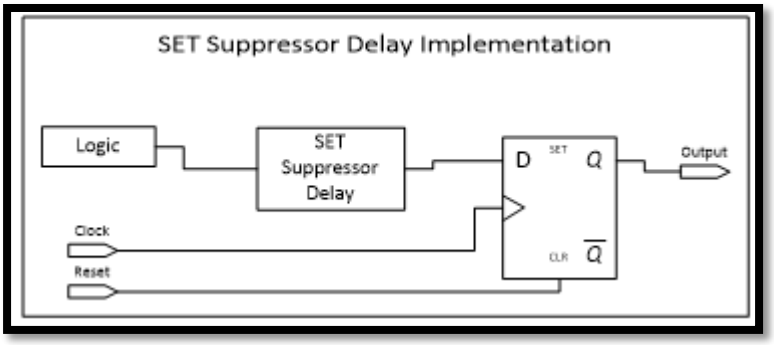


Figure 5.7 SET Suppressor Delay Implementation

The SET Suppressor delay element was inserted in front of every D flip flop in an attempt to mitigate any SET errors that might occur. In this implementation all the logic and flip flops were kept the same as the VGA Default implementation. Therefore, there is no modular redundancy protection. Initial observation of this system indicates that this implementation should have the most errors occurring since there is minimal protection.

Table 5.8 below indicates the errors occurring and at which flux they occurred at.

Table 5.9 VAG SET Delay results

Signals	**1 nA**	**2 nA**	**3 nA**	**4 nA**
Colour	14	32	0	0
Horizontal Sync	0	4	0	0
Vertical Sync	0	0	0	0
Total	14	36	0	0
Time(s)	90	104	120	62

The cross section for VGA SET Suppressor Delay at 1nA was calculated to be:

$$\sigma_{SEU} = \frac{14}{1.32 \text{ X } 10^{10} \text{ X } 1237}$$

$$\sigma_{SEU} = 85.74 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

The cross section for 2nA was calculated to be:

$$\sigma_{SEU} = \frac{36}{3.06 \text{ X } 10^{10} \text{ X } 1237}$$

$$\sigma_{SEU} = 95.11 \text{ X } 10^{-14} \text{cm}^2/\text{bit}$$

The SET Suppressor Delay mitigation method was the only implementation where no modular redundancies were implemented. This mitigation scheme used the delay of a signal in an attempt to mitigate the SEU errors, as explained in Chapter 2.This mitigation method was not an effective method because the SEU cross section only improved by a factor of 1.1 at 2nA and gave a worse SEU cross section at 1nA which was surprising.

The reason for the lack of improvement of the SEU cross section is that no memory elements were protected because there were no majority voters used and the only protection that was used was the delay element in front of the flip flop as can be seen by Figure 5.7. It can also be noted that a 2-inverter delay was used as explained in Chapter 2. Since a SET has a specific pulse width, if the delay caused by the string of inverters is no longer than the SET pulse width then the error will be captured. In an attempt improve the system's SEU tolerance the string of inverters could be made longer by 2xN to ensure that the delay is longer than the SET pulse width. To conclude, this method is an effect method if implemented with more protection as was tested by [45], however, the implementation tested in the way as illustrated in Figure 5.7 is not an effective method to combat SEU's.

5.2.3.4 Multiple Bit upset

In this implementation an alternative method to the majority voter was tested. The multiple upset filter is an adaption of the SET filter discussed in Chapter 2 .This filter uses the same configuration as the SET Filter; however, this is a three-input filter instead of two as indicate in figure 5.8 below.

The purpose of this mitigation scheme is to attempt to eliminate multiple bit upsets (MBU). However, the energy level at which testing was conducted at is not sufficient to cause multiple bit

upsets. Therefore, to test this mitigation scheme a MBU had to be forced. A MBU was successfully forced when the output of one flip flop was connected to the set/reset of the next flip flop (Indicated by the red connections in Figure 5.8).

The reason that the output of the flip flop was not connected to the input of the next flip flop, was because one could not have been confident that the flip flops would output simultaneously. Figure 5.8 illustrates the configuration of the circuit and how the MBU was forced.

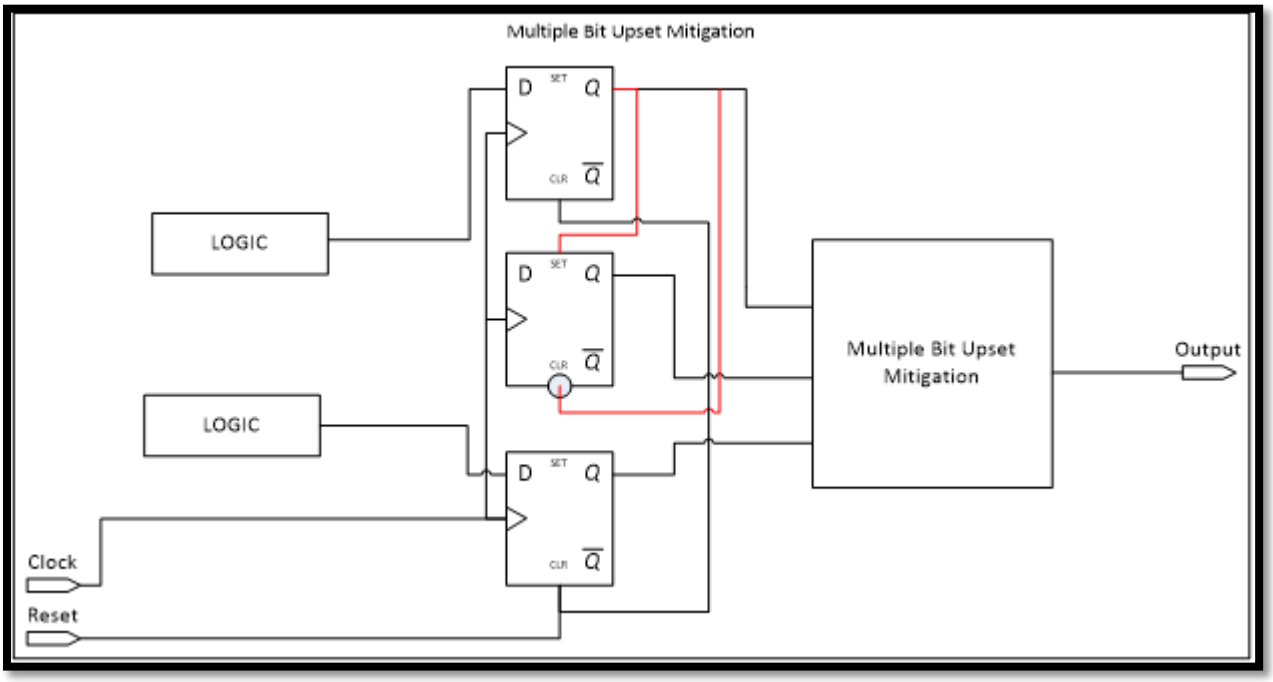


Figure 5.8 - VGA MBU implementation

The VGA MBU implementation was tested on Day 2. It covered approximately 25% of the FPGA chip. There has not been much MBU mitigation testing conducted in the past. Therefore, it was unclear what to expect during testing. There were four different tests conducted at various current values. The tables below indicate necessary information required to calculate the effectiveness of the MBU Filter.

Table 5.10 VGA MBU implementation results

Signals	**1 nA**	**2 nA**	**3 nA**	**4 nA**
Colour	2	8	0	0
Horizontal Sync	0	3	0	0
Vertical Sync	0	0	0	0
Total	2	11	0	0
Time(s)	90	104	120	62

The cross section for 2nA was calculated to be:

$$\sigma_{SEU} = \frac{N}{P \text{ X } FF}$$

$$\sigma_{SEU} = \frac{11}{3.06 \text{ X } 10^{10} \text{ X } 3351}$$

$$\sigma_{SEU} = \ 10.73 \text{ X} 10^{-14} \text{cm}^2/\text{bit}$$

With this cross section a comparison can be made with the other mitigation schemes.

As mentioned earlier there are not many MBU mitigation schemes that have been designed and tested. The different techniques that have been designed to date are all complicated systems i.e. Salamon code and hamming code [62]. Other techniques involve making sure memory elements are not adjacent to one another in the memory cells in the FPGA [63]. This has been proved effective; however, one immediately loses half of the FPGAs memory.

