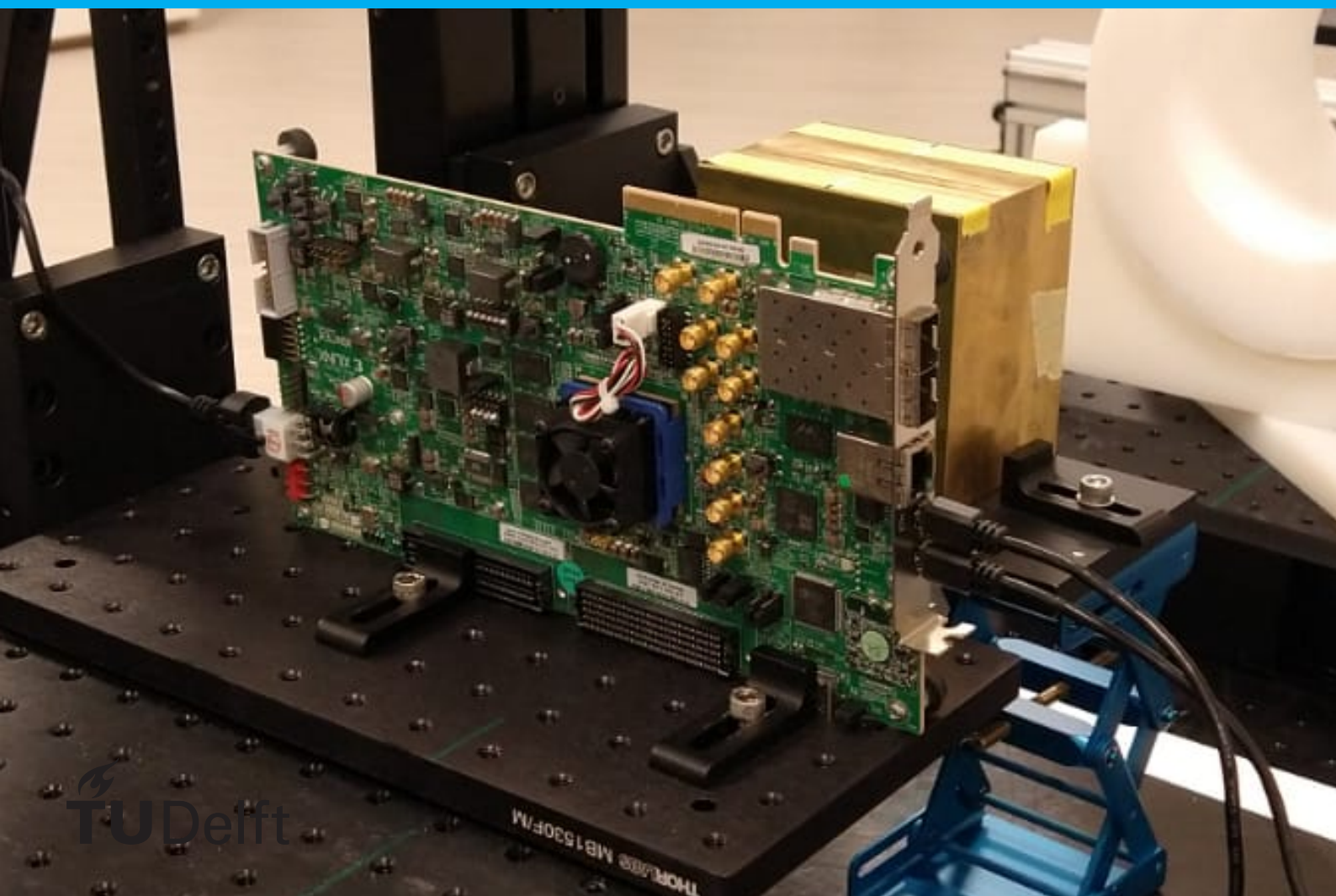


# Single Event Up- set Characterisa- tion of NOEL-V soft processor

T. Hendrix

**Supervisors:** A. Menicucci  
S. Di Mascio





# Single Event Upset Characterisation of NOEL-V soft processor

by

T. Hendrix

to obtain the degree of Master of Science  
at the Delft University of Technology,

Student number: 4446143  
Project duration: Februari 15, 2021 – January 12, 2022  
Thesis committee: Dr. A. Menicucci, TU Delft, supervisor  
S. di Mascio, TU Delft, supervisor



# Executive summary

This thesis evaluates the susceptibility of the NOEL-V soft processor (SP), a promising and highly modular soft processor by Cobham Gaisler. The processor is promising as it makes use of the RISC-V architecture, an open source instruction set architecture. The openness of the RISC-V makes for numerous benefits such as improvement in security and fault-tolerant fields [1]. The RISC-V architecture is seen as a possible successor for the now used SPARC ISA, which has been used for all recent space missions conducted by ESA.

In order to characterise the performance of the NOEL-V SP in the harsh space radiation environment, the KCU105 development board is used as Device Under Test (DUT) and irradiated with high energy protons. A toolchain to synthesise the processor on the DUT is provided by the processors vendor and together with the the DUT itself making use of the same silicon as a promising radiation tolerant FPGA. This FPGA is both practical and interesting to investigate. Radiation tests have provided valuable information about the behaviour of the SP in the presence of a harsh space environment. Among others, debug link, energy dependency and memory susceptibility are investigated, with the main goal being the radiation characterisation of the NOEL-V processor as a whole and the influence of microarchitectural differences between different configurations of the SP.

The NOEL-V is provided as a highly configurable synthesizable VHDL model. Due to this configurability, multiple different architectures of the NOEL-V can be tested and microarchitectural differences can be exposed. In this research these microarchitectural differences will be used to obtain data about the different parts of the processor. It is known that microarchitectural differences are influential for the radiation susceptibility of multiple designs based on the same ISA [2].

The test setup shows that through backside irradiation it is not necessary to remove any material from the PCB and the added fan will not be in the beam path. Testing with different energy protons resulted in a very small decrease of upset rates at 70 MeV as opposed to 150 MeV (within 10%), and it was thus concluded that the proton threshold is reached at 70 MeV. Spread out over three separate tests, a combined test time of 174 minutes is reached in which  $2.14 \cdot 10^{11} p/cm^2$  were fired at the DUT.

User logic upsets and configuration SRAM upsets are both extracted and compared for multiple configurations of the processor. Although the higher performance configurations utilise more resources of the FPGA and are, therefore, expected to be subject of more upsets in configuration memory, this has not necessarily proven to be the case. CRAM cross sections are found to be comparable to earlier works on cross sections of the used DUT.

Upsets in caches are found at similar cross section to the CRAM upsets, whereas ROM memory is shown to be much less susceptible to upsets, showing a cross section in the order of  $10^{-4}$  less.

The biggest influence on user logic upsets is observed to be the use of an operating system. A decrease in susceptibility is found when employing the floating point unit of the processor, and the most advanced processor configuration is found to be least susceptible to radiation effects. The inclusion of L2Cache being the most likely reason for this decrease. An in-orbit failure rate of one failure every 395 days is found for for a 51.6 ° circular orbit at 420 km altitude.

Findings indicate that the NOEL-V processor, with the implementation of targeted fault tolerant measures, can be a viable choice for space missions. Due to its modularity, the processor can be used for a multitude of mission types ranging from high performance general-purpose to low-end microcontroller applications. Error Detection And Correction will be needed to protect make sure upsets in caches do not lead to a failure of the processor.



# Acknowledgements

I am glad to have had the opportunity to pursue this thesis topic as the pinnacle of my academic career. Besides the circumstances being one of the hardest ever occurring in my life, I am happy to have had this great opportunity to learn an enormous amount of new things in the field of electronics and to further my knowledge of space related topics. This endeavour will forever sculpt my life and all further projects I pursue.

I want to express my appreciation for the great assistance by A. Menicucci and S. di Mascio. Without your assistance I would not have been able to reach this goal. I would also like to acknowledge the support of my relatives, friends, roommates, and the thoughtfulness and English skills of my girlfriend, who have helped me to reach this pinnacle of my academic career, and above all, keep my sanity in these uncertain times.

*T. Hendrix  
Delft, December 2021*





# Contents

<b>1</b>	<b>Introduction and motivation</b>	<b>1</b>
1.1	Motivation for research . . . . .	1
1.2	Literature study and research motivation . . . . .	3
1.2.1	Electronics in space . . . . .	3
1.2.2	Proton radiation testing guidelines. . . . .	7
1.2.3	Susceptibility of FPGA . . . . .	7
1.3	Research framework . . . . .	11
1.3.1	Research question(s). . . . .	11
1.3.2	Objective . . . . .	12
1.4	Methodology . . . . .	12
1.4.1	The microprocessor . . . . .	13
1.4.2	Radiation testing . . . . .	14
1.4.3	Data handling. . . . .	16
1.5	Thesis outline. . . . .	16
<b>2</b>	<b>Test setup</b>	<b>17</b>
2.1	Facility. . . . .	17
2.2	Test setup. . . . .	18
2.3	Beam properties . . . . .	20
<b>3</b>	<b>NOEL-V soft processor</b>	<b>23</b>
3.1	General . . . . .	23
3.1.1	IP cores . . . . .	23
3.1.2	Component resource utilisation . . . . .	26
3.1.3	Existing mitigation measures . . . . .	28
3.2	NOEL-V example designs . . . . .	28
3.3	Configurations of interest. . . . .	29
<b>4</b>	<b>Test plan</b>	<b>31</b>
4.1	Particle test 1 . . . . .	31
4.1.1	Developed test . . . . .	32
4.2	Particle test 2 . . . . .	33
4.2.1	Developed test . . . . .	33
4.3	Particle test 3 . . . . .	35
4.3.1	Developed test . . . . .	35
4.3.2	Combinations and expected results . . . . .	36
<b>5</b>	<b>Results</b>	<b>39</b>
5.1	Particle test 1 . . . . .	39
5.1.1	Errors in CRAM. . . . .	39
5.1.2	Error modes. . . . .	40
5.1.3	Particle test 1 discussion. . . . .	40
5.2	Particle test 2 . . . . .	41
5.2.1	Total CRAM error rates. . . . .	41
5.2.2	Debug link & Benchmark. . . . .	41
5.2.3	Error mitigation and Energy . . . . .	43
5.2.4	dCache . . . . .	43
5.2.5	Error modes. . . . .	44
5.2.6	Particle test 2 discussion. . . . .	44

---

5.3	Particle test 3 . . . . .	45
5.3.1	Cache, I/O blocks & ROM memory . . . . .	45
5.3.2	Influence of operating system . . . . .	46
5.3.3	Error modes. . . . .	46
5.3.4	Cross sections of different configurations . . . . .	46
5.3.5	Particle test 3 discussion. . . . .	47
<b>6</b>	<b>Conclusions and recommendations</b>	<b>53</b>
6.1	Conclusions. . . . .	53
6.1.1	Conclusions. . . . .	53
6.1.2	Research questions . . . . .	53
6.2	Recommendations . . . . .	55

# List of Acronyms

<b>ALU</b> Arithmetic Logic Unit.	<b>LET</b> Linear Energy Transfer.
<b>ASIC</b> Application specific integrated circuit.	<b>LUT</b> Look Up Table.
<b>CLB</b> Configurable Logic Block.	<b>MBU</b> Multiple Bit Upset.
<b>COTS</b> Commercial Off-The-Shelf.	<b>MIG</b> Memory Interface Generator.
<b>dCache</b> data cache.	<b>MMU</b> Memory Management Unit.
<b>DD</b> Displacement Damage.	<b>OBC</b> Digital Signal Processor.
<b>DSP</b> Digital Signal Processor.	<b>PLIC</b> Platform Interrupt Controller.
<b>DSU</b> Debug Support Unit.	<b>PMP</b> Physical Memory Protection.
<b>DUT</b> Device Under Test.	<b>RISC</b> Reduced Instruction Set Architecture.
<b>EDAC</b> Error Detection And Correction.	<b>ROM</b> Read-only Memory.
<b>FEC</b> Functional Error Cross section.	<b>RTOS</b> Real-Time Operating System.
<b>FER</b> Functional Error Rate.	<b>SDC</b> Silent Data Corruption.
<b>FF</b> Flip-Flop.	<b>SDRAM</b> Synchronous Dynamic Random-Access Memory.
<b>FF</b> Fatal Failure.	<b>SEE</b> Single Event Effects.
<b>FI</b> Fault Injection.	<b>SEFI</b> Single Event Functional Interrupt.
<b>FPGA</b> Field Programmable Gate Array.	<b>SER</b> Soft Error Rate.
<b>FPR</b> Floating Point Register File.	<b>SET</b> Single Event Transient.
<b>FPU</b> Floating Point Unit.	<b>SEU</b> Single Event Upset.
<b>FT</b> Fault Tolerant.	<b>SF</b> Safe Failure.
<b>FTE</b> Fluence-To-Error.	<b>SP</b> Soft Processor.
<b>GCR</b> Galactic Cosmic Rays.	<b>SRAM</b> Static Random-Access Memory.
<b>GPR</b> General Purpose Register file.	<b>STAT</b> Status Registers.
<b>HTIF</b> Host Target Interface.	<b>TID</b> Total Ionising Dose.
<b>IC</b> Integrated Circuit.	<b>TMR</b> On-Board Computer.
<b>iCache</b> instruction cache.	<b>uC</b> Micro-controller.
<b>ISA</b> Instruction Set Architecture.	<b>UF</b> Unsafe Failure.



# Introduction and motivation

## 1.1. Motivation for research

Nominal operation of a spacecraft is ensured by numerous integrated circuits (IC). These ICs have similar functions when used in terrestrial applications, but with the downside of having to deal with the hostile environment in space. Due to this environment and the to be taken counteractive measures, electronics used in space usually lag behind electronics used for terrestrial applications in the performance, area usage, and power fields by three to ten years [3].

As a consequence of these circumstances the development of an IC for use in space is not straightforward. The main driving forces for manufacturing for space applications are constraints on reliability and availability, meaning the ability to perform its function in its intended way and the ability to perform this function without interruption, respectively [3].

In addition to reliability and availability, ICs in space have a large list of desired properties. Minimising resources (low power, high performance, small area) and space qualification being notable ones. These properties are usually each other's opposites, an IC that has space proven performance lags behind a state-of-the-art ICs' resource usage. Whereas, using a state-of-the-art non-space proven, termed Commercial Off The Shelf (COTS), component has the disadvantage of not being flight proven.