The most important aspect of the VGA MBU testing was to ensure that a MBU is guaranteed to occur and that was accomplished by connecting the output of one of the flip flops to the set and reset of another flip flop. Therefore, if an upset is captured by the first flip flop it will propagate the second and two errors will be sent to the MBU filter.

Since this method was designed as an alternative to the majority voter, comparison needs to be made between the VGA TMR and VGA MBU because these two implementations have no protection in front of the flip flops like DMR and GG. The TMR improved the original design ability to protect itself from SEU occurrences by a factor of 2.32 whereas the MBU mitigation filter improved the original design by a factor of 9.71. This is a significant improvement. As can be seen by Figure 5.8 for the MBU implementation the combinational logic was not tripled like the VGA TMR implementation, it was doubled. So it is clear that the MBU is an appropriate

alternative to the majority gate, since it has less overhead and protects not only against SBUs but MBUs as well. More detailed testing was conducted by Stefan van Aardt. His test involved implementing the MBU mitigation with string of inverters and used latches instead of flip flops. He also implemented the MBU filter in different configurations to indicate the optimum configuration for the MBU mitigation filter.

As expected, at 3nA and above the FPGA chip started malfunctioning as was the case for all the implementations tested on Day 2.

5.2.4 Combination of Day 1 and Day 2

This section will discuss the results of Day 1 and Day 2 combined. Only two days were available for testing at NRF iThemba Labs. In these two days six different VGA implementations were tested, these were the original VGA design without mitigation and five implementations protecting the system from SEU errors. Some of the mitigation methods were well known methods (i.e. TMR) whereas others are new designs by different researchers which haven't been thoroughly tested.

After all the tests were conducted it was noticed that errors were observed on all implementations while still being confident that the chip did not malfunction at 2nA. For that reason only the errors that occurred at 2nA will be discussed.

Table 5.10 gives the number of errors of all the implementations tested over the two days at 2nA, the SEU cross section per circuit and per bit and the combination logic elements used for each implementation. Table 5.11 indicates the factor of improvement each implementation had and the increase of combinational logic compared to the original VGA Default design.

Table 5.11 Cross section of each implementation at 2nA

	Proton fluence × 10^9	**Number of SEU's counted**	**Combination logic elements**[24]	**SEU cross-section per circuit × 10^{-12}**	**SEU cross-section (cm^2/bit) × 10^{-14}**
VGA Default	67.7	158	139	2333.86	104.19
VGA DMR	66.2	10	511	150.8	5.2
VGA GG	30.6	28	712	915.04	38.27
VGA SET Delay	30.6	36	284	1176.51	95.11
VGA TMR	30.6	48	429	1568.7	44.82
VGA MBU	30.6	11	312	455.74	10.73

From Table 5.10 above it can be seen that the VGA Default, which had no mitigation techniques implemented, had the worst SEU cross section, which was as expected. The best mitigation technique that was tested during the two days was the VGA DMR which had the SET Suppressor implemented in front of the flip flops. As the SET Suppressor is an alternative SET filter for the Guard Gate, a comparison had to be made. The GG implementation did improve the system's SEU tolerance, however, not nearly as much as the SET Suppressor used in VGA DMR and it can be noticed from Figure 5.4 and Figure 5.6 and the VHDL code found on the attached CD[25] that these two mitigation techniques were implemented the exact same way. Therefore, the SET suppressor is a more viable method of mitigation than the Guard gate filter technique designed by Actel.

All the other mitigation techniques did improve the SEU cross section. However, due to the fact that the TMR implemented in VGA TMR was not full global TMR, its SEU cross section was not the best. If the TMR implementation was full global TMR it most probably would have had the best cross section as was concluded by other tests conducted at iThemba labs [42].

The results obtained from the experiments indicated that the mitigation techniques did improve the VGA system's susceptibility towards SEU occurrences, however, as mentioned in this dissertation, all mitigation techniques come with a compromise in the area usage of the implementation. To conclude if the mitigation technique is worth compromising the area usage in the chip, one has to look at how much the combinational logic increased before and after mitigation techniques got inserted. VGA default had a total number of combinational logic cells of 139 as indicated in Table

[24] The combination logic elements are of a non-duplicated implementation
[25] Found under folder VHDL code

5.10. Table 5.11 indicates the factor that each mitigation technique increased the combinational logic by and the factor the mitigation improved the system's ability to protect itself from the effects of SEUs. The combinational logic consists of logic gates (i.e. ANDs and ORs) to memory elements (i.e. flip flops and latches).

Table 5.12 Factor of improvement compared to VGA Default at 2 nA

Implementation	**Factor of decreasing the SEU cross section compared to the VGA Default**[26]	**Factor of the combination logic increase compared to the VGA Default**[27]
VGA DMR	20	3.68
VGA GG	2.72	5.12
VGA SET Delay	1.1	2.04
VGA TMR	2.32	3.09
VGA MBU	9.71	2.24

It can be noted from Table 5.11 that the Guard Gate implementation increased the combinational logic by a factor of 5.12. This is more than all the other implementations, however, it only decreased the SEU cross section by a factor[28] of 2.72. The VGA DMR that was implemented in the same circuit structure as the Guard Gate implementation, increased the combinational logic by a factor of 3.68, however, it decreased the SEU cross section of the VGA system by a factor of 20. This is a significant decrease. Both these methods increased the combinational logic more than the TMR implementation. The TMR improved the SEU cross section by a factor of 2.32 and the combinational logic by a factor of 3.09. The implementation that increased the combinational logic the least was the VGA MBU implementation. The MBU only increased the combinational logic by a factor of 2.24 and decreased the SEU cross section by a factor of 9.71.

When one take these two factors into account one can conclude which mitigation technique is a viable option even if the increase in chip area is a consequence. TMR is the most well-known and most effective method known to protect circuits from SEU effects. Therefore, mitigation techniques need to be compared to TMR to decide if they are a better option than the conventional TMR. Since GG only improved the VGA system by a factor of 1.17 more than TMR but increased

[26] Factor was calculated by dividing the SEU cross section of the mitigation implementation with the non-mitigated VGA Default.
[27] Factor was calculated by dividing the combinational logic cells of the mitigation implementation with the non-mitigated VGA Default.
[28] All factors mentioned are compared with the VGA Default with no mitigation unless stated otherwise.

the combinational logic by a factor 1.65 more than TMR did, it can be concluded that the GG technique does improve the system's ability to mitigate the SEU's . However, the increase in combinational logic does not warrant its implementation over the use of the trusted TMR method.

The DMR with the AND-OR multiplexer decreased the system's SEU cross section by a factor of 8.62 and increased the combinational logic by a factor of 1.19. This indicates that the AND-OR multiplexer is a viable option to use instead of the TMR.

Another viable option that could replace the well-known TMR implementation is the MBU filter. This technique decreased the system's SEU cross section by a factor of 4.18 more than the TMR and increased the combinational logic 1.37 times less than the TMR.

It is important to note that in order to obtain the best possible results the test should have been conducted multiple times to be able to validate the results, unfortunately this was not possible since the iThemba Labs schedule did not allow for this.

5.2.4.1 Bendel 1 curve

Due to the fact that only one energy level[29] was used in testing for Single Event Upsets in the FPGA's, normal Weibull fit graphs will not suffice in plotting a cross sectional graph. As discussed in Chapter 2, W.L Bendel [64] designed a method to construct a cross section vs proton kinetic energy graph by using a one-parameter function.

The equation used to construct this graph is indicated below.

Equation 5.1 - Bendel Curve Equation

$$a = \left(\frac{24}{A}\right)^{14}\left(1 - \exp\left(-0.18Y^{0.5}\right)\right)^{4}$$

$$y = \left(\frac{18}{A}\right)^{0.5}(E - A)$$

[29] 66 Mev was used.

The formulas were entered into a Microsoft Excel[30] spreadsheet and the constant, A, was calculated for each implementation. The A values can be seen in Table 5.13 for tests conducted at 1nA and Table 5.14 for tests conducted at 2nA.

Table 5.13 Bendel Constant for 1 nA

Implementations	A
VGA Default	21.79
VGA TMR	23.4
VGA Delay	21.75

These A values above were inserted bank into the Bendel equation. Using Microsoft Excel, the Bendel graph, indicated in Figure 5.9, was constructed. Since errors only occurred for the three implementations indicated in Table 5.13 at 1 nA, these are the only implementations shown in Figure 5.9 using the Bendel equation.

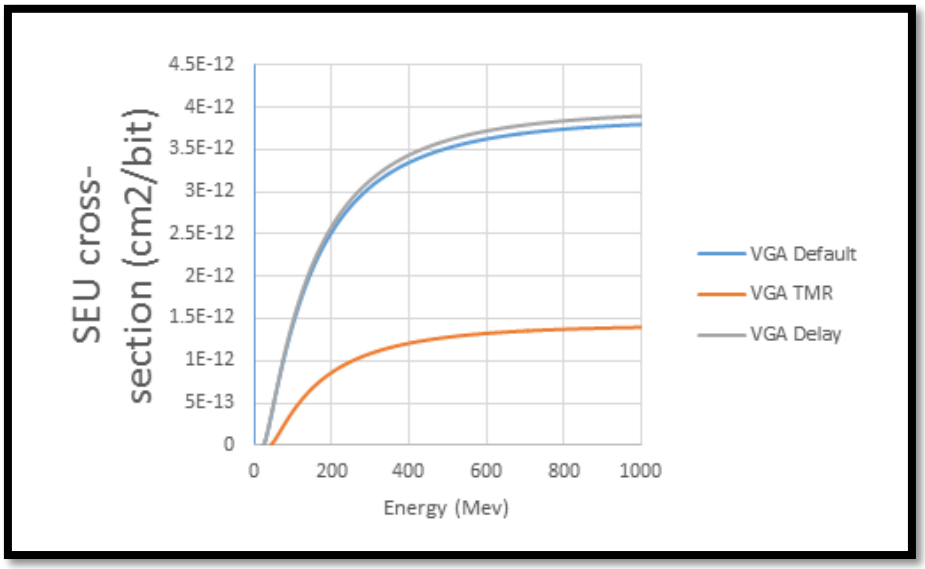


Figure 5.9 Bendel curve at 1nA

It can be seen in Figure 5.9 that the VGA Default and VGA Delay have a very similar cross section and that VGA TMR improved the system's capability to handle SEE's occurring. It was only at 2nA that errors were observed for all implementations. Therefore, at this beam current, a clear indication can be given as to which implementation was the best at eliminating SEE's in the VGA system.

[30] The excel spread sheet can be found on the CD in Folder called Excel Calculations

The same calculations that were done for the 1nA results were done for the 2nA results, using the same Excel spread sheet.

Table 5.14 Bendel Constant for 2nA

Implementations	A
VGA Default	21.46
VGA DMR	26.15
VGA TMR	22.7
VGA GG	22.93
VGA Delay	21.59
VGA MBU	24.94

It can be noted by equation 5.2 that the higher the constant A is, the lower the SEU cross section will be and vice versa. Knowing this and looking at Table 5.14, one can quickly note that VGA Default has the lowest value for A and VGA DMR the highest. When looking at the A values some of the values might seem close to one another, however, when the results are plotted on a Bendel curve graph, the differences between the A values becomes much clearer, as can be seen by Figure 5.10.

Figure 5.10 illustrates the SEU cross section vs the Proton Beam energy of all the implementations that were tested at 2nA.

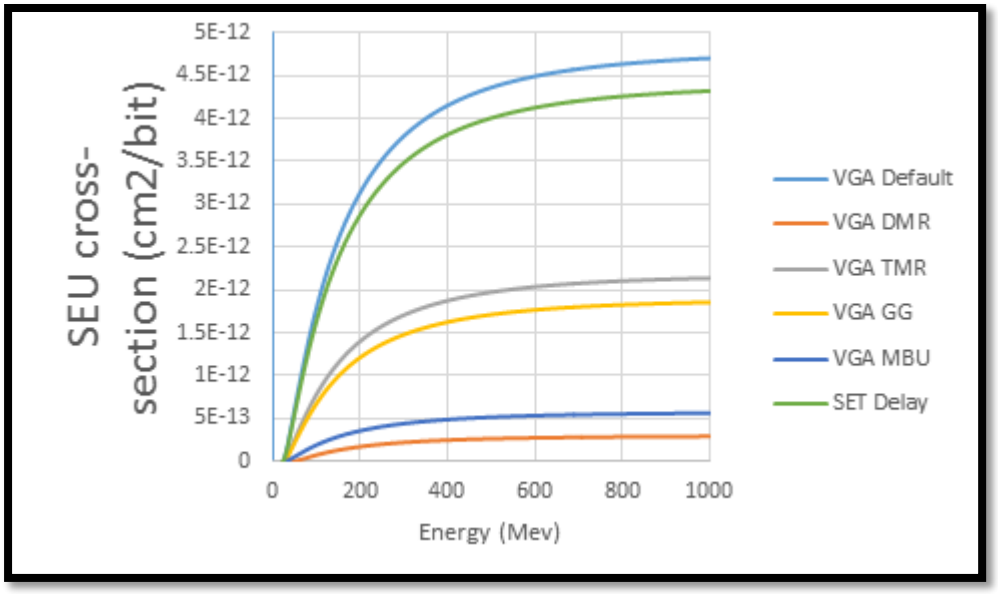


Figure 5.10 Bendel curve of Implementations of 2nA

This graph indicates how the SEU cross section per bit will change if the energy of the proton beam was to increase. This is a useful graph to have when only one energy-level is available for testing because it gives a theoretical means to indicate the effect the energy will have on the cross section as it is increased. It also indicates at what energy level the cross section remains the same even when the energy of the protons increases.

Looking at Figure 5.10 it is clear that the VGA Default was the most susceptible to SEU's occurring. This was expected since this was the only implementation that had no SEU protection. The best SEU protection technique, according to the graphs, is the VGA DMR with the AND-OR multiplexer. This graph confirms the comments made in the previous sections about each implementation and that for the tests conducted at NRF iThemba labs the AND –OR multiplexer mitigation technique gave the best results when it came to reducing SEU and SET effects on the system.

Chapter 6

6. Conclusion

This chapter will discuss the results that were obtained, further work that can be done in this field and improvements that could be made on the mechanical and electrical set up.

6.1 Overview

The work presented in this dissertation was carried out for the purpose of designing and building a SEE test set-up that could be used for testing small microelectronic DUTs at NRF iThemba labs. The normal procedure of testing different SEU mitigation technique involves using string of inverters as the combinational logic, however, this dissertation focused on a practical application which is used in a satellite system. The application chosen was a VGA controller as South Africa

is looking at investing in earth observation satellites; therefore, this application was ideal to test for SEE's.

Using the FPGA boards mounted onto the mechanical set-up, different mitigation techniques were compared to one another in attempt to mitigate any errors that occurred in the VGA controller and conclude which mitigation technique is the most effective.

The SEE mechanical set-up was one of the first set-ups of its kind which allowed for the testing multiple DUTs under one vacuum session. This gave the opportunity for testing multiple designs on different DUTs. Five different mitigation techniques were tested to observe their capability of protecting the VGA controller from SEE effects.

All the mitigation techniques that were tested improved the system's vulnerability against SEEs. Some of them filtered out SETs before they could become SEUs whilst other techniques voted out the SEU.

The OR-AND Multiplexer SET filter offered a simple and effective method to reduce the number of upsets in the VGA system. The SET filter could be implemented in two different methods: one with a delay element and the other with a common DMR implementation. The SET filter with a 2-inverter delay was implemented at the input of every memory element[31] in the VGA system with no modular redundancies implemented on the any of the memory elements; this was not effective at all. It only gave a small improvement compared to the default, non-protected, VGA controller. The SET filter was then implemented with DMR on the combinational logic and local TMR on the flip flops. This gave the best results against SEE upsets. This concluded that a SET filter without some sort of redundancy, be it local TMR or DMR, is not an effective method to mitigate SEEs in a system.

Triple modular redundancy improved the system significantly; surprisingly though, it was not the best mitigation technique that was tested. This could have been due to the fact that it was not full global TMR as the clock and reset lines were not doubled and only one majority voter was used instead of the traditional three.

[31] D Flip Flop

The alternative guard gate implementation which made use of the NAND C element filter was implemented in the same manner as the OR-AND Multiplexer SET filter. This mitigation technique improved the system by a factor of 2.32. This is similar to the improvement factor of the TMR of 2.72.