Preparing such COTS electronics for space operation involves the cost of the unit itself, testing, development of Fault-Tolerant (FT) mechanisms and validation of those mechanisms. As expected and investigated in [4], this process makes using a terrestrial system viable for space application much more expensive opposed to using already space-proven components. The only justification for the use of a COTS system in space is the improvement in performance, area and/or power usage. A trade-off always exists between time and money spent on extensive space testing of new(er) electronics, and the use of an already proven part which will have a compromised performance. The rise of CubeSats and SmallSats in the space market is causing a shift, in the direction of the former, in this trade-off [5].

Typical (ESA) satellites employ between 47 and 400 ICs, with the Inmarsat2 taking the crown employing roughly 1700 ICs [6]. These ICs are a mix of microcontrollers (uC), Application Specific Integrated Circuits (ASIC) and Field Programmable Gate Arrays (FPGA). All with its specific tasks and, therefore, specific requirements. For example, the On-Board Computer (OBC) performs different functions compared to the payload processor, therefore being subject to different requirements, where the operation of the former will be optimised for dependability, whereas the latter will benefit more from high performance.

The interest for complex ICs in space is increasing, and with that also the use of FPGA. This is further reinforced by the relative amount of FPGAs increasing compared to, for example, ASIC. FPGA usage in spacecraft has increased from about 100 units in 2004 to 250 units in 2014 [6]. On two satellites alone, the Sentinel-2 duo, 155 FPGA's are employed both for payload instruments and platform avionics [6].

Multiple different types of FPGA exist, the difference between these types is the type of memory the configuration bitstream is saved in. The DUT in this case is an SRAM-based FPGA, the volatile nature of SRAM leading to a complete loss of all configuration and user data in case of a power off. Another type of FPGA, the flash-based FPGA opposed to the SRAM-FPGA, does already have flight heritage. Examples include the GR732 and ROCKET-RTG4, both using the SPARC ISA.

There is an advantage that SRAM-based FPGAs pose over ASIC, this advantage being the ability of SRAM-FPGA to be reconfigured. This makes soft- and hardware updates possible after manufacturing of the component, whereas an ASIC only provides the possibility of software updates after the ASIC has been produced. FPGAs used in space until recent dates have not been of the SRAM type, but recent developments show a rise in these types of FPGA, with accompanying radiation tolerant versions.

FPGA configurability is ensured by the FPGA being made up of logic resources that do not have a set function, but can be made to perform different functions depending on the configuration. Re-configuration can be helpful during the development process, where design changes will not lead to manufacturing of an entirely new component. But also when the FPGA is already in orbit, (configuration) bug fixes can be performed from the ground station, and even repurposing of the FPGA is possible.

The hardware employed on a FPGA is often in the form of a Soft Processor (SP). The SP consists of similar parts as a normal processor, but usually has more configuration options. Soft-cores allow for flexibility in the design in case of changing requirements during the development process. The difference with more traditional hardcore processors is that the SP is described in a hardware description language which configures the logic resources available on the FPGA to form the processor. This makes for a SP that can be implemented on a wider range of devices, if not made device specific. Hard processors have their logic resources such as Look Up Tables (LUT) implemented within the silicon of the device itself and can thus not be altered after manufacturing.

Examples of SP used in space applications include the LEON3 and LEON4 cores by Cobham Gaisler. An important difference between processors is the instruction set architecture (ISA) they are based upon. The former are based on SPARC ISA and the latter two are based on a RISC ISA. Investigations on the different implementations of soft processors show a very wide range of soft processors employing different ISA. [1] It is shown that processors based on a RISC-V ISA have comparable performance to processors based on different ISAs and are, therefore, suitable candidates for space systems. Due to its openness and modularity the ISA has brought about numerous different processors with a wide variety of applications.

The SPARC ISA has been the most used ISA by the European Space Agency in recent years, but due to lost momentum of this ISA in terrestrial applications the way for a new ISA to be adopted by ESA is paved [7]. RISC-V is also based on the Reduced Instruction Set Architecture (RISC), and offers the same openness as SPARC, with added benefits of modularity, compact code and larger address size (64-bit and 128-bit availability) [8]. Contrary to SPARC, RISC-V shows momentum by wide adoption by academia and backing of big commercial companies, like google [1, 7]. As the source code is open-source, device/application specific improvements can be realised easier and quicker. Therefore, the open nature allows improvements in fault-tolerant and security fields to be application specific. The modularity also helps to make the designer to implement an architecture which best represents the application envisioned, by not needing overhead of unnecessary extensions.

A new RISC-V based SP just brought to market is the NOEL-V, developed by Cobham Gaisler, which also provides the LEON processor, based on SPARC ISA. This processor is important since, as reported in [3], "LEON2- and LEON3-based SoC are now the main workhorses for all the major manufacturers of On-Board Computer Systems in Europe." and that all OBCs used by ESA on new missions will be LEON based. If ESA wants to stay up with state-of-the-art performance, these processors will become obsolete and need rises for a more modern SP. Familiarity and experience with the developer will then come in handy for the NOEL-V processor to take over operations, as it will be seen that NOEL-V outperforms the other soft processors.

This thesis evaluates the NOEL-V soft processor implemented on a Xilinx Kintex FPGA. In order to characterise the performance of a NOEL-V soft processor in the harsh space radiation environment, the KCU105 development board is used as Device Under Test (DUT) and tested with a high energy proton beam. The KCU105 employs the Xilinx Kintex Ultrascale xcku040 which has the same silicon as the xcku060 [9]. This higher performance FPGA also exist in a space grade version, leading to comparable behaviour to a space grade processor. Numerous efforts have been taken to characterise the FPGA in the space environment using fault injection and particle test, but the performance of a RISC-V based processor has yet to be tested on a FPGA of this type.

Radiation tests will provide valuable information about the behaviour of the SP in the presence of the harsh space environment. To complete this research successfully, an understanding of the theoretical

influence of radiation on ICs, and specifically FPGA, is needed. A summary of the researched subjects is provided below.

## 1.2. Literature study and research motivation

The literature study initially aims at investigating the most impactful influences an ionising particle can have on an FPGA, this is addressed in subsection 1.2.1. Afterwards, existing radiation test guidelines are reviewed in subsection 1.2.2, followed by an investigation of the FPGA and SP used in Figure 1.2.3.

### 1.2.1. Electronics in space

The space environment is hostile, with a mixed field of different ionising particles. These cause instantaneous effects in the form of Single Event Effects (SEE), and causing effects that occur over time due to accumulation of physical defects in the form of total ionising dose (TID) and displacement damage (DD). The ionising radiation is caused by energetic protons, electrons and heavy ions.

Considering the effects that happen over time, TID and DD are very gradual and will progressively lead to diminishing device properties, like current consumption. This research will mainly focus on SEE. For the effects of TID and DD on this particular DUT, the reader is referred to other sources.

A SEE can be further divided into destructive and non-destructive SEE. Destructive events include the single event latch-up, single event burnout and single event gate rupture. Non-destructive effects are single event transients (SET), single event upset (SEU) and single event functional interrupt (SEFI). This work will mainly be focused on non-destructive errors, as those are much more common, especially for this particular FPGA [10], and these are usually dealt with technology and architectural level as opposed to at the architectural level [11].

#### The space radiation environment

Closest to Earth, where most satellites operate, the most abundant energetic particles are protons and electrons. These particles have been trapped by Earth's magnetic field, thereby threatening the lifespan of satellites but also shielding those same satellites from galactic cosmic rays (GCR) and solar influence.

Further out in space, at around 4 Earth radii, the amount of trapped particles starts to diminish, giving way to new dominant sources of radiation called galactic cosmic rays and solar effects. As the name suggests, GCR originate from the far away galaxy and are thus very energetic and constant. This in contrast to the solar radiation which is (generally) less constant and less energetic. GCR and solar radiation will be in the form of heavy ions, the former will have higher energy, be more constant but will also be less abundant.

There are also big differences in energy and abundance within trapped protons and electrons. This energy and abundance varies with altitude, longitude, latitude and time. An example distribution of protons with varying altitude is shown in Figure 1.1.

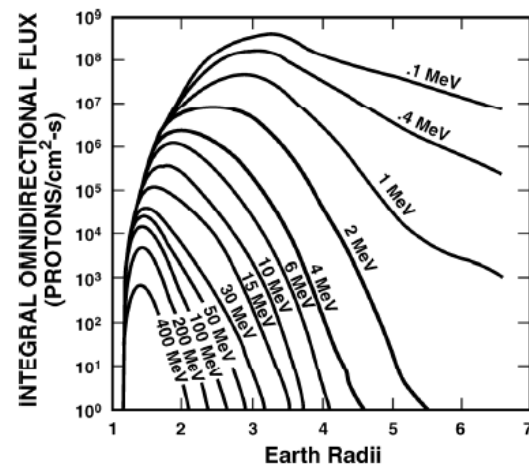


Figure 1.1: Change of proton energy concentrations with increasing distance from Earth [12].

#### Non-destructive errors

As this research will be mostly focused on non-destructive errors, it is important to get a good understanding of the mechanisms leading to these errors and the impact these errors can have.

##### Single event transient

One impact a non-destructive error can have is a SET. A SET occurs when an energetic ion traverses an electronic device, and as long as the ion has sufficient energy. Metrics such as location where the

particle hits and device characteristics deciding if the energy is sufficient.

Depending on the location, energy, surrounding layout, particle energy or Linear Energy Transfer (LET) and many other factors, the SET might lead to an error or vanish into the device. A SET thus not always leads to an unfavourable situation.

Figure 1.2 shows a SET occurring close in a sensitive device. In this case a particle hit occurred close to the sensitive area of a reverse biased N+/P diode.

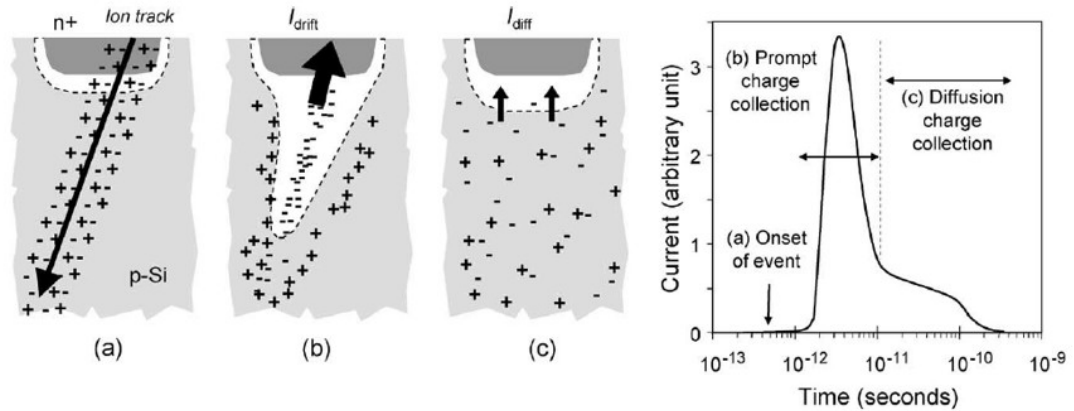


Figure 1.2: Current transient produced by a charged ion impinging a diode, phases and current over time graph [13].

In this figure, the three general phases of a SET happening in an electronic device are shown. Starting immediately after the ion hit, the onset of the event is shown in a. The ion leaves a track of electron hole pairs. In this example, for a reverse biased N+/P diode, the positively biased node attracts electrons and repels the positively biased holes.

The local electric field offers enough force to push the negatively charged electrons to the cathode and the positively charged holes to the anode. This movement is called drift, and happens quickly after the onset of the event. This effect is depicted in b.

After the initial fast charge collection due to the drift effect, the gradient force leads to another current. The local concentration gradient pushes the electrons towards the depletion region which leads to the so called diffusion charge collection, as depicted in c.

This example is for a singular, isolated node, but this is not the case in real world electronics. The effect this SET will have, if any, will be highly dependent on the surrounding circuitry. A depiction of a SET in combinatorial and sequential logic is shown in Figure 1.3. It is also possible, especially with decreasing transistor sizes, that multiple nodes have been struck by the same particle.

As can be seen in Figure 1.3 there are multiple ways that the SET can cause an error, but also multiple ways it can go unnoticed. Prerequisites of the SET to cause an effect are for example its strength, the existence of a valid path and arrival time. When the SET arrives on a clock edge, it can take an effect in the sequential or memory component. When it is out of sync with the clock edge it will not invoke an effect.

It logically follows from this that the probability the SET having an effect increases with the frequency of the clock. Furthermore, it means that the wider the SET (longer time of effect), the probability of coinciding with a clock edge increases.

A sufficiently strong SET can even cause a false falling and rising edge on the clock tree. This would lead to activation of sequential circuits when they should not be activated, possibly leading to invalid data in- and outputs. This, however, needs very high energy/LET particle strike as the clock tree usually has higher capac-

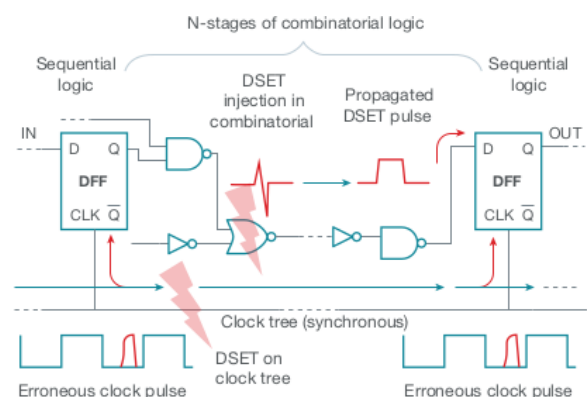


Figure 1.3: SET in a digital circuit [14].



itance [14]. Therefore, this effect will not occur frequently, but if it occurs the effect will be impactful.

### Single event upset

A SEU occurs when radiation damage occurs within a storage component. This storage component being, for example, Static Random Access Memory (SRAM), a latch or a flip-flop (FF). The result is a lasting error that could hinder proper operation of the device. This erroneous data can be functional or non-functional. In the former case leading to wrong operations down the line, or in the latter case having no effect at all. A good thing about an SEU is that the circuit is not damaged and the upset can be corrected. An example of a bit-flip in a SRAM cell is depicted in Figure 1.4.