The final implementation was a mitigation technique for Multiple Bit Upsets. This technique is unlike other MBU technique that has been previously used (i.e. hamming code) which involves complicated correction code, whereas this technique replaces the traditional majority voter. The MBU filter was used in place of the majority voter in the same set-up as the TMR. The probability of a multiple-bit upset occurring at 66Mev was low; therefore, an MBU was forced by connecting the output of one flip flop to the set/reset of another flip flop. The MBU filter proved to be a viable option for SBU and MBU filtering as it improved the system's resistance to SEUs and MBUs by four times more than the traditional TMR circuit.

The results indicate that mitigation techniques are viable options in protecting circuits in a practical application. The OR-AND Multiplexer SET filter decreased the SEU cross section of the VGA system more than any other implementation and only increased the combinational logic elements 1.19 times more than TMR implementation. The mitigation technique that had the most promising results was the MBU filter technique this technique decreased the systems SEU cross section by a factor of 9.72, which is 4.18 times more than TMR and the MBU filter only increased the combinational logic by a factor of 2.24, this is less than TMR that increased the combinational logic by 3.09. This is a significant finding as this technique mitigates for MBUs as well while still having a less overhead than that of the TMR implementation.

6.2 Improvements

Despite all the work done and results obtained, there is still room for improvement of (the mechanical and electrical set up and implementations.in some areas.

6.2.1 Mechanical Set-up Improvements

The mechanical set-up which was built did satisfy the requirements that were needed for adequate SEE testing; however, improvements could still be made. Since there are many motors which are used to rotate and lift the structure, there will be some electrical noise created by these motors. This made the length and neatness of the wires very important as noise can influence the results.

This could be improved by shielding each motor and ensure the motor wires and the wires carrying the error signals do not cross one another.

6.2.2 Electrical Set-up improvements

The electrical set-up consists of the DUT, monitoring and control board and the NI DAQ card used to display the errors on the computer.

6.2.3 DUT improvement

The DUT could be improved by designing and creating a PCB[32] board that is application-specific for a VGA controller. The development board served its purpose as it was easy to interchange FPGA chips once they malfunctioned, however, a smaller board that is VGA-specific will have fewer electronic components that could potentially be affected by the radiation.

Extra features could also be added to the PCB which could improve analyses and results i.e. chip temperature and core current.

6.2.4 Monitoring and control board

The monitoring and control board was used to compare signals and capture errors. It successfully completed the required operations and gave the correct output when two signals did not match up perfectly. This monitoring board ran at a faster frequency than that of the DUT, therefore, it did not miss any upsets that occurred. The only improvement that could be made is to either use a Flash based FPGA or to upload the program to an external flash memory on the DE2 Development board. Since the program was uploaded on the SRAM module, the program had to be uploaded to the FPGA each time it lost power. The vacuum chamber took 5 hours to create a vacuum that was adequate for testing to commence, therefore, if anything had to happen to the power of the control board while the vacuum was being created, the vacuum-pumping process would have to be stopped and then restarted once the control board was reprogrammed

Alternatively, a connector could be designed for the vacuum chamber which provides a facility to program the FPGAs from outside of the vacuum chamber.

[32] An PCB design can be found on the CD under PCB design files

6.2.5 DAQ card

The DAQ card was used to read the number of errors that the control board has counted. It was connected to a laptop computer 80 meters away from the testing chamber via an Ethernet cable; this laptop computer displayed the incoming data on a GUI interface. The main problem was that this DAQ card only operates at 143 kHz. This required the monitoring and control board to be programmed to latch all the errors it counted to ensure the DAQ did not miss any errors that had been captured by the control board.

This could be improved by using a microprocessor to capture the data and store the errors capture in a memory module. A C program could then be written to transfer the errors using the Ethernet protocol and display it on a computer monitor. This method will require more complicated programming than is required with LabView, however, one would be guaranteed that no errors were miscounted.

6.2.6 Implementation Improvements

Since all the implementations were programmed on one single FPGA chip, the chip area and resources had to be shared. The ideal testing situation would have each implementation occupy 100% of the FPGA memory as well as have many test runs for a single implementation to get an average of the errors that occurred, to reassure the repeatability of the test. However, this was not possible due to the limited beam time that was granted by NRF iThemba LABS.

Different energy levels of the proton beam would have given a more variety to the test results. It was mentioned by Dr Nchodu from NRF iThemba labs that because there is now a permanent SEU set-up available at the facility, future testing at this facility can be granted longer beam time at different energy levels.

6.3 Further work

Since this dissertation has proved that mitigation techniques are a viable option in a practical space application, further research can now be conducted in more complicated applications. Since a VGA controller is used on satellites, the method in which it gets radiated in space is different to the

testing that is conducted on Earth because a satellite has an outer shell and many other electrical components, therefore, the amount of scattering that is caused in space cannot be compared to the test conditions here.

One component which can be incorporated into the current mechanical set-up is to design a structure to hold a completed CubeSat. This will enable the testing of a complete satellite system and obtain more realistic results. This set-up further will allow further investigation into how to improve the SEE mitigation software, the type of shielding which could be used or finding the most ideal place for electronics to be situated within a CubeSat.

This dissertation indicates that there is still room for improvement of the mitigation techniques which are currently being used today. Although most of them execute correctly and do protect the circuit in some form or another, this is done at the expense of the amount of area used on the chip and the power consumption. As the sizes of components are decreasing, the critical charge of the components is decreasing as well. This will lead to SEEs occurring more regularly in space and even on the ground. Therefore, mitigation techniques need to adapt with this change. Evolutionary computing might be a solution to this problem. A self-adapting mitigation technique may be ideal since (SEEs occurrence are about probability and it is not guaranteed that an upset will occur. A self-adapting mitigation technique will only adapt when an error is observed. This could save on chip area space.

Another observation that came up in this dissertation is that when a SET filter is used with a delay element, it is only effective if the SET pulse width is less than that of the delay element. Research could be conducted on developing a self-adapting mitigation were a set pulse width is measured and the system will self-adapt to insure that the time delay of the delay element is always longer than that of the SET. This will, however, only protect the system's combinational logic and not the memory elements in the system.

References

[1] F. Smith, “Single event upset mitigation by means of a sequential circuit state freeze,” *Microelectronics Reliability*, vol. 52, no. 6, pp. 1233-1240, 2012.

[2] T. Holman, “Radiation hardening tecqniques for analog and mixed signal intergrated circuits,” Holman,T, 2008.

[3] J. Srour, “Radiation effects R&D in the 1970s: a retrospective view,” *IEEE Nuclear and Plasma Sciences Society*, vol. 41, no. 6, pp. 2660-2665, 1994.

[4] W. Harris, “Dev.Mat.Reliab,” *IEEE Trans*, no. 5, pp. 336-342, 2005.

[5] P. Bradley, “Single Event Upsets in Impantable Cardioverter Defibrillators,” *IEEE Nuclear Science*, vol. 6, no. 45, pp. 2829-2940.

[6] D. Sinclair, “Radiation Effects and COTS parts in SmallSats,” in *27th Anual AIAA/USU Confrence on Small Satelites*.

[7] J. Barth, *What is space radiation?*, NASA Space Radiation Analysis Group, Johnson Space Center.

[8] d. Soria-Snatacruz, “Design of a space-borne antenna for controlled removal of energetic Van Allen belt protons,” in *Aerospace Conference, 2014 IEEE*, Montana, 2014.

[9] H. Richard, “Harsh Enviroments: Space Radiation Enviroment,Effects and Mitigation,” *Technical Digest,* vol. 28, no. 1, pp. 17-28, 2008.

[10] F. Sturesson, “Space radiation and its Effects on EEE components,” *EPFL Space,* vol. III, p. 1, 2009.

[11] J. Barth, *Space & Atmospheric Environments,* NASA/Goddard Space Flight.

[12] L. Biermann, “Solar Corpuscular Radiation and the Interplanetary Gas,” *The Observatory,* vol. 77, pp. 109-110, 1957.

[13] E. Parker, “Interaction of Solar Wind with the Geomagnetic Field,” *Physics of Fluids,* vol. 1, no. 3, pp. 171-187, 1958.

[14] J. Barth, “NASA Technical Reports Server,” 14 February 2002. [Online]. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20030053331.pdf. [Accessed 24 November 2014].

[15] V. Arlington, “Test procedures for the mesurments of single-event effects in semiconductor devices from heavy ion irradiation,” *EIA,* p. 49, 1996.

[16] M. A. Clemens, *ENERGY DEPOSITION MECHANISMS FOR PROTON- AND NEUTRON-INDUCED SINGLE EVENT UPSETS IN MODERN ELECTRONIC DEVICES,* Nashville, Tennessee, 2012.

[17] P. Dodd and S. F, “Critical charge components for CMOS SRAMs,” *IEEE Transactions on Nuclear Science,* vol. 42, no. 6, pp. 1764-1771, 1995.

[18] J. Kim and C. Nicopoulos, “A probabilistic model for soft-error rate estimate in combinational logic,” 1st International workshop on Probabilistic Analysis Tecqniques for Real-Time and Embedded Systems, 2004.

[19] J. Leavy and R. Poll, “Radiation-Induced Intergrated Circuit Latchup,” *IEEE Transaction on Nuclear Science,* vol. 16, no. 6, pp. 96-103, 1969.

[20] M. Nicolaidis, “Soft Errors in Modern Electronic System,” New York, Springer, 2011, pp. 70-80.

[21] P. Dodd, “Radiation e↵ects in the space telecom environment: Ii. basic mechanisms,” in *IEEE NSREC Short Course*, 1999.

[22] A. Kobayashi and V. Univ, “The effect of metallization Layers on single event susceptibility,” *IEEE Transactions on Nuclear Science,* vol. 52, no. 6, pp. 2189-2193, 2006.

[23] R. Reed, “Implications of Nuclear Reactions for Single Event Effects Test Methods and Analysis,” *IEEE Transactions on Nuclear Science,* vol. 53, no. 6, pp. 3356-3362, 2006.

[24] P. McNutty, “Proton-Induced Nuclear Reactions in Silicon,” *IEEE Transactions on Nuclear Science,* vol. 28, no. 6, pp. 4007-4012, 1981.

[25] K. Holbert, “Keith E. Holbert, Ph.D., P.E.,” The Aerospace Corparation , 18 January 2006. [Online]. Available: http://holbert.faculty.asu.edu/eee560/see.html. [Accessed 4 Ocktober 2014].

[26] T. Tohmatsu, “Compendium of Aeronomy,” Japan, TERRAPUB, 1990, p. 38.

[27] M. Johnson, “88-Inch Cyclotron,” Webmaster, [Online]. Available: http://cyclotron.lbl.gov/base-rad-effects/protons. [Accessed 06 November 2014].

[28] B. Von Przewoski, “Indiana University Cyclotron Facility,” May 2006. [Online]. Available: http://iuco.indiana.edu/doc/user-guide-technical.pdf. [Accessed 14 Novemebr 2014].

[29] Texas A&M University, “Radiation Effects Facility,” [Online]. Available: http://cyclotron.tamu.edu/ref/chamber.php. [Accessed 25 June 2013].

[30] R. Bark, “Radioactive Beam Proposal for iThemba LABS,” 2011. [Online]. Available: http://www.tlabs.ac.za/pdf/RIB/RobWorkshop2011.pdf. [Accessed 20 April 2014].

[31] J. Carnaris and S. Whitaker, “SEU hardened memory cells for CCSDS Reed Solomon encoder,” *IEEE trans Nuclear Science,* vol. 502, no. 38, pp. 1471-1477, 1991.

[32] R. B. A. N. B. B. B. O. N. P. Shuler, “The Effectiveness of TAG or Guard-Gates in SET Suppression Using Delay,” *IEEE Transactions on Nuclear Science,* vol. 53, no. 6, pp. 3428-3431, 2003.

[33] S. Rezgui and W. JJ, “New methodologies for SET characterization and mitigation in flash-bassed FPGAs,” *IEEE Trans Nuclear Science,* vol. 54, no. 6, p. 469, 2006.

[34] A. Balasubramanian and B. Bhuva, “RHBD techniques for mitigating effects of single event hits using guard-gates,” *IEEE trans Nuclear Science,* vol. 52, no. 6, pp. 2531-2535, 2005.

[35] S. Mitra, “Combinational Logic Soft Error Correction,” in *Test Conference, 2006. ITC '06. IEEE International*, Santa Clara, CA, 2006.

[36] F. Smith, “A new methodology for single event transient suppression in Flash FPGA's,” vol. 37, no. 3, pp. 213-318, 2013.

[37] S. Write, “SourceTech,” 28 April 2013. [Online]. Available: http://sourcetech411.com/2013/04/top-fpga-companies-for-2013/. [Accessed 1 November 2014].

[38] F. R. Kastensmidt, “Fault-Tolerance Techniques for SRAM-Based FPGAs,” Netherlands, Springer, 2006, pp. 60-63.

[39] D. Choi, “New Non-Volatile Memory Structures for FPGA Architectures,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol. 16, no. 7, pp. 874-881, 2008.

[40] J. Wang, “CERN Accelerating science,” [Online]. Available: http://cds.cern.ch/record/712037/files/p34.pdf.

[41] Microsemi, “FPGA Products Update,” 2013.

[42] F. Smith, *66 MeV Proton Testing of the AND-OR Multiplexer SET filter circuit implemented on a Flash FPGA*, Cape Town, 2014.

[43] K. Khurshid, “A survey of camera modules for CubeSats - Design of imaging payload of ICUBE-1,” in *Recent Advances in Space Technologies (RAST), 2013 6th International Conference on*, Istanbul, 2013.

[44] D. R. Velazco, “Single Event Effects on digital intergrated circuits: Origins and Mitigation Tecqniques,” 19 May 2011. [Online]. Available: http://www.in2p3.fr/actions/formation/microelectronique11/Velazco%20%20SEU%20and%20Latchup.pdf. [Accessed 05 January 2014].

[45] D. O'Neal, “Commercial communications satellites: history and highlights,” *Microwave Symposium Digest, 1999 IEEE MTT-S International*, vol. 4, pp. 1919-1921, 1999.

[46] S. Garber, “NASA,” [Online]. Available: http://history.nasa.gov/conghand/navsat.htm#REF23-1. [Accessed 4 November 2014].

[47] T. R. W. McCaskill, O. Oaks and R. Beard, “Performance of global positioning system (GPS) on-orbit Navstar clocks,” in *Frequency Control Symposium, 1995. 49th., Proceedings of the 1995 IEEE International*, San Francisco, CA, 1995.

[48] Defense Industry Daily staff, “Defence Industry Daily,” 24 Aug 2005. [Online]. Available: http://www.defenseindustrydaily.com/the-gps-constellation-now-and-future-01069/. [Accessed 25 November 2014].

[49] SANSA, “South Africa National Space Association,” 12 September 2014. [Online]. Available: http://www.sansa.org.za/spacescience/resource-centre/news/724-cme-from-solar-flare-heads-towards-earth. [Accessed 05 October 2014].

[50] “Weather Questions,” 1 November 2011. [Online]. Available: http://www.weatherquestions.com/How_do_weather_satellites_work.htm. [Accessed 25 Septermber 2014].

[51] South African Space Association, *Space Youth Congress*, Cape Town, 2014.

[52] P. Blau, “spaceflight101,” 28 02 2014. [Online]. Available: http://www.spaceflight101.com/33-cubesats-released-from-international-space-station-february-2014.html. [Accessed 10 March 2014].

[53] S. Arnold, “Energy budgeting for CubeSats with an intergrated FPGA,” in *Aerospace Conference IEEE*, Montana, 2012.

[54] D. Selva and D. Krejci, “A Survey and assesment of the capabilities of cubesats for Earth observation,” *Acta Astronautica*, vol. 74, pp. 50-68, 2012.

[55] K. Woellert, “Cubesats: Cost-effective science and technology platforms for emerging and developing nations,” *Advances in Space Research,* vol. 47, no. 4, pp. 663-684, 2010.

[56] “Hamsterworks,” 28 June 2013. [Online]. Available: http://hamsterworks.co.nz/mediawiki/index.php/OV7670_camera. [Accessed 05 May 2014].

[57] R. Read, P. Marshal, H. Kim, McNutty and P.J, “Evidence for angular effects in proton-induced single-event upsets,” *IEEE transaction on Nuclear Science,* vol. 49, no. 6, pp. 3038-3044, 202.

[58] VARVEL, “VARVEL,” [Online]. Available: http://www.varvel.com/. [Accessed 24 September 2013].

[59] G. Lazaridis, “PCB Heaven,” 7 May 2010. [Online]. Available: http://www.pcbheaven.com/wikipages/How_Stepper_Motors_Work/. [Accessed 10 September 2014].