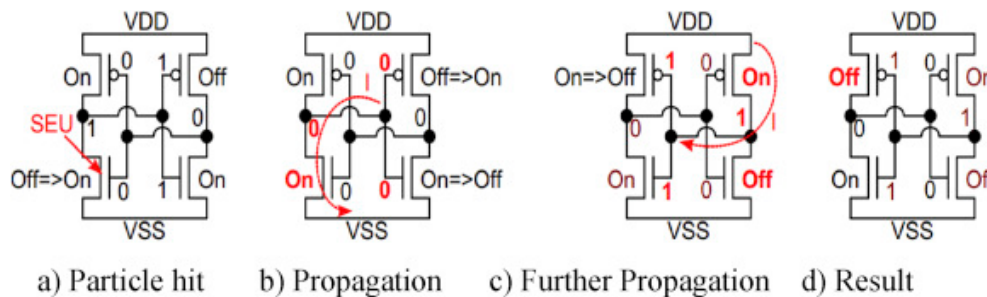


Figure 1.4: An SEU leading to a bit-flip in SRAM [15].

In this figure it can be seen that, for a typical SRAM cell, an ion hit can mimic a temporary turn-on of the M2 transistor. The current that will flow due to a short circuit is in the same direction as the current produced without an ion hit. This current will flow through n1 towards M3 and M4, whose states are reversed. If these reverse states last sufficiently long, feedback to the M2 and M1 transistors will reverse the states of these transistors.

Vulnerability is increased by increased switching speed of SRAM cells as this will heighten the likelihood of the reverse states lasting an adequate time. As is usually the case, smaller transistor sizes coming to market will decrease the likelihood of a particle hit, however, the simultaneously decreasing working current will lead to a larger spectrum of particles containing the required energy. Furthermore, transistors will be more closely packaged increasing the likelihood of multiple bit-flips by the same particle. In a similar manner the vulnerability of FF and latches increases with decreasing node sizes.

Within the SRAM cell itself, regions of increased vulnerability can be identified. For the side storing the "1" state such a vulnerable area is the collection of electrons by the drain of the N1, and for the side storing the "0" state, it is the collection of holes by the drain of P2. Although, this type of detail is not required for the research at hand, it would be interesting to know when trying to make a RHBD SRAM cell.

In addition to an ionising particle causing one bit-flip, it can also induce multiple bit-flips resulting in a Multibit Upset (MBU). This can be caused by one particle hitting multiple sensitive regions or by indirect ionisation. Indirect ionisation is the ionisation by particles refracted from other particle collisions [16]. If a proton strikes a neutron in the silicon, this silicon will be expelled at high energy and be able to strike other nodes. MBU's are more impactful to the operation of the device as they are much harder to identify and often impossible to correct.

As observed before, MBU's are mostly caused by higher energy or higher LET events [14]. These events are much rarer, resulting in a MBU rate of approximately 5-15% of the SEU rate for operational SRAM cells.

The sequential logic devices making operation of the FPGA possible, such as latches and flip-flops, can be used as storage elements. These devices are usually used to buffer data between combinatorial elements. A SEU or MBU in such a storage device can have a big impact on the operation of the combinatorial elements as they will be fed wrong data.

Looking at the devices, a latch also makes use of a feedback loop as was the case for a SRAM cell, bringing about a SEU sensitivity close to that of a SRAM cell. Similar properties determine the sensitivity of a latch, like speed of operation and working current.

Flip-Flops, however, do use a different architecture, as shown on the right in Figure 1.5. This leads to a different sensitivity to SEU at different nodes in the FF. Further leading to fewer sensitive nodes and, therefore, more robustness to SEU.

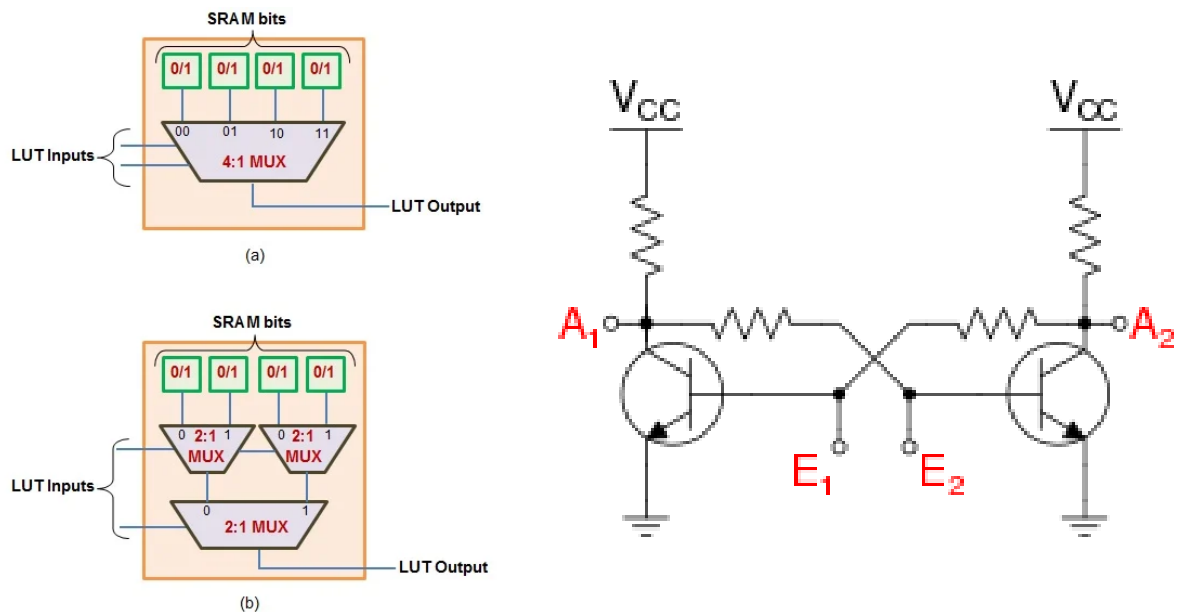


Figure 1.5: Schematic of a look up table (left) and schematic of a Flip-flop (right) <sup>1</sup>.

### Single event functional interrupt

Not so much a stand alone event, a SEFI can be seen as a particularly bad SEU. A SEU can occur in all elements storing a value. When this upset bit is of a particular importance in, for example, the control logic for decoding operations numerous upsets result with large blocks of bits being computed faulty. This is visually depicted in Figure 1.6.

Vulnerability of a device to SEFI thus depends on the vulnerability of that device to withstand SEU in the "important" bits. Meaning speed and transistor sizes are also of importance to the vulnerability to SEFI, as it was for the vulnerability to SEU. Resolving a SEFI usually requires a reset or power cycle.

Another major drawback of a SEFI over the occurrence of a SEU is that it usually leads to errors that are undetectable and unrecoverable by employed mitigation techniques [17], making SEFI an important contribution to failures in otherwise well protected systems.

### Destructive errors

As noted before, the destructive errors are deemed of lower priority during this research. However, the effects have to be known and studied to avoid taking the risk of destroying the electronics. A small safety assessment of the destructive errors is discussed below.

It is known that the biggest safety concern for an FPGA is a latch up. However, for this particular FPGA it has been proven that the chance of an SEL during proton irradiation is negligible, with zero

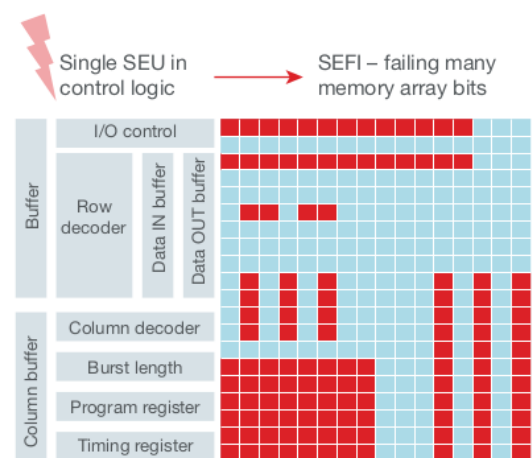


Figure 1.6: Single event functional interrupt [14].

<sup>1</sup>LUT: <https://www.allaboutcircuits.com/technical-articles/purpose-and-internal-functionality-of-fpga-look-up-tables/m>  
FF: [URL:https://en.wikipedia.org/wiki/Flip-flop\\_\(electronics\)](https://en.wikipedia.org/wiki/Flip-flop_(electronics))

occurring latch ups reported on multiple occasions [10, 18].

Reported high current events in 7-series and Ultrascale devices have been investigated [19]. The conclusion of this research is that the high-current events are due to an employed scrubbing technique.

As no scrubbing technique is employed in the configuration loaded onto the FPGA (GRSCRUB and NOEL-VFT are commercial products for which the license is not available), it is assumed that no latch-ups will occur and no additional measures have to be taken.

For a more in-depth review of destructive errors in electronics the reader is referred to [14] and [20].

### 1.2.2. Proton radiation testing guidelines

A FPGA can be tested in multiple different ways. For these tests, due to availability, it was chosen to employ a proton particle test as opposed to fault injection (FI) or neutron particle tests. In addition to the availability of a proton test facility, particle testing is deemed favourable over fault injection due to accuracy. It has been reported that the failure rate observed during particle irradiation is larger than the estimated rate using FI [21].

#### Guidelines

Test guideline documents are available from different institutions, like JEDEC<sup>2</sup>. [22] and [23] cover test procedures for the measurement of SEE from proton and heavy-ion irradiation respectively. Furthermore, procedures for terrestrial cosmic ray induced destructive effects, alpha particles and cosmic ray induced soft error testing are covered by JEDEC.

Other guidelines concerning similar topics are developed by ASTM<sup>3</sup> and by European institutes like the European Space Components coordination (ESCC). All these documents can be used to produce a guideline for conducting a radiation experiment. Although the guidelines are made by renowned institutes, the ever developing technology of semi-conductors makes for unexperienced effects during times of writing them. The test conductor must thus always be aware of shortcomings in these guidelines.

#### Proton testing

Proton penetration will be high enough that no material needs to be removed, as would be the case for heavy-ion testing. It is common to start with the highest energy protons, as this will limit the degradation due to TID and DD [22]. Testing is usually performed with a flux between  $1 \cdot 10^5 - 1 \cdot 10^9$   $p/cm^2 \cdot s$ . The time of the test, or the corresponding fluence is then chosen to obtain the desired results.

Beam fluence, energy and uniformity will have to be obtained from the facility. This data is very important during the analysis of the data. Usual proton accelerators have a small offset in their output energy, up to 10% can be expected [22].

These three properties of the beam are used to completely characterise it. The fluence is the amount of particles expelled by the collimator per unit area, measured in  $p/cm^2$ . The particles of this beam have a certain energy, which is directly correlated with its velocity in the energy equation. Higher energy protons will thus have a higher speed, this is known to have an effect on the influence the proton has on an IC, both causing different effects (more direct ionisation) and more effects (up until a certain threshold).

The beam will be directed at the DUT, to be most representative of space the area of irradiation should have an equal distribution of particles over the entire area. Unfortunately, perfect uniformity is impossible to reach. A small non-uniformity always be present but by minimising it one can assume that it does not have a significant impact.

### 1.2.3. Susceptibility of FPGA

In order to assess the susceptibility of a FPGA, a quick overview of what a FPGA is made up of is investigated.

---

<sup>2</sup>URL:<https://en.wikipedia.org/wiki/JEDEC>

<sup>3</sup>URL:[https://en.wikipedia.org/wiki/ASTM\\_International](https://en.wikipedia.org/wiki/ASTM_International)

## Field Programmable Gate Array

The basic lay-out of a SRAM-based FPGA is depicted in Figure 1.7. The FPGA itself consist of Configurable Logic Blocks (CLB), configurable switch matrices and IO blocks. The layout of a CLB is device specific, but for Xilinx devices usually looks similar to the CLB in the figure. The CLB can have a multitude of functions depending on the configuration programmed into it. The Look Up Table (LUT) for example can have different outputs for the same inputs when programmed to fulfil a different function.

The switch matrix is also dependent on the configuration programmed, the user of the FPGA can decide which CLBs are connected to each other.

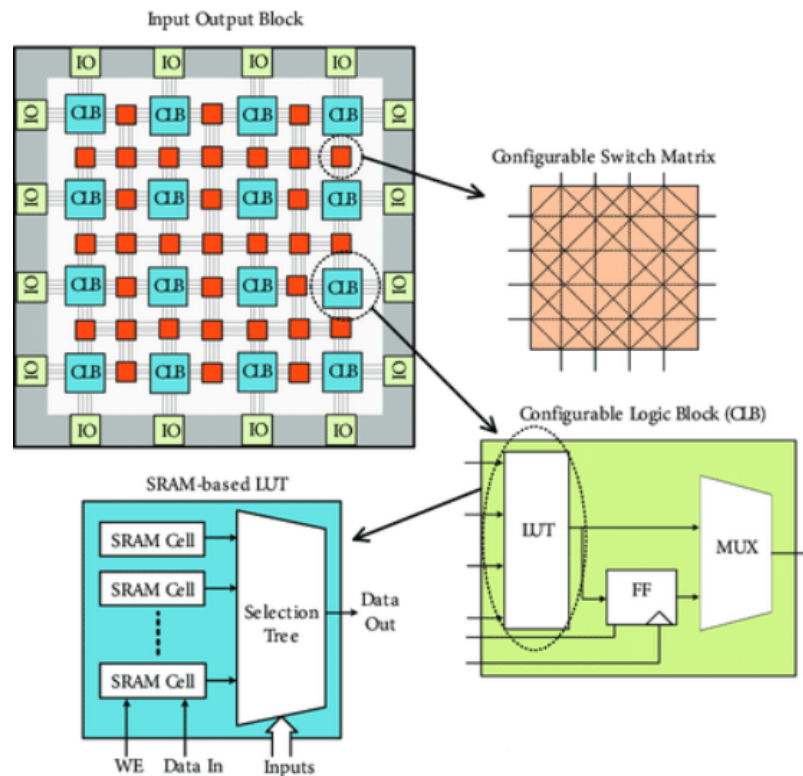


Figure 1.7: General overview of the constituents of an FPGA [24].

### Radiation effects

As the research will be performed on an SRAM FPGA, which stores all of its configuration bits in SEU-vulnerable SRAM. The susceptibility of a SRAM FPGA will be leveraged. SEU in SRAM-based FPGAs can be grouped into three categories [25]:

1. Configuration upsets (Upset in CRAM).
2. User logic upsets.
3. Architectural upsets.

All of these find their origin in the same manner as discussed in subsection 1.2.1, but their effects can be vastly different, although for all the case of an error not disrupting function whatsoever is present.

In the case of the used FPGA, the Kintex Ultrascale xcku040, the bitstream has a size of 128,055,264 bits. If an upset in one of these bits will affect the configuration depends on the specific designs utilisation of the device resources.

Such a configuration upset, in this case occurring in a CLB, is shown in Figure 1.8. Similar upsets in the switch matrix could lead to connections between CLBs being broken or wrong connections being made.

User logic is not available for readback in the bitstream, these bits will also be included in the configuration bitstream but will change during use. These bits describe logic elements like block RAM,

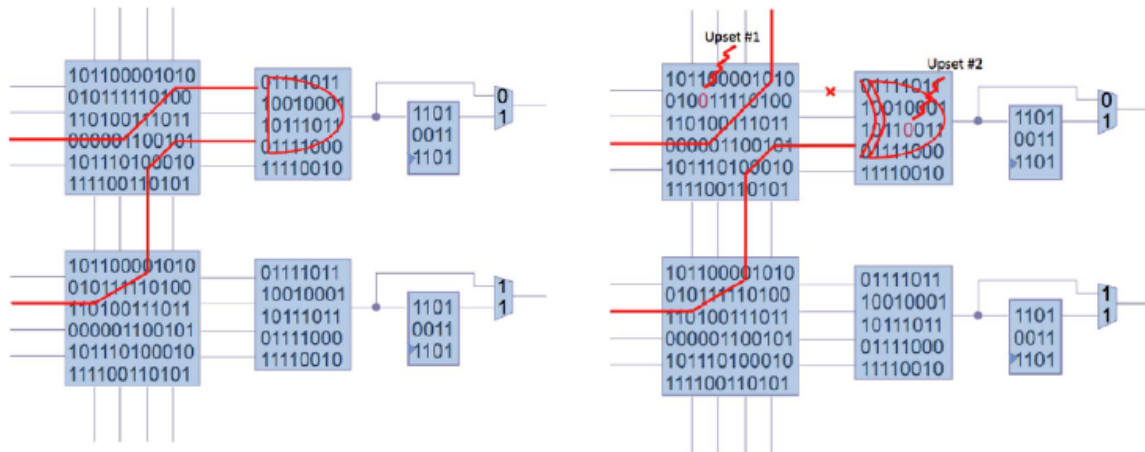


Figure 1.8: Occurrence of an SEU changing logic in CLB, with the correct bits encoding an AND gate (left), which is broken by an SUE (right) [26].

Flip-Flops and I/O block Flip-Flops. The constant changing of these bits makes it harder to detect when an upset occurs. For proper upset detection all correct bit states should be known at all times during operation, which is infeasible, leading to a different strategy as opposed to configuration RAM upset detection.

Due to this inability of detection of upsets, upsets can only be mitigated during operations by employing redundancy, such as triple modular redundancy (TMR), implemented by a user in the FPGA logic design. Observability is ensured by implementing even more overhead into the design.

An upset occurring in control elements of the FPGA will be deemed a architectural upset. This could be the configuration control circuit, which when upset, would write completely different bits as planned. SEUs in such control elements will need to be observed and linked to the control element function.

### Susceptibility of Xilinx Kintex Ultrascale

The Xilinx Ultrascale family makes use of 20 nm transistor technology, as opposed to for instance 28 nm technology used the older 7-series. Transistor size shrinking is beneficial from a performance, and energy consumption standpoint. However, it can be detrimental to its performance in a radiation environment.

Although the smaller sizes will have a smaller area to be hit by particles, the density of transistors and the lower charge required to have an effect has an adverse effect on the SEE susceptibility of transistors, leading to an overall increased susceptibility with shrinking transistor sizes.

Furthermore, radiation effects are not limited to space applications. At the surface of Earth general radiation effects are also present, although much rarer. Being of general interest to every user of the FPGA, the manufacturer already tries to reduce radiation effects.

In [28], measures taken in the design of the FPGA are discussed. At the native device level, circuit design and layout techniques are employed. These consist of interleaving RAM cells and built in error detection and correction logic.

To increase mitigation even more, use can be made of the Soft Error Mitigation IP core to enable better error detection and correction measures [29].

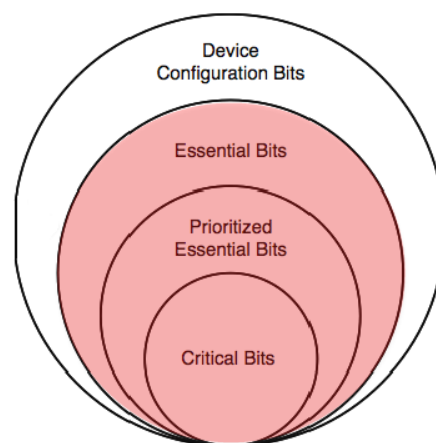


Figure 1.9: Naming scheme used to discern the bits in CRAM into different importance [27].

Upsets can occur in all bits of the CRAM, however only a number of the bits, if upset, will have an effect on the configuration of the system. These bits are the ones that actually implement the design in the FPGA fabric, and are deemed essential. If these bits also affect the function of the design it is deemed critical. A visual representation of this naming scheme is given in Figure 1.9. [30]

According to experiments performed by Xilinx, the uptime of a Kintex Ultrascale at Earth's surface in New York is at least 99.9988%, improving with additional measures as shown in Figure 1.10.

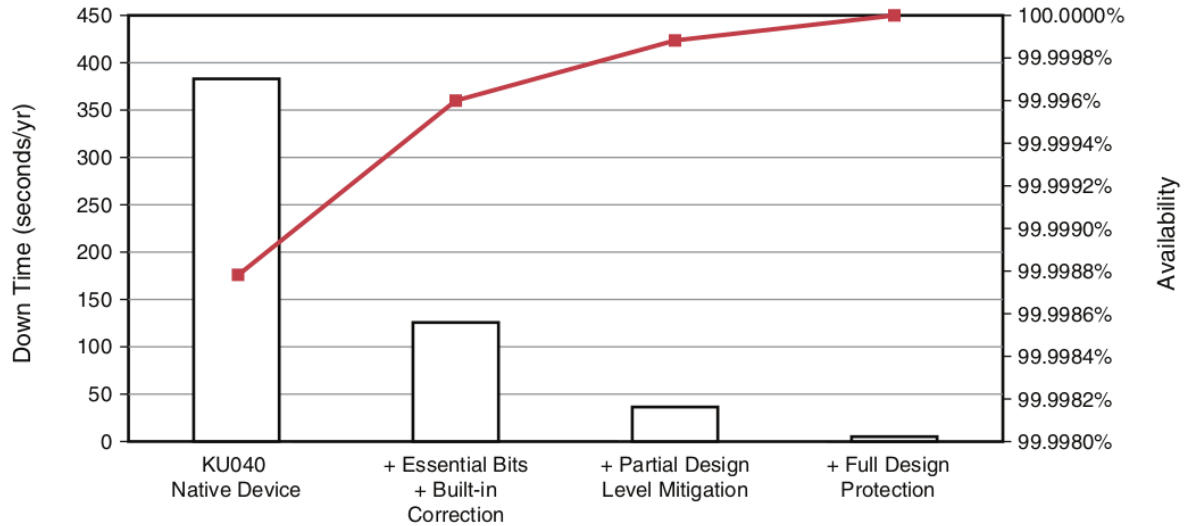


Figure 1.10: Xcku device availability employing different mitigation levels [28].

This is not an indication of the uptime in space operation of the same FPGA, but it does show that the FPGA, in this case the xcku040, inherently already resists SEEs. The most striking evidence of the measures taken effectiveness is given by Xilinx and can be seen in Figure 1.11 [28].

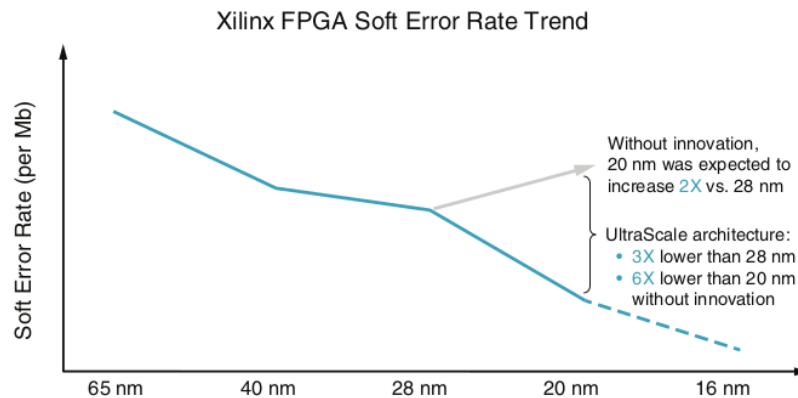


Figure 1.11: Xilinx FPGA Soft Error Rates vs. Process Technology Node [28].

The figure shows that the Soft Error Rate (SER) does not increase, although this would be the expected trend with decreasing transistor size. These findings of Xilinx are further proven by other researchers in [31].

### Susceptibility of Soft processors

The susceptibility of soft processors is determined by the ability of the soft processor to handle user logic upsets and configuration upsets. Like CRAM upsets, a user logic upset does not always lead to notable change in operation of the processor.

Soft processors have been subject of radiation testing to get a feel of their viability of use in space. Example of soft processors evaluated are the Rocket [32] and TAIGA [33]. The fact that soft processors are useful for space applications has already been proven by the use of LEON soft processors by ESA.

Being from the same manufacturer as all currently used system-on-chips by ESA, the NOEL-V by Cobham Gaisler is definitely an interesting processor to analyse. Provided with the processor are board support packages to make software development using kernels possible, and because it is based on the RISC-V ISA all compilers and kernels based on RISC-V can be used. Prebuilt toolchains for RTEMS and Linux are available for free to aid in software development.

In terms of performance the NOEL-V is promising. Comparing the processor to other general purpose processors using [1], and using the reported CoreMark score of  $4.03 \text{ CM}/\text{MHz}$ , one can see that processor compares to the best performing processors like the Berkeley Out-Of Order Machine (BOOM). But as [1] suggests, the better comparison is to compare it to processors employing the same level of instruction level parallelism. Here it outperforms the other dual issue, in-order processors by a factor of 1.33x (on average).

## 1.3. Research framework

Knowing the absence of radiation characterisation of the NOEL-V processor and its potential for use in space applications, this research makes an effort to characterise radiation tolerance of NOEL-V. As is the case for all ICs used in space, some form of radiation protection will be needed, which will always be at a cost. In order to limit the cost of making the processor space ready, this report aims to find areas of the processor which, when addressed, lead to the highest gain at the lowest cost.

To achieve this goal, a research question with associated sub-questions is formulated. The research question is formulated in such a way that an answer will advance knowledge about the behaviour of the NOEL-V processor under radiation, and in which way this behaviour can be altered. At first these will be discussed in this section, whereafter the methodology proposed to achieve the objective will be addressed in section 1.4.

### 1.3.1. Research question(s)

The main research question of this thesis is:

**Is a (modified) version of the NOEL-V soft processor suited for space applications and what changes would lead to the highest improvement of the radiation susceptibility at the lowest cost?**

With this as the main question, a few sub-questions can be formulated:

1. How to design a radiation test such that it most ideally mimics in-space operation?
  - (a) What software can be used to mimic in-space operation?
  - (b) What particles are best used?
  - (c) What duration should the test take?
2. How to extract information about radiation effect from the device under test?
  - (a) From CDRAM
  - (b) From cache memory
  - (c) From I/O blocks
  - (d) From ROM
  - (e) From Floating Point Unit (FPU)
3. What parts of the NOEL-V soft processor are most vulnerable to ionising proton radiation?
  - (a) How to best test per part susceptibility?
  - (b) What is the cost of applying mitigation to a specific component?
4. Which configuration of the NOEL-V processor is best suited for specific tasks?

### 1.3.2. Objective

The research objective of this research project is to get a general overview of the susceptibility to radiation of the the NOEL-V soft processor and its individual parts, by means of using the modularity of the processor and software layers to characterise different parts of the processor.

This research will be realised in the following global steps:

1. Obtain hardware and develop software.
2. Soft- and hardware verification.
3. Irradiation testing with high energy proton beam in a specialised facility.
4. Post-processing of the data and validation of the models.

These steps aim to achieve different sub-goals along the way:

- The first sub-goal is to design a radiation test that best mimics space but stays elementary by looking into the advantages/disadvantages of different particle tests at the available facilities.
- The second sub-goal is to identify which components in the FPGA are most important to investigate by reading relevant literature and comparing this to the device under test.
- A third sub-goal is to make extraction of radiation effect possible from the FPGA by developing an IP that is able to point out SEE without causing too much overhead.
- The last sub-goal will be to identify highly susceptible parts of the NOEL-V processor and validating test results by comparing the found data with literature.

## 1.4. Methodology

In order to characterise the NOEL-V soft processor, three procedures are identified:

- Synthesise the desired configuration, and exhaustively test this configuration to get a grasp of how well it will perform when employed in the hostile space environment.
- When the processor is synthesised, the user will have access to all knowledge of all parts of the FPGA used, such as the amount of Flip-Flops and block RAM. Using the susceptibility of these individual components the susceptibility of the processor as a whole can be estimated.
- Using the modularity of the NOEL-V processor, the researcher can test multiple configurations in combination with the use of software identifying bottlenecks.

A drawback of the first approach is it taking numerous tests to discover all possible error modes. These test results will not be applicable to other configurations, despite these configurations only differing slightly. A radiation hardness characterisation would still be needed when employing the NOEL-V in a different configuration.

The second option also requires a lot of testing, but a lot of data is already known, like the susceptibility of flip-flops in the Kintex Ultrascale [18, 34]. However, to tie this all together into the susceptibility of the processor as a whole, a lot of additional effects will have to be taken into account. Mechanisms such as fault masking will be configuration-specific and cannot be easily estimated without proper radiation testing of the complete configuration.

The third option will be a compromise in the completeness of the radiation assurance, as the characterisation of a single configuration will not be as in-depth as for the first option. However, the test method will make data applicable to more configurations.

The third option will be chosen in order to have a more generalised overview of the radiation effects on the NOEL-V processor, which can also be transferred to slightly modified configurations. In addition, by examining radiation hardness of different components constituting to the processor, one can identify which components benefit most from radiation hardness, at the lowest cost.

Knowledge of the different constituents of a microprocessor is needed to complete the objective(sub-goal two), as not all parts will be equally interesting to characterise. A closer look is taken into the functionalities of the constituents and how they could prove to be a hindrance when employed in a radiation environment.