[60] G. Donzellini, “Deeds FPGA Extension,” [Online]. Available: http://www.esng.dibe.unige.it/deeds/LearningMaterials/LM/Tutorials/DE2/DE2_layout_1000m.JPG. [Accessed 28 October 2014].

[61] J. Insoo, “Effects of secondary particles on the total dose and the displacement damage in space proton environments,” *IEEE Nuclear and Plasma Sciences Society,* vol. 48, no. 1, pp. 162-175, 2002.

[62] S. Rhee, “Concatenated Reed-Solomon Code with Hamming Code for DRAM Controller,” in *Computer Engineering and Applications (ICCEA), 2010 Second International Conference*, Bali Island, 2010.

[63] D. Mavis, “Multiple Bit Upsets and Error Mitigation in Ultra-Deep Submicron SRAMS,” *EEE Nuclear and Plasma Sciences Society,* vol. 55, no. 6, pp. 3288-3294, 208.

[64] W. Bendel, “PROTON UPSETS IN ORBIT,” *IEEE Transactions on Nuclear Science, Vol. NS-30, No. 6,,* p. 4481, 1983.

[65] Microsemi, “Packaging FAQs,” 2011.

[66] M. D, “SEU and SET mitigation tecqniques for FGA circuit and configuration bit design,” in *Military and Aerospace Applications of Programmable Devices and Technology*, 2000.

[67] A. Corporation.United States of America Patent 828,821, 23 December 2010.

[68] D. N, R. R and M. M, “Charge generation by secondary particles from nuclear reactions in beol materials,” *IEEE Transaction on Nuclear Science,* vol. 56, no. 6, pp. 3172-3179, 2009.

Appendices

Appendix A

Scattering of the proton beam

The scattering of the proton beam was simulated using software called The Stopping and Range of Ions in Matter (SRIM). This program allows the user to enter the parameters required for simulating scattering, which are:

- Proton Element – Hydrogen was used
- Protons energy – 66Mev
- Target information – The viewer consisted of aluminium oxide. The outer shell of the FPGA is predominantly copper and the inside of the FPGA consist of silicon.

The user interface of the scattering software used can be seen in Figure A.1

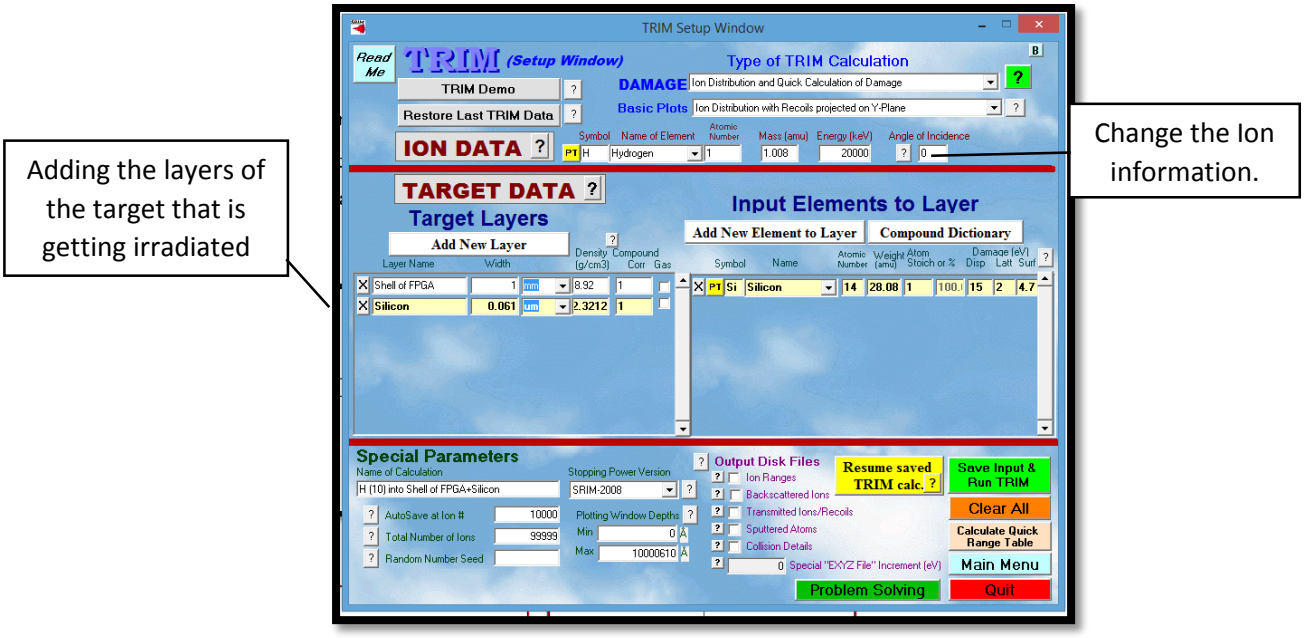


Figure A. 1 - User interface for TRIM

The software requires the user to enter the correct information of the system that is required to be analysed. In this dissertation the system is the system inside the vacuum chamber, consisting of the viewer and the FPGA that is being irradiated. The system can be observed in Figure A.2.

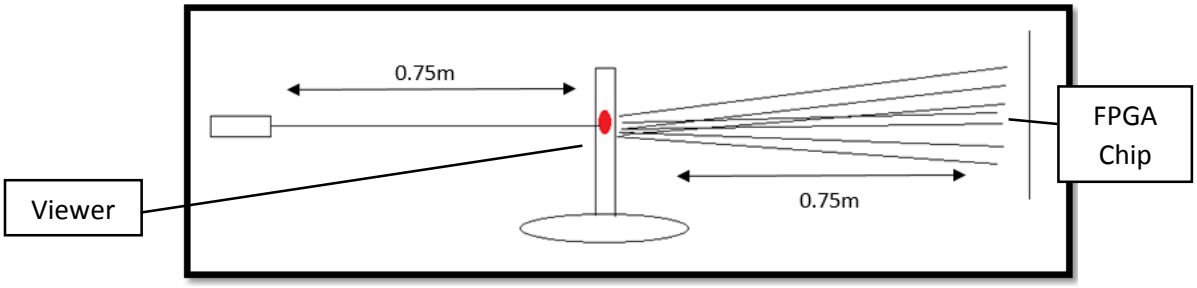


Figure A.2 - Diagram of viewer in scattering chamber

The diagram above illustrates the different aspects that are radiated while testing. The viewer that is 0.75m away from the entrance of the proton beam is the first target that is hit by the proton beam. The Second target that is in the path of the beam is the FPGA chip. It is important to note a FPGA chip consist of different materials, however the most prominent material is copper for the outer shell and silicone for the circuitry inside the chip [65].

Initially just the effect of the outer shell of the FPGA was tested. Figure A.3 and A.4 illustrated the scattering effect the outer shell has on the proton beam.

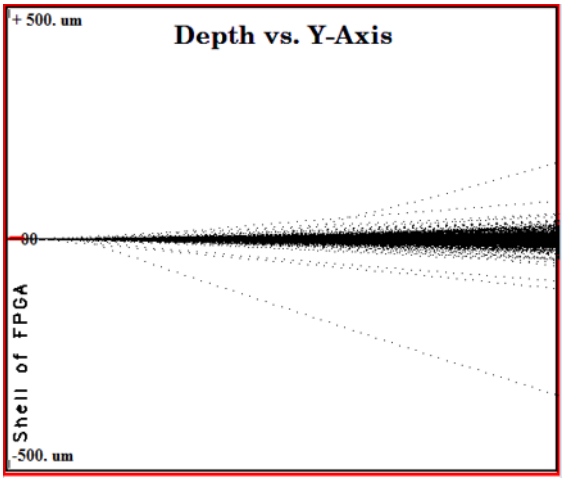


Figure A.3 - Scattering of proton beam caused by FPGA casing, viewed from the side

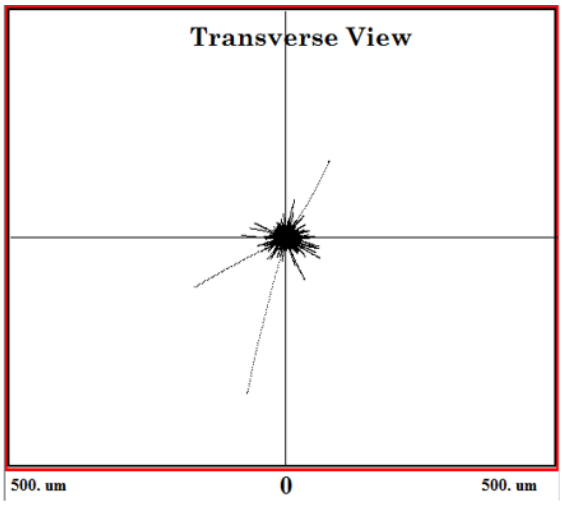


Figure A.4 - Scattering of proton beam caused by FPGA casing, viewed from the front on

It is clear that the outer shell of the FPGA has a scattering effect on the proton beam, however, it is important to take not of the axis of the graphs. The Y axis and X axis range from -500um to +500 um. Therefore what looks like a big variant of scattering, is in fact not and negligible

The next target that was simulated was the viewer. The viewer consist of aluminum oxide. The viewer is 0.75m away from the FPGA chip, therefore a small angle of scattering at the viewer will be enlarged significantly at the FPGA chip. Figure A.5 illustrates the scattering caused by the viewer.