### 1.4.1. The microprocessor

A microprocessor is made up of several components constituting to making it functional. These components contribute together to the radiation hardness of the processor itself and are thus important to investigate.

#### Processor pipeline & memories

The main block is the main processor pipeline, for NOEL-V this is a 7-stage pipeline. This pipeline enables the processor to do all necessary computations by having stages that fetch, decode and execute instructions. The pipeline itself includes Arithmetic Logic Units (ALU), multipliers and many other parts that can all be affected by radiation.

Data and instructions that the pipeline fetches are fetched from memories. This memory can be divided into many levels, with smaller faster memories combined with slower, bigger memories normally present in a processor. In the case of the NOEL-V, depending on configuration, the processor will use L1cache, optionally L2cache and DDR4 memory. Data and instructions stored in the caches will compromise the operation of the pipeline and are thus very important to characterise. A schematic representation of the memory structure used in NOEL-V is shown in Figure 1.12. In [11] it is shown that caches are the most radiation vulnerable part in a processor, making the caches important parts to investigate.

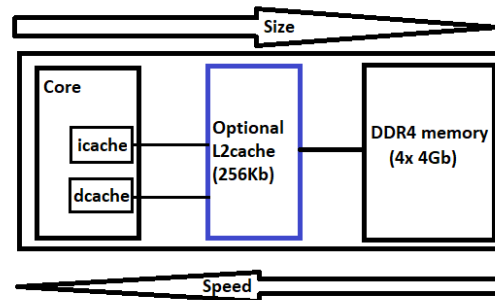


Figure 1.12: Schematic of the memory employed on the NOEL-V processor. 4 DDR4 memories of 4 Gb each are available [35].

The L1cache and L2cache used in the NOEL-V uses write-through policy, this is only one of many specifics about the cache that there is always a copy of the data in another memory. An upset in the data could then either lead to a cache miss, resulting in the fetch of the data from a different memory. This could take up to 150 clock cycles. Another outcome of a fault in the cache could be a fetch of wrong data, possibly leading to false computation.

Vulnerability due to hierarchy has been investigated in [36]. It is found that the dCache vulnerability increases with size, but this size increase also results in a lower vulnerability of the L2cache. The L2cache is found to be the most vulnerable cache level with the iCache being the least vulnerable.

The DDR4 memory is not located on the FPGA, but in close proximity of it on the development board. To enable data transfer, the processor must have some kind of memory interface. This memory interface is of importance to the processor as instructions and data that cannot be stored in the caches will have to be loaded from the DDR4 memory through the memory interface and IO ports.

#### Input Output blocks

Any interaction with external devices, like fetching data from the DDR4 memory discussed before, will make use of I/O blocks. As many as 520 I/O pins are present on the Kintex Ultrascale 040 FPGA which are all susceptible to radiation [9].

#### Configuration RAM

The bitstream describing the FPGA is stored in the CRAM. Upsets here can lead to a faulty design programmed into the FPGA and, therefore, wrong or no operation of the processor. However, not all bits stored in the bitstream are used to implement the design. In fact, only a small percentage of the available CRAM is actually essential. Research has shown that of the Xilinx Ultrascale, depending on resource utilisation, only up to 35% of the configuration bits is essential. Of these, only a subset (~10%) will be critical. This was found in [32], where fault injection is used to flip every essential bit individually. Only 10% of the upsets in essential bits actually led to a hang in FPGA operation.

#### Floating Point Unit

In addition to the integer pipeline, some configurations offer a dedicated FPU. This FPU will take over some of the operations otherwise being processed by the integer unit, making it of importance

to radiation effects. The FPU is capable of handling single, or double precision computations. The FPU is available for the configurations with the *F* (single precision) and *D* (double precision) RISC-V extensions.

A FPU failure could lead to false computations and thus poor operation of the processor, or could lead to a hang of the entire system making no computations possible at all.

Fault injection in [37] shows that applications are less susceptible to upsets in floating point register files (FPR) as opposed to general register files (GPR). The RISC-V standard FPU does include separate register files for the different pipelines.

### Read Only Memory

The fabric of an FPGA embedded memory elements can be configured to be multiple types of memory. If configured as Read Only Memory (ROM) it is termed programmable ROM, as it can be programmed after manufacture.

On an FPGA the ROM can be instantiated by using a RAM bank with the write-enable disabled by hard-wiring the write-enable. Upsets in the ROM thus act like upsets in the RAM. The ROM will also be volatile and will have to be programmed every time the FPGA is powered. Upsets can be of higher severity due to the data in ROM being of higher importance (for example being the boot code).

### Other components

Together with the already discussed components, numerous other components are present in a processor. Components like Memory Management Unit (MMU), Physical Memory Protection (PMP), Platform Interrupt Controller (PLIC) and the bus architecture play a big role in correct operation of the processor, but are not all necessary parts.

The MMU performs virtual memory management by translating virtual memory addresses to physical memory addresses. In doing this, the MMU handles memory protection, cache control, and bus arbitration. It thus controls which data and instructions are stored in the caches.

The PMP allows specification of physical memory locations and controls the memory access permissions.

The processor employs two busses, and AHB and an APB bus, both with a dedicated controller. The AHB bus controls the memory system and the external peripherals connected to the core. The APB bus controls the internal peripherals.

If present on the NOEL-V processor, the MMU, PLIC and PMP are all RISC-V standard.

## 1.4.2. Radiation testing

In order to (partially) fulfil the first and third sub-goals effective particle tests have been developed.

### Logistics

The test setup used during radiation testing will be made using literature on already conducted tests on the KCU105 development board. Conducted tests show a test setup where a secondary KCU board is used for data extraction [18, 34]. This is deemed unnecessary as a PC in the radiation room can be used in the facility and no second development board is available to the researcher.

Data acquisition to the safe room where the researchers are observing is realised using LAN or wifi connection provided by HollandPTC and the entire setup is verified to work prior to the test.

General procedures for the test are based on literature of proton tests on KCU105 [18, 34]. An in-depth lay-out of the test setup, procedures and the facility is given in chapter 2.

### Desired data

In order to obtain the knowledge needed to answer the research question, the needed data needs to be identified.

For CRAM, this data are the error cross sections, and readback data to trace the location of the upset. The cross section( $\theta$ ) is defined as the possibility of a particle causing a bit-flip. It is calculated using Equation 1.1.

$$\theta = \frac{\#oferrors}{fluence \cdot \#ofbits} \quad (1.1)$$

For the user logic the desired data is more extensive. As was chosen to characterise using multiple configurations, and use the modularity of the NOEL-V SP to obtain knowledge about parts of the processor. The user logic will upset multiple times, but not every upset is impactful. An upset that is impactful will cause an error, if such an error occurs, it is deemed functional as the processor can not continue proper operation. The different error types are classified later in this section.

The types of data are then the Functional Error Cross section (FEC) and Fluence-To-Error (FTE) of the different configurations, where FEC is defined as  $1/FTE$ . In addition, using techniques from literature the upset rates of a component in orbit can be calculated, these will be termed SER and Functional Error Rate (FER), for the likelihood of an upset in CRAM memory and likelihood of a functional error respectively.

Investigations will be done to decide whether software or alternative methods can be used to determine the susceptibility of identified important parts of the SP. This specific data on processor parts can be used in conjunction with the overall data about the configurations.

By calculating the FEC from the FTE, the researcher assumes that only one functional error occurs at one time. Furthermore, only observed failures will be calculated.

For the earlier discussed CRAM cross-sections, the cause is always due to a CRAM upset. The latter FER of different configurations and specific part error rates can be due to both CRAM and user logic upsets. Any (relative) difference in CRAM cross-section and FER between configurations can then indicate an increase in user logic errors.

In addition, the influence of impacting mechanisms will need to be studied. Fault masking can be done by existing error mitigation measures like the SEM IP core.

### Data extraction

The data obtained will be extracted using measures made available by the manufacturers of the hardware and software. Cobham Gaisler provides the grmon3 debugger which has numerous commands available for data extraction purposes. The commands used will be described in detail in chapter 4, and in addition to these commands, the debugger provides the opportunity to print statements and provides handling of running and stopping programs.

Xilinx, the manufacturer of the development board, provides their own ways of data extraction from the DUT, like verification and readback procedures. These latter forms of data extraction mainly provide data on configuration upsets, whereas the former grmon commands will provide data on user logic and architectural upsets.

The Xilinx verification process checks if the written bitstream has changed using a mask file and a golden copy of the bitstream. The mask file is used to determine the essential bits. The output from the verification thus already takes into account the importance of some bits over others, but it does not mean that every bit flip reported using verification will lead to a hang in FPGA operation. Vivado Readback additionally uses a logic location file to be able to locate a possible upset in the logic.

### Mimicking in space operations

During the test the soft processor will not be idle. To mimic in space operations as best as possible benchmarks will be used running on the processor. Different benchmarks stress different parts of the processor but no single one can be said to be the most representative as there are numerous use cases in space. The used benchmarks will be explored in more detail in chapter 4.

Real life systems perform multiple tasks and are therefore usually running a Real-Time Operating System (RTOS). Using such a RTOS poses significant advantages by employing a scheduler and synchronisation mechanisms at the drawback of added complexity and thus harder test and debug operations. Running the benchmarks without a RTOS, termed bare-metal, is the most deterministic due to complete knowledge of the code being executed<sup>4</sup>. As noted, toolchains for RTEMS, a type of RTOS, are provided with the NOEL-V processor and can thus be used. This RTEMS code extensive and used as a black box by the researcher, therefore leading to a less deterministic analysis.

Ensuring that the test best mimics in-space operations, the effects of the employed debug link and run software will need to be investigated. It is chosen to employ both bare-metal and RTOS benchmarks, thereby finding an optimum in determinism of the test and representativeness of space. Furthermore, as in space not just one proton energy is present, the influence of proton energy will also be subject of testing.

<sup>4</sup>URL:<https://www.embeddedrelated.com/thread/5762/rtos-vs-bare-metal>

### 1.4.3. Data handling

All data obtained from the testing will be of a specified form and can, therefore, be handled by scripts. Together with observations of the researcher during the test, this qualitative data will be used to provide an answer to the research questions. In order to manage the data properly, a classification of failures should be adhered to.

#### Error classification

A number of tests will be performed and many distinct error modes will likely be found. To distinguish these error modes, a grouping and naming is made, dividing the failures into Silent Data Corruption (SDC), Safe Failure (SF), Unsafe Failure (UF) and Fatal Failure (FF). These error modes will be grouped under the name Functional Errors to make sure no confusion arises with soft/hard errors.

SDC will occur when an upset is inadequate enough that the processor cannot complete its tasks properly, like computing a false value. But this improper performance does not lead to the processor aborting. When the processor safely exits during operation as a result of a single event, this means that the software has found a mismatch. Therefore, this type of malfunction will be deemed a hang. If the system is not able to safely abort operation by itself, an unexpected termination has occurred. The debug links and outputs from grmon are important in distinguishing between a hang and UT. This debug link can also get corrupted leading to the inability of the researcher to extract information after an error. A error mode representing this case is deemed a fatal UT.

#### Validation

Error rates in CRAM will be comparable to values reported for other kcu105 development boards in proton irradiation. Found values can thus be verified against findings in [34]. Since it was noted in [2] that a strong correlation of the occurrences of the different error modes is present between different implementations of processors using the RISC-V ISA, verification and validation of the processor can be validated using other RISC-V based processors.

## 1.5. Thesis outline

In this chapter, the reader has been taken through an initial literature review to gain knowledge about the effects of radiation on electronics. Using this literature review, the problem statement, objectives and the methodology to achieve the set objectives were revealed, in section 1.3 and section 1.4.

The next chapter, chapter 2, will continue with the setup of the test. This will be followed by an in depth review of the NOEL-V processor in chapter 3 which will be of great use in processing obtained data. The different tests and the ways in which they will be beneficial to the conducted research will be discussed in chapter 4.

Results of these tests will thereafter be shown in chapter 5. Lastly, conclusions are drawn based on the obtained data and the research questions will be answered, together with a discussion on the limitations of this work and recommendations for further research, these are shown in chapter 6.



# 2

## Test setup

In this chapter the test setup is explained. The test setup will make sure the FPGA is fully irradiated without much interference on the development board. Figures are included to give a visual indication of the test setup. This setup is validated to perform its purpose before the first facility test.

During particle tests no anomalies were found in the setup, the setup was thus used unchanged for subsequent tests.

### 2.1. Facility

The proton tests are conducted at HollandPTC<sup>1</sup> in Delft. The facility offers a range of fluxes available, and provides exact values for the fluxes used after the test.

Furthermore, the facility provides the tester with multiple cameras pointed at the DUT to observe in real time.

The facility itself consists of multiple rooms, of which three are dedicated to patient treatment. These can be used for human research outside of treatment hours. Electronics and other research objects that do not need the equipment used for patients can be tested in a different radiation room, where no patient treatment is conducted.

The dedicated R&D bunker is equipped with a horizontal and stationary beamline. This beam can only be used outside of treatment hours for approximately 20 hours per week. It can only be used outside treatment hours since the proton beam is generated by a cyclotron connected to all four radiation rooms. This beam is then redirected by magnets to one of the different rooms in which the proton beam is requested.

The researcher is accompanied by an employee from the facility. There is control over the energy and the intensity of the beam. Reported value is the dose of protons. Using this dose, the flux can be calculated using the following formula:

$$Flux = \frac{dose * \rho}{S * c}$$

Where:

$\rho$  = density of the air in the radiation room =  $0.001205 \text{ g/cm}^3$

$S$  = Mass stopping power of air =  $2.92 \cdot 10^{-3} \text{ MeV/cm}$

$c$  = constant =  $1.60 \cdot 10^{-10}$

---

<sup>1</sup><https://www.hollandptc.nl/>

## 2.2. Test setup

The FPGA is mounted in the beam path by the facility operators. A number of remarks about the test setup are given below.

- The FPGA will be mounted on the KCU105 evaluation kit. This evaluation kit provides a development environment, including clock generation and DDR4 memory.
- As the development board also provides cooling for the FPGA, a fan is mounted on top of the FPGA. The influence of this fan on the proton beam is unknown and would have to be simulated. Therefore, it has been chosen to irradiate the backside of the FPGA. This is possible as the FPGA used is interconnected using the flip-chip method. Taking off the fan was no viable option as the temperature of the FPGA would become excessive for the maximum rated temperature.
- The FPGA is irradiated at a normal angle of incidence with the room kept at room temperature.
- Collimators are used to bring the beam area back to the size of the FPGA, an area of  $4 \times 4 \text{ cm}^2$ . This also means that the DDR4 memory is not in the beam path.
- Mounting of the development board at exactly the right position of the beam is ensured by vertical and horizontal lasers. These lasers show the exact middle of the proton beam, while also ensuring the FPGA is horizontal and not at an angle.

Figures of the test setup as used in the tests are shown in Figure 2.1 and Figure 2.2.

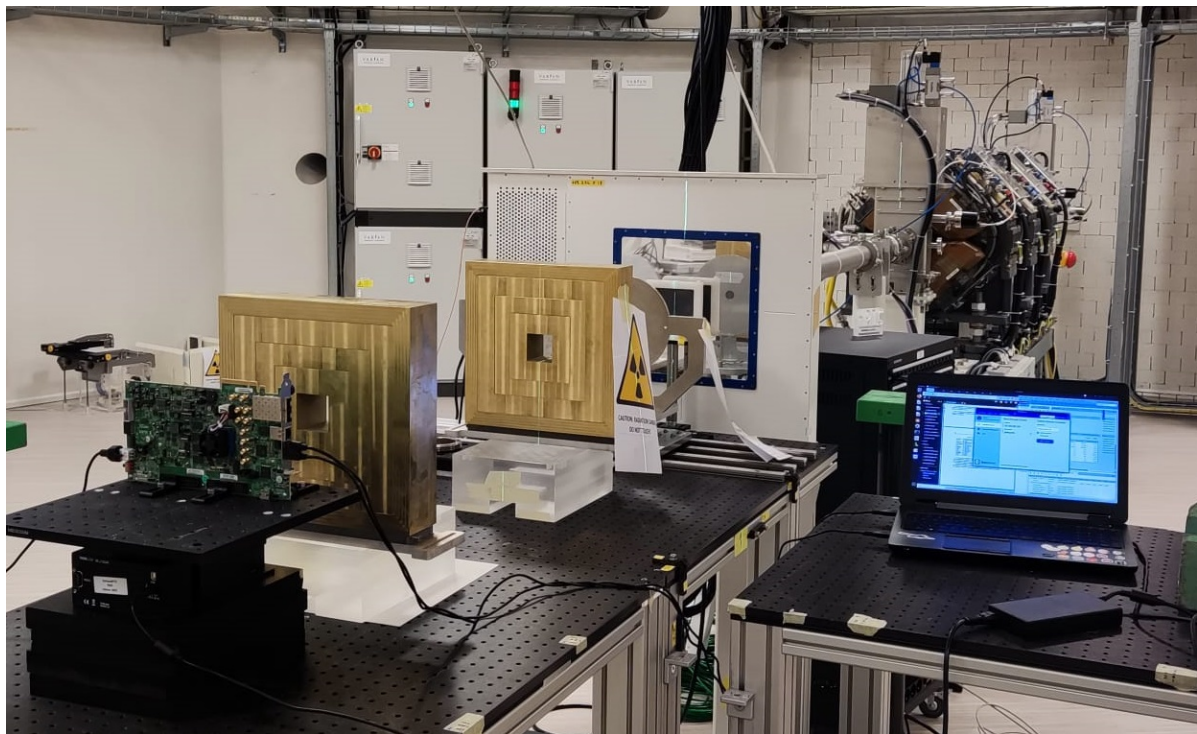


Figure 2.1: KCU105 board clamped in place for the test.

In the left figure of Figure 2.2 the distances between the different elements in the beam path can be seen. The green line in the right figure in Figure 2.2 shows the beam centred on the center of the FPGA. It can be seen that the DUT is backside irradiated.

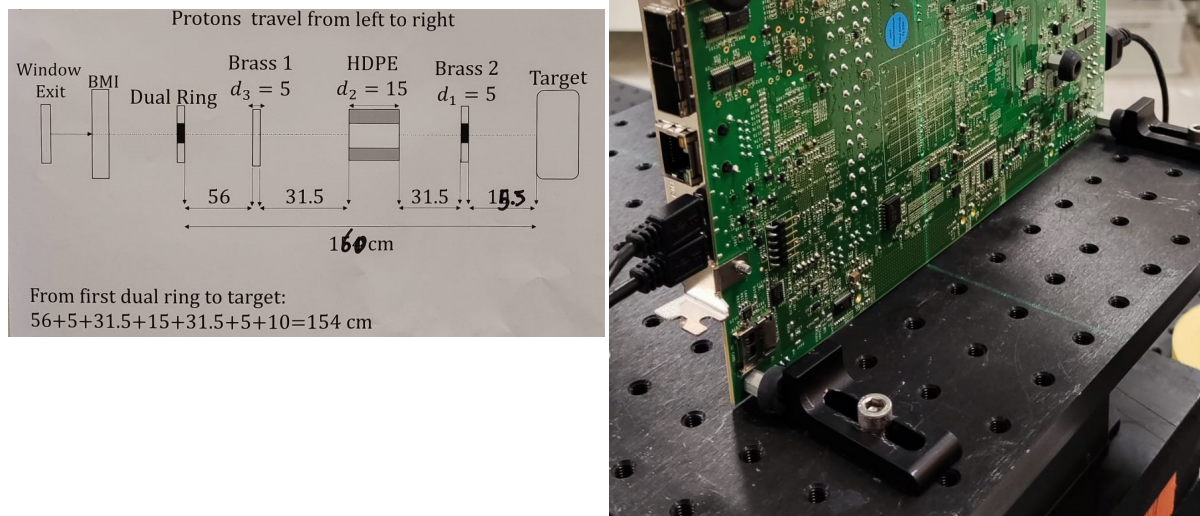


Figure 2.2: Beam line setup (left) the irradiated side of development board (right).

### Safety

Safety of the DUT has been discussed in subsection 1.2.1. Safety of the researchers involved is guaranteed by the facility in a number of ways.

There will be no people present in the room during irradiation and if the room is entered after irradiation, all applicable components will first be checked for radioactivity. Furthermore, all researchers will be equipped with a dosimeter measuring the radiation received.

To prohibit any excessive radiation on the researchers after the test, the DUT will not be given to the researcher after test but will first have to reach a certain threshold to be picked up from the facility.

### The test bunker at HollandPTC

The DUT is connected to a laptop in the radiation room. This laptop in turn is connected to a laptop in the control room via TeamViewer. A LAN connection is preferred and used over a wifi connection to ensure a stable connection. Using this set-up, the KCU105 board could be controlled by the researcher at all times.

The facility also provided live camera footage of the DUT, which proved helpful on multiple occasions. For example when a hard-error occurred a led turned red, which indicated operation should be stopped and the board should be investigated. It also proved helpful in the last facility test, where the cameras were used to have an easy observation of GPIO behaviour.

Entry to the radiation room was possible when needed, for example to test the functionality of the hardware reset in case the debug connection failed, or when a board power cycle was needed after a hard reset. A power cycle could not be performed remotely.

A schematic representation of this set-up is shown in Figure 2.3.

### Connecting to the board

NOEL-V is developed and provided by Cobham Gaisler. A company which also provides the GRMON debug tool. This debug tool will be used to connect to the FPGA programmed with NOEL-V. The PC can be connected to the development board using JTAG, UART or ethernet.

An extensive list of commands possible in GRMON is given in Cobham Gaisler AB [38]. Options used while connecting to the board are, for instance, baud rates. When a connection is made, load, verification and backtrace operations can be performed on the FPGA using GRMON.



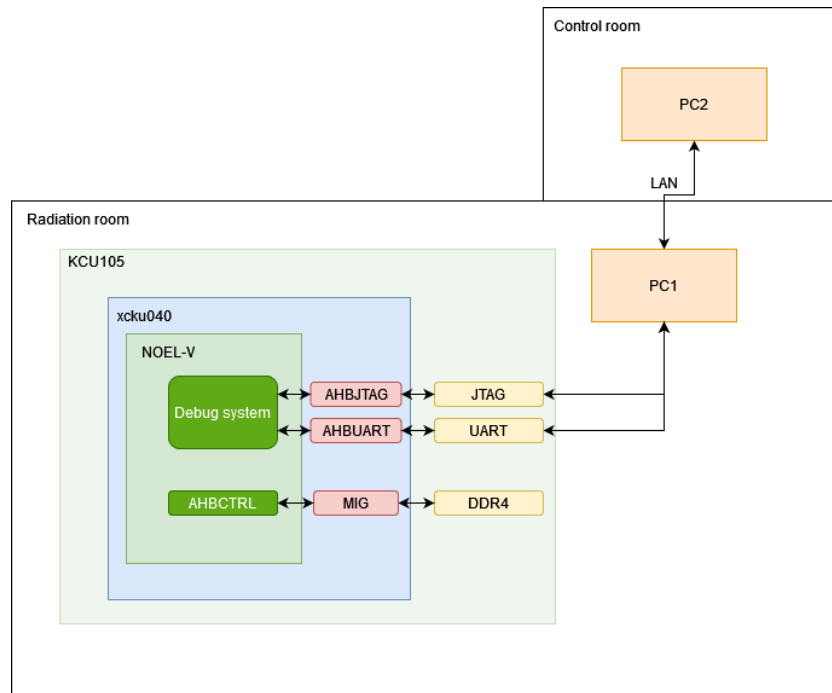


Figure 2.3: Schematic representation of the setup in the radiation room and how two PC's will be used for communication.

## 2.3. Beam properties

As no previous tests employing an electronics device as DUT had been performed at the facility, not much was known about the beam properties before the first test. At the facility it became evident that three beam energies were available, namely 100, 150 and 250  $MeV$ . This translated to 70, 120 and 220  $MeV$  at the DUT due to the effect of the air path to the device and collimators. Having only 3 (high) energies available means that it might not be possible to construct a energy-cross section graph if all three energies are above the 'knee' region.

The beam area used is  $4 \times 4 \text{ cm}^2$ . This area was chosen as it coincides almost perfectly with the FPGA area. As the beam area is not easy to change within tests, the beam area was kept constant throughout all tests.

For the test setup the facility tested the uniformity of the beam. Plots of this uniformity are made as can be seen in Figure 2.4. Any change in collimator spacing would have lead to new calibration operations.

### Flux characteristics

The beam flux, being possible within a range of values, is chosen empirically. Error rates were observed for multiple levels of the flux. First starting off at the lowest flux available at  $2 \cdot 10^6 \text{ p/cm}^2 \cdot \text{s}$ , after which the flux is increased until a desired error rate was found. The highest energy available is used to perform these flux calibration tests to limit the accumulation of radiation damage [22].

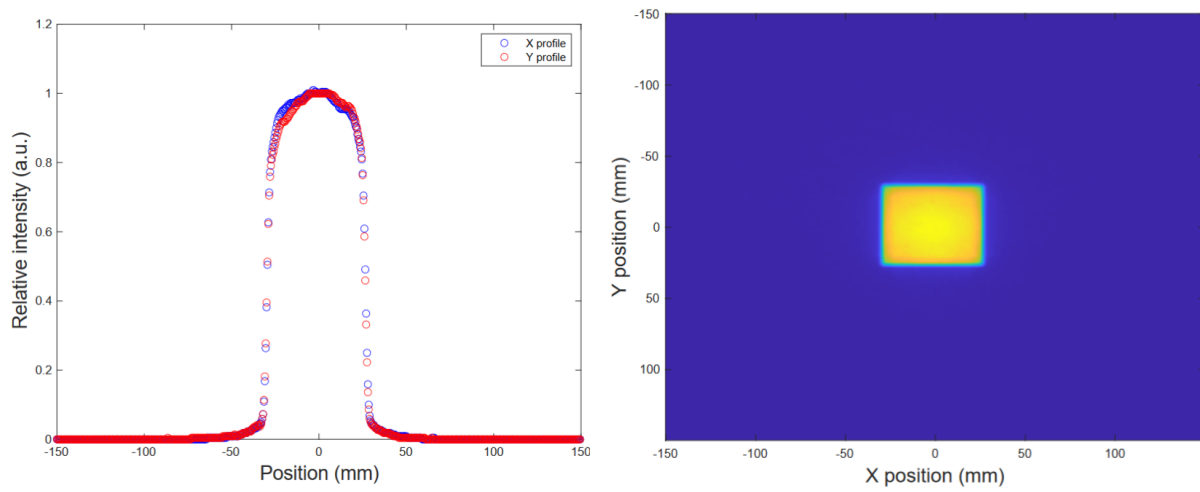


Figure 2.4: Uniformity of the beam over the width of irradiated area (left) and a plot of the beam 4x4 cm area (right).



# 3

## NOEL-V soft processor

So far a proper radiation hardness assurance of the NOEL-V soft processor has not been performed. Therefore, this research focuses on this processor and its operation under proton irradiation. To perform this radiation characterisation effectively, a complete understanding of the soft processor is needed.

Cobham Gaisler, the developer of NOEL-V, provides the processor as a synthesisable Vivado project (VHDL model) and in the form of ready-to-use synthesised processors in 3 different configurations. At first the general components of the soft processor will be discussed assessing their general susceptibility to radiation effects. Afterward the possibilities of different configurations and the ready-to-use processors will be discussed.

### 3.1. General

NOEL-V is a model of a processor based on the RISC-V ISA. In its most elementary form (the so-called Tiny configuration) the processor makes use of only integer and multiplication operations, employs two small 1 *KiB* instruction and data caches with a single issue pipeline. The model is highly configurable, additions in the form of floating-point unit (FPU), parallelism of the pipeline, frequency, and size of L1cache are all possible when desired.

A schematic of the NOEL-V core is shown in Figure 3.1. As noted, the FPU is optional as is the MMU.

In addition to the NOEL-V core, a number of peripherals are added for proper operation. These peripherals can be customised. One can for instance add an Ethernet debug link or L2cache.

The general architecture of a NOEL-V soft processor is depicted in Figure 3.2.

#### NOELV core parts of interest

The level 1 cache size can be altered by the user. The L1 cache follows a Harvard architecture, meaning there are separate pathways for data and instruction caches(hereafter called dCache and iCache respectively). The instruction and data caches can be individually addressed and written to.

In its smallest configuration, the dcache and icache have a size of 1 *KiB*. This is achieved by employing a 1-way 32 line cache with 32 bit lines. The caches can be configured to have up to 16 *KiB* size. The biggest cache employs 4-way, 128 line cache using the same 32 bit lines. One valid bit is present for every data cache line.