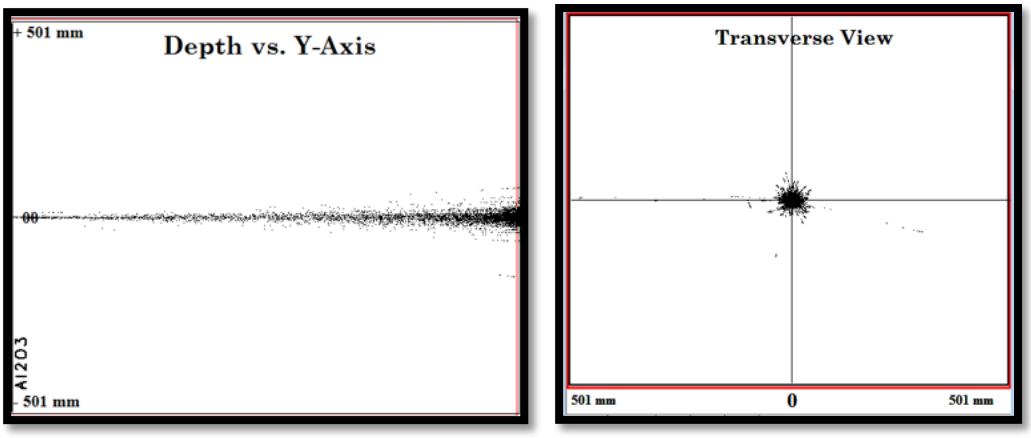


Figure A.5 - Scattering of proton beam caused by FPGA casing, viewed from the side and from front on

In this instance the Y-axis was from 501mm to -501mm. This indicates that the viewer causes more scattering than the outer shell of the FPGA. This is only due to the fact that the viewer is placed at a further distance away from the FPGA increasing the angle of scattering.

It was mentioned in the dissertation that the lower the energy of a particle is the higher its LET is, which transpires to more errors occurring. A simulation was conducted to observe the effect a lower energy will have on the scattering of the particle, this can be seen by Figure A.6

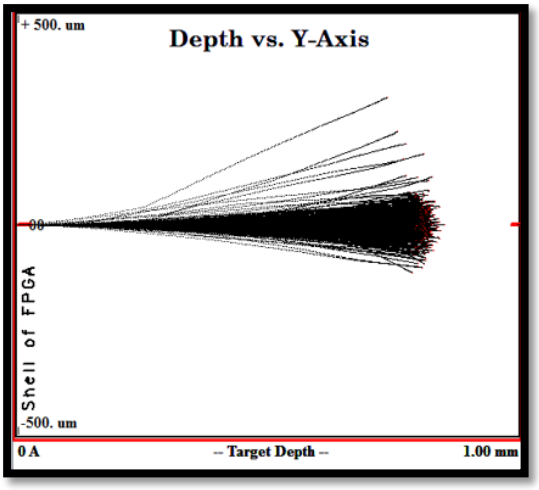


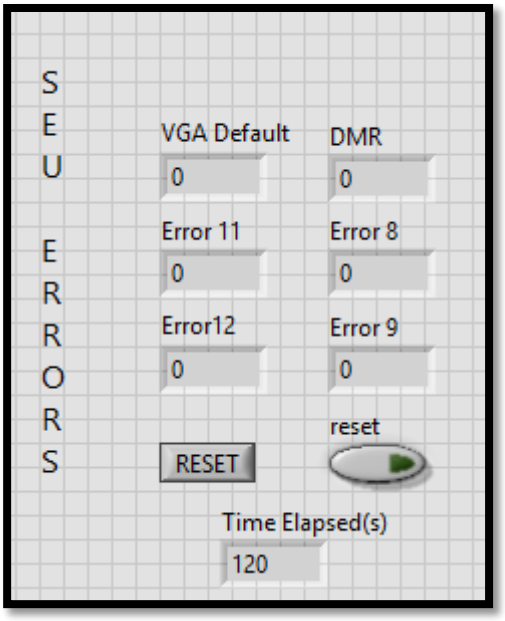
Figure A.6 - Scattering of the beam caused by the FPGA outer Shell at 20 Mev

The beams energy was reduced to 20Mev instead of the 66Mev used in the previous simulations. This simulation showed that at 20Mev the particle does not have enough energy to penetrate through the FPGA casing. This means that all the particle energy was deposited inside of the FPGA chip, which leads to more upsets.

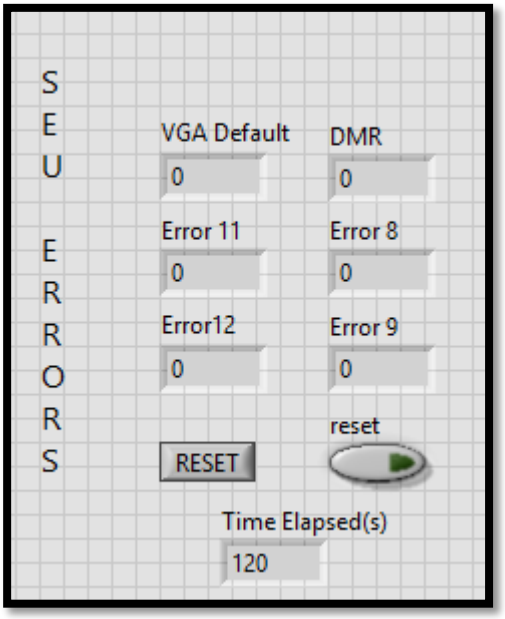
Appendix B

Appendix C

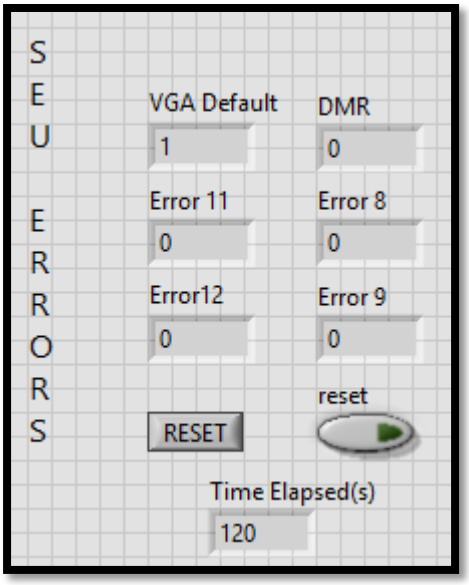
Labview Raw Data of 1st Day at 1nA no Viewer



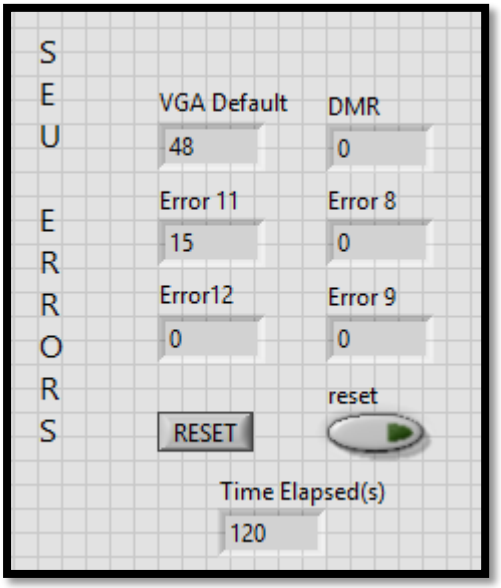
Labview Raw Data of 1st Day at 2nA no Viewer