#### 3.1.1. IP cores

It is important to know the functions of the parts to be able to make an assessment of points of interest for radiation testing. Not all components will be equally at risk of radiation damage and focussing on the more important components will improve the characterisation of the radiation hardness of the soft processor.

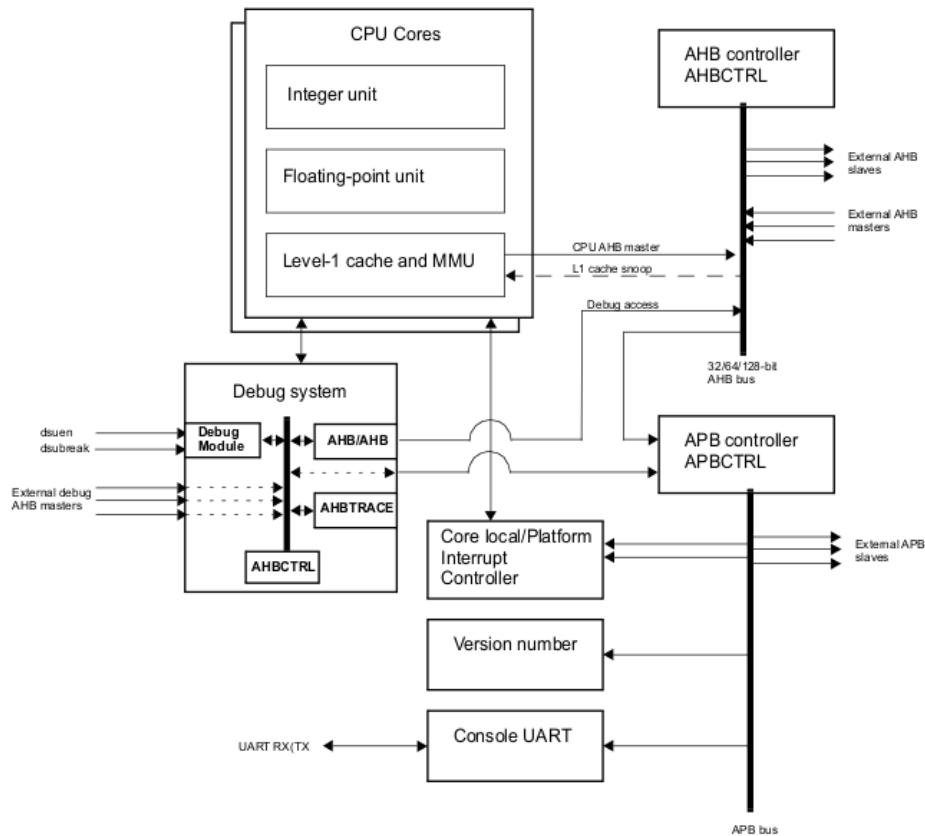


Figure 3.1: Schematic of the NOEL-V subsystem [39].

## General IP cores

### AHB/JTAG & AHB/UART debug links

Two debug links are available, namely the JTAG and UART debug links. The possibility also exist to use both at the same time.

Differences between the links are easily visible when looking at the block diagrams, as depicted in Figure 3.3. Where the JTAG has one interface with the AMBA AHB bus for both receiving and sending data, the UART link has two interfaces to perform the receiving and sending separately. It is not immediately evident from these figures which one will be more radiation hard, but it is evident that there is a possibility of different behaviour.

### AHB Read Only Memory

As the name suggests, a read-only memory (ROM) is generated within the NOEL-V system. The ROM is configurable with different bit widths, 32, 64 or 128 bits. In all the configurations used during this project the 32 bit ROM is used to increase comparability.

Stored data in the ROM is susceptible to SEU, the ROM size is 512 MB.

### GRGPIO General purpose I/O port

A schematic of one IO line is shown in Figure 3.4. From this figure it is directly evident that this core encloses multiple susceptible elements. An upset in the any of the flip-flops will lead to wrong behaviour, as well as the multipliers being susceptible.

### MIG

The Memory Interface Generator (MIG) is provided by Vivado as opposed to the rest of the components which are IP from Cobham Gaisler. The MIG provides the interface between the AHB Bus and

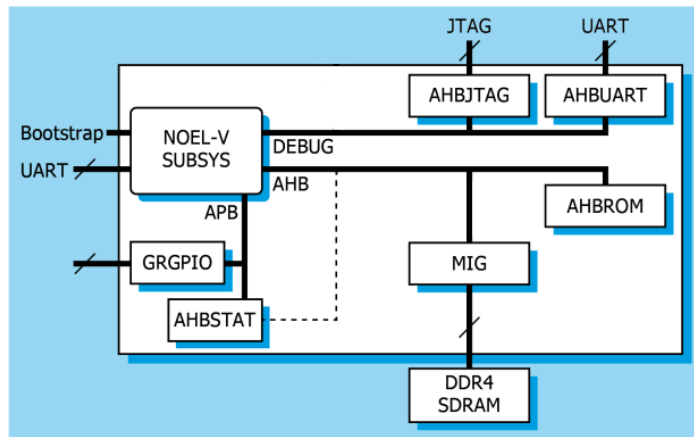


Figure 3.2: General architecture of a NOEL-V soft processor [40].

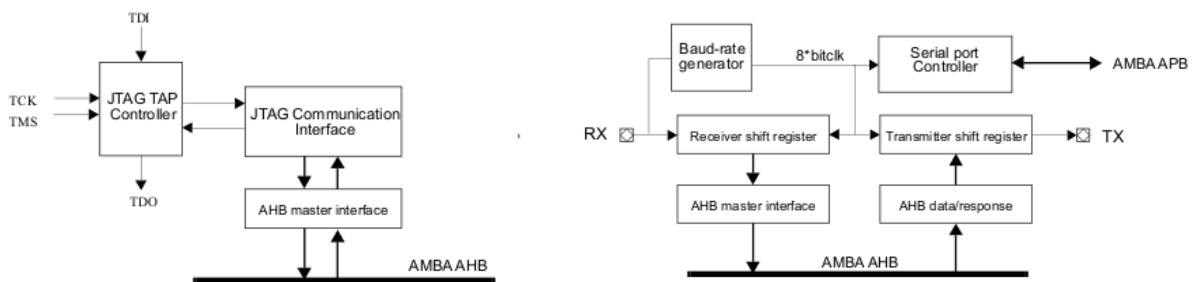


Figure 3.3: Block diagrams for AHBJTAG (left) and AHBUART (right) [39].

the (in the case of the kcu105 board) DDR4 Synchronous dynamic random-access memory (SDRAM).

The DDR4 SDRAM is located on the development board itself and not in the FPGA. Therefore, the radiation received by this component will be negligible compared to the FPGA. The MIG however is located on the FPGA and is therefore of interest.

**Supporting IP cores**

**AHBSTAT**

The status registers (STAT) monitor the AHB bus for erroneous transactions. When a erroneous transaction occurs, information about the transaction is stored and an interrupt is thrown. In case Error Detection and Correction (EDAC) is present in any of the components, the error is also saved but no interrupt is thrown. As the used configuration of NOEL-V does not contain any component that employ EDAC, this is not applicable to the use case.

**AHBTRACE**

The AHB Trace buffer stores data transfers in a circular manner. This is very helpful for debug operations and during radiation testing. Using the `inst x` command in GRMON the trace buffer can be accessed and the last

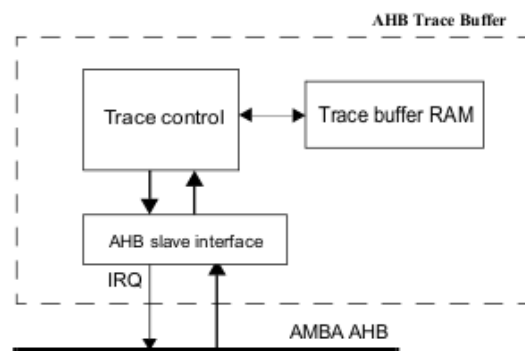


Figure 3.5: Block diagram of the AHBTRACE IP core [39].

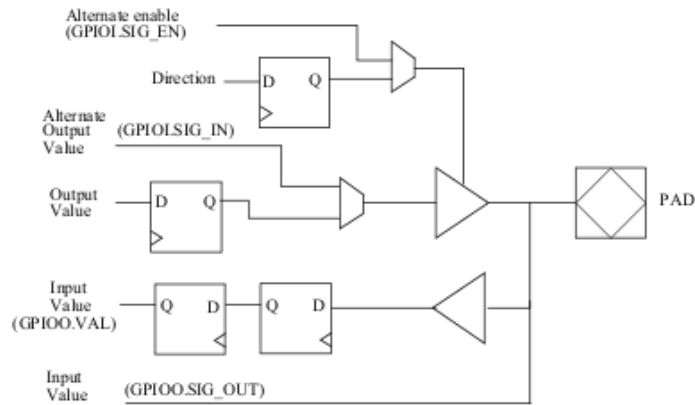


Figure 3.4: Layout of the GPIO ports [39].

x entries in the trace buffer will be printed.

An upset due to radiation in this buffer would lead to false information when reading it out but will not lead to a fault during the test.

As expected, and confirmed by the block diagram depicted in Figure 3.5, the core implements some storage volume to store the contents of the buffer. The researcher needs to keep in mind that this RAM can also suffer from upsets.

The size of the trace buffer RAM can be controlled but is set to 1 *kbytes* by default. This accounts to 64 lines of 16 *bytes* length. Together with the fact that the trace buffer is constantly updating, the information gathered from the trace buffer is deemed reliable. To help extraction of data from the processor, the AHBTRACE size was increased by 8x.

### AHBCTRL

The AHB controller is a multi-purpose core, fulfilling the functions of bus arbiter, bus multiplexer and slave decoder. The AHBCTRL uses play&play information from the connected masters and slaves. This data is stored in a ROM address area of 4 *kB* width. Of this, a maximum of 25% are essential bits. The AHBCTRL core is, therefore, seen as a low risk core.

### APBCTRL

Unlike the name suggests, is the APBCTRL an AMBA AHB to APB bridge. The component fulfills the tasks of the APB bus master. It supports up to 16 slaves, and just like the AHBCTRL it decodes information from the slaves.

To fulfil this function, information about the connected slaves is stored in the top 4kB of the bridge address space. This means that there is a very small chance that an upset in this component occurs, and an even smaller chance that this is an essential bit.

### 3.1.2. Component resource utilisation

All the above mentioned components will be synthesised and programmed into the FPGA. The resource utilisation of every component can be an indicator of the extend the impinging particles can influence its operation.

As an example, the implemented design of the tiny processor will be investigated. As discussed above, this is the tiniest implementation of the NOEL-V subsystem. The size of the subsystem is variable but the sizes of the peripherals are not. The area breakdown can be seen on the right in Figure 3.6.

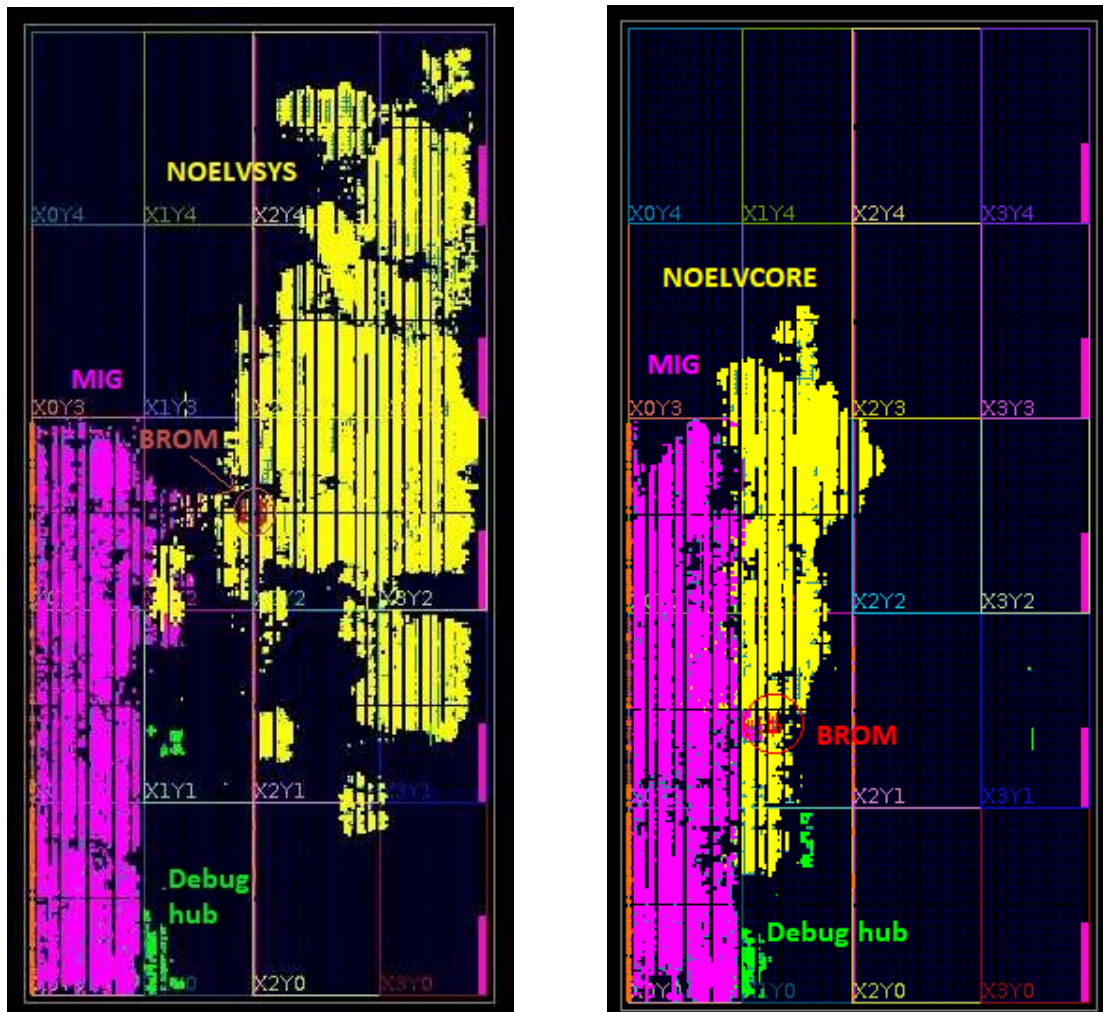


Figure 3.6: Marked layouts of the dual configuration (left) and the tiny configuration (right).