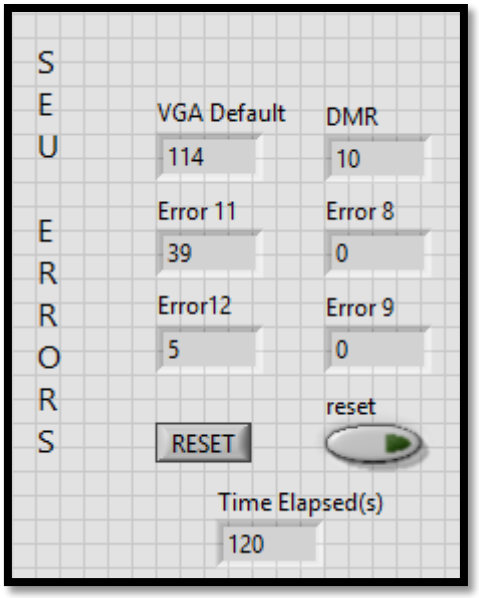
Labview Raw Data of 1st Day at 4nA no Viewer



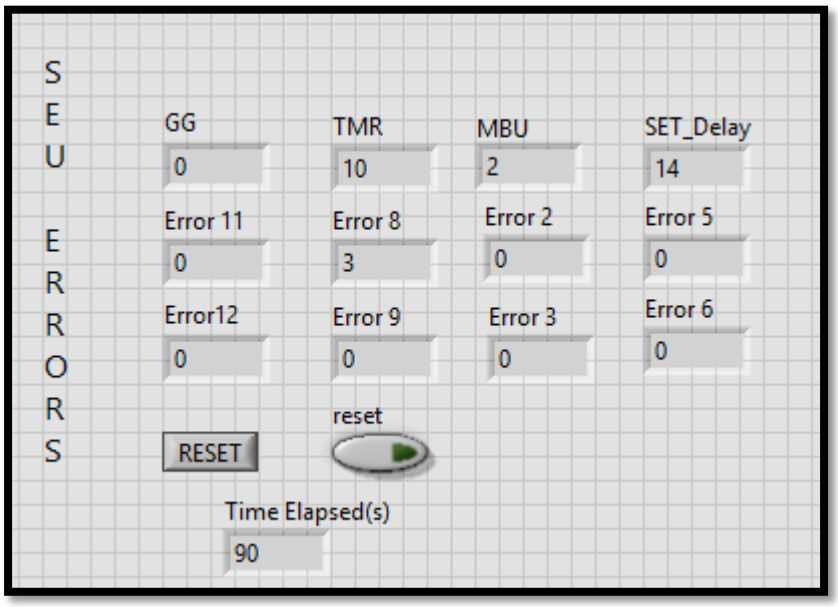
Labview Raw Data of 1st Day at 1nA with Viewer



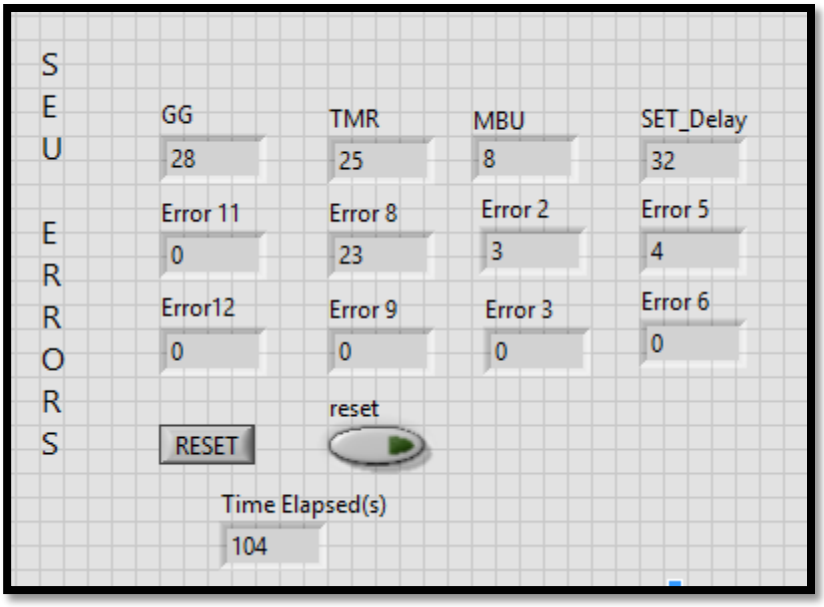
Labview Raw Data of 1st Day at 2nA with Viewer



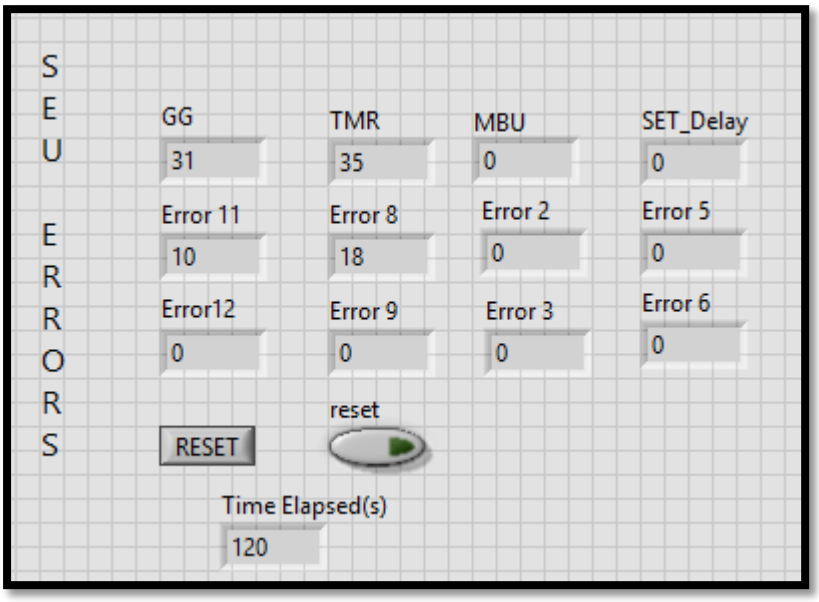
Labview Raw Data of 2nd Day at 1nA



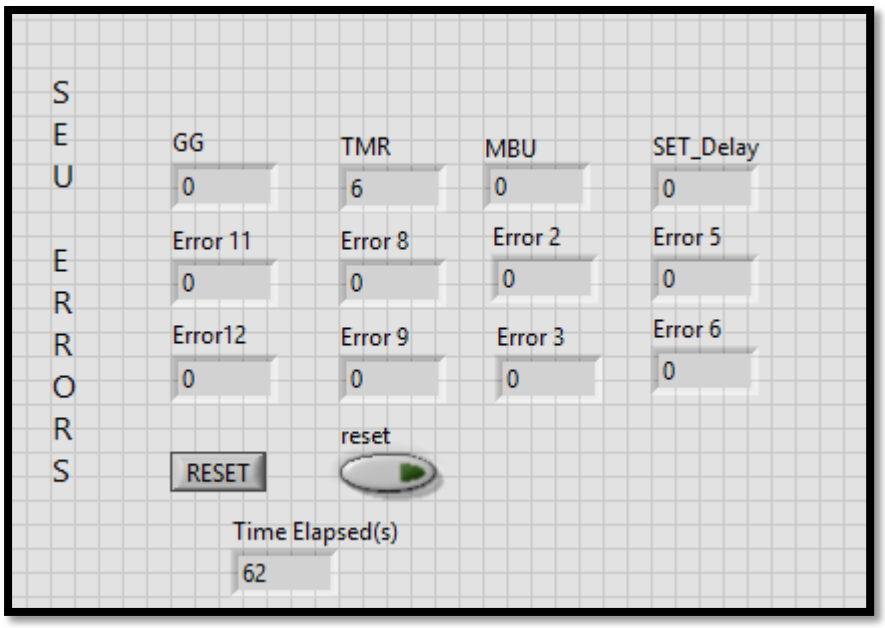
Labview Raw Data of 2nd Day at 2nA



Labview Raw Data of 2nd Day at 3nA



Labview Raw Data of 2nd Day at 4nA


SEU
ERRORS
GG
0
TMR
6
MBU
0
SET_Delay
0
Error 11
0
Error 8
0
Error 2
0
Error 5
0
Error12
0
Error 9
0
Error 3
0
Error 6
0
RESET
reset
Time Elapsed(s)
62

Appendix D