It can be seen that for this configuration the NOEL-V subsystem and MIG use the most resources, with the debug hub and ROM only taking small amount compared to the former two. GPIO pads are very hard to spot as they take up only very small area. The absolute number of the resource utilisation are given in Vivado. With this resource table the visual expectations can be put into numbers. Here, one can see that for example the MIG controller uses about nine times as much LUT as RAM than the NOEL-V subsystem, but the latter uses many more Digital Signal Processors (DSPs).

The NOEL-V core can be broken down into its constituents. This is depicted in Figure 3.7. This is again for the tiny configuration. It can be seen that the integer pipeline takes most resources and will, therefore, be vulnerable but also more expensive to apply TMR to.

Resource utilisation for every configuration has been compared and has shown that the general purpose processor (GPP) configurations lead to more than double the resource utilisation for the integer pipeline. The resource utilisation for the L1cache is three times as big, but components like the AHBTRACE stay exactly the same. This can be very helpful for identifying the origin of the error.

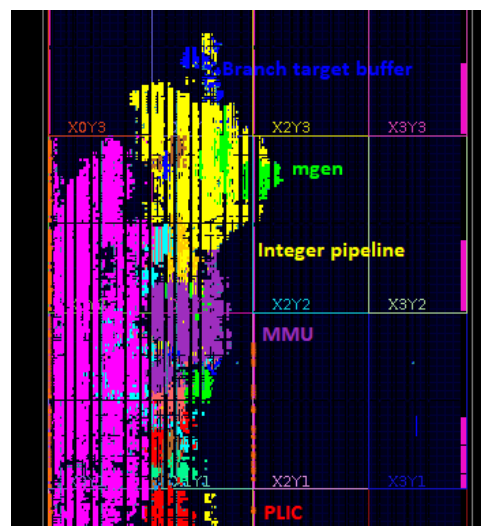


Figure 3.7: Marked layout of the NOELVSYS in the tiny configuration.



### 3.1.3. Existing mitigation measures

As discussed in Figure 1.2.3, there are some mitigation features in the form of the ECC and SEM IP cores ready to use that are provided by Xilinx. It is not determined, however, to which extent these mitigation features have been implemented into the NOEL-V soft processor.

Documentation does not contain conclusive information. Investigation of the VHDL file through Vivado shows the inclusion of the SEM IP core but no UART output is found. Investigation during particle testing will have to determine whether the core is actually implemented and working.

As has been the case for the LEON processors, fault tolerant by design processors will be brought to the market by the manufacturer. This fault-tolerant version of NOEL-V will be available under a commercial license.

## 3.2. NOEL-V example designs

Cobham Gaisler also provides some ready-to-use NOEL-V processors. The differences with the aforementioned synthesisable VHDL model are that they employ some features that are not freely available, like the L2cache. The architecture is very similar to the one shown above, and is depicted in Figure 3.8.

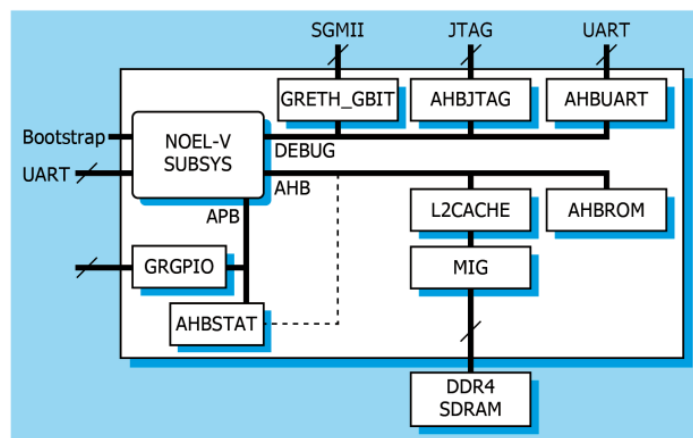


Figure 3.8: General architecture of a NOEL-V soft processor with L2cache [40].

As new components are added to the soft processor, these ready-to-use processors can be used to evaluate the performance of these parts under radiation.

The available example designs are summarised in Table 3.1.

Table 3.1: Available ready-to-use configurations provided by Cobham Gaisler. Two more are expected during 2021 but are not available as of writing.

Configuration	Risc-v Extension [39]	L1Cache size	# of processor	MMU	PMP	L2cache	FPU
<b>EX1</b>	IMA	8 <i>kiB</i> , 2 ways	1	No	Yes	Yes	-
<b>EX2</b>	IMAFDH	16 <i>kiB</i> , 4 ways	1	Yes	Yes	Yes	NanoFPU
<b>EX4</b>	IMAFDH	16 <i>kiB</i> , 4 ways	4	Yes	Yes	No	NanoFPU

Although beneficial features are included, the downside of these configuration is, however, that they are in no way configurable and not all supporting files are provided. This means that all include a Ethernet debug link, which is not used during testing/in operation leading to an area overhead without providing any benefits.

Furthermore, the lack of supporting files means that Vivado verification, which reports the CRAM upset rates, cannot be performed. However, this should not be a problem as data from synthesizable configurations can be extrapolated for the area increase.

### 3.3. Configurations of interest

The possibilities of configurations are just short of endless. To keep the workload manageable, five useful configurations have been chosen to be used. These configurations, including some important characteristics, are depicted in Table 3.2. Three are of the self-synthesizable type as discussed in section 3.1 and two are ready-to-use as discussed in section 3.2.

Table 3.2: Used configurations in the beam test. Cache in all configuration follows a Harvard architecture with separate data and instruction caches. GPP can be configured in single- and dual issue pipeline.

Configuration	RISC-V Extension [39]	L1 Cache size (both)	MMU	PMP	L2cache	FPU
<b>Tiny</b>	IM	1 KiB, 1 way	No	No	No	No
<b>Minimal (EX1)</b>	IMA	8 kiB, 2 ways	No	Yes	Yes	No
<b>GPP (single)</b>	IMAFD	16 kiB, 4 ways	Yes	Yes	No	NanoFPU
<b>GPP (dual)</b>	IMAFD	16 kiB, 4 ways	Yes	Yes	No	NanoFPU
<b>HPP (EX2)</b>	IMAFD	16 kiB, 4 ways	Yes	Yes	Yes	NanoFPU

The configurations above have been chosen to, in conjunction with different test programs, give the best possible overview of the radiation hardness of components in the NOEL-V soft processor.

The most notable difference between GPP and HPP configurations is the inclusion of a bi-directional AHB-AHB bridge in the HPP configuration to increase performance. This feature is not available under the free-to-use GPL license.

#### Resource utilisation

The properties of the configurations of interest lead to different amounts of utilisation by the subsystem. The utilisation of different components on the FPGA and its percentage of the available resources is depicted in Table 3.3.

Utilisation will have an effect on the amount of essential bits the configuration bitstream contains, as more design elements need to be implemented. It is therefore expected that CRAM upsets will increase with increasing area overhead.

Table 3.3: Utilisation of available resources by the different configurations

Type	Utilisation (%)				
	Tiny	Single	Dual	EX1	EX2
LUT	34677(14.3)	54396(22.44)	62625(25.84)	39393(16.25)	49101(20.26)
LUTRAM	2044(1.81)	1748(1.55)	1748(1.55)		
FF	33819(6.98)	41802(8.62)	43245(8.92)	26563(5.48)	29554(6.10)
BRAM	35(5.83)	56(9.33)	62(10.33)	104(17.33)	107(17.83)
DSP	19(0.99)	21(1.09)	21(1.09)	16(0.833)	18(0.94)
IO	153(29.42)	153(29.4)	153(29.42)		
BUFG	10(2.08)	10(2.08)	10(2.08)		
MMCM	1(10)	1(10)	1(10)		
PLL	3(15)	3(15)	3(15)		

As can be noted, statistics about the example configurations are not as complete as the data that can be gathered from Vivado. However, its only the LUTRAM that changes between the tiny and general purpose configurations. It can be assumed that the LUTRAM for EX1 is close to the amount of tiny and EX2 closer to the amount in single & dual configurations.

It is also important to keep in mind is that after communication with Cobham Gailser, they admitted that the data was out of date and will be changed in the newer update. Therefore, conclusions involving

the utilisation of the example configurations should be taken doubtfully.

### Performance

Previous to particle testing the CoreMark scores for all configurations are already determined, these are:

- Tiny configuration: 1.63 CM/MHz
- EX1: 3.01 CM/MHz
- single issue GPP configuration: 3.05 CM/MHz
- Dual issue GPP configuration: 3.38 CM/MHz
- EX2: 4.46 CM/MHz

These scores are different from the one reported by Cobham Gaisler themselves, where they report a (high) score of  $4.69 \text{ CM}/\text{MHz}$ . It is not clear whether the dual issue GPP or EX2 configuration is used, but it is clear that found scores here under perform. The difference is possibly due to the processor configuration and toolchain used. However, the difference is only marginal and the processor still scores very high compared to other general purpose soft processors as discussed in subsection 1.2.3.

The performance of the tiny configuration can be compared to low to medium end microprocessors, whereas the other 3 configurations (EX1, tiny, dual) outperform processors employing similar levels of instruction level parallelism. All configurations can thus be of use to a satellite designer, depending on requirements.

A notable delta between CM scores, or actually a lack of delta, is observed between the EX1 and single issue configurations. The scores being similar is likely due to the inability of CM to use the FPU. The inclusion of the FPU is, together with the increase in L1cache and absence of L2cache, the biggest architecture change. It is believed that the L2cache and HPP-to-HPP bridge inclusion compromises for the lack of L1cache size for the EX1 configuration. The lack of FPU is of no influence as this is not used by CM. It can then be seen by the comparison of dual issue and EX2 that the influence of the L2cache and HPP-to-HPP bridge have a very big influence on processor performance under CM.

### Scaling

The total percentage of bits in the bitstream used cannot be found out by the researcher. It is chosen to take the three most available components and their respective utilisations and take a weighted average of this. As can be seen on the left in Table 3.4, those are LUT, LUTRAM and FF. For every configuration the utilisation of these 3 components is averaged and this is taken as the resource utilisation of the configuration. This resource utilisation will be used to scale configuration by their CRAM usage.

As the LUTRAM values are not given for the example configurations, it is chosen to take the values of the Dual configuration for the EX1 and EX2 configurations. The (estimated) resource utilisations for the configurations are reported on the right in Table 3.4.

Table 3.4: Example of utilisation of components, for this case the Dual configuration, with the respective amounts (left) and calculated resource utilisation for every used component (right).

Resource	Utilisation	Available	Utilisation (%)
LUT	62616	242400	25.83
LUTRAM	1748	112800	1.55
FF	43239	484800	8.92
BRAM	62	600	10.33
DSP	21	1920	1.09
IO	153	520	29.42
BUFG	10	480	2.08
MMCM	1	10	10.00
PLL	3	20	15.00

Config	Utilisation (%)
Tiny	8.4
EX1	8.1
Single	11.7
Dual	12.8
EX2	9.6



# 4

## Test plan

Availability of multiple testing opportunities at the facility means that a thorough test plan is needed to make sure time is used effectively. To maximise qualitative data extracted during these tests, different tests are used to gather different data about data extraction, mitigation measures, and general susceptibility of the NOEL-V processor. Starting off with high level test requirements outlined below, every test will have its own associated requirements, which, when fulfilled will lead to a complete characterisation of the processor.

- Initial aim of test one is to determine an optimal flux that maximises data output without needing extensive test times. This is achieved using error rate and functional failure rate of the processor. Using the flux, the FPGA is configured to run a benchmark and radiation characteristics of the FPGA and processor are extracted.
- Test two is used to further investigate data extraction methods and to gain more insight in the workings of the NOEL-V soft processor. This leads to additional radiation data and insight into the effects of changing energy, use of benchmark and mitigation techniques.
- Test three is more in-depth into the core of the processor. Lessons learned during the first two tests are used to get as much information about the susceptibility of the different NOEL-V configurations as possible. Data obtained during the third test will be combined with gathered data during the previous tests to get a complete picture of the radiation characteristics of the NOEL-V processor and draw conclusions.

The development of the tests is explained in more detail in the following sections.

### 4.1. Particle test 1

For the first facility test, requirements are drafted as shown in Table 4.1.

Table 4.1: Requirements for the first facility test

Code	Test requirement	Rationale
FT1-DAQ-1	Data shall be transferred via JTAG.	Communication with the board shall be possible.
FT1-DAQ-2	Upset amounts in CRAM shall be retrieved.	Upset rates are necessary for CRAM cross-section calculations.
FT1-DAQ-3	Readback capture shall be performed after irradiation.	Readback file combined with logic location file can be used to upset locations.
FT1-DAQ-4	After a failure, communication shall still be possible with the DUT	Data shall be extracted after a failure.
FT1-MO-1	The DUT shall be backside irradiated	Influence of the fan has to be circumvented.
FT1-MO-2	The DUT shall be irradiated at 90° beam angle.	To allow maximum comparability with other conducted tests.
FT1-MO-3	The DUT shall not be able to move during the irradiation.	Beam is centered on the FPGA.
FT1-MO-4	The DUT shall not be moved in between tests.	Different tests must be able to be compared to each other.
FT1-BE-1	An optimum for the beam flux shall be determined.	An optimum between statistic relevance and acceptable behaviour is to be found.
FT1-TI-1	The test shall not take in excess of 4 hours.	Proton beam is only limittly available.

#### 4.1.1. Developed test

In order to fulfil the requirements, a test plan is developed. The FPGA will be loaded with the bitstream using Vivado, afterwards *GRMON* will be used to connect to the board. A log will be kept at all times for every run. Dhrystone will be run until failure to find common error modes and get a feel of the susceptibility of the soft processor. This test flow is depicted in Figure 4.1.

After irradiation the *GRMON* is disconnected, after which a tcl file is run which reconnects Vivado to the FPGA and runs verification and readback operations. The error rates are looked at to determine if a desired error rate is reached and otherwise the flux is altered and the test is ran again.

Particle test one has to start off with flux optimisation. Choosing a flux too high will result in multiple errors occurring at once, severely inhibiting the researchers ability to discern radiation effects. However, a too low flux would lead to excessive testing times, which would inhibit the researcher from performing the desired amount of tests and reach statistical significance of the data, as test time is limited.

To reach the desired error rate, use is made of Vivado verification and statistical regularities where one out of approximately 100 errors is a functional error. Starting with the lowest flux, if desired, the flux will be increased until the desired error rate is found.

The desired error rate is determined to be about 1 error per second, leading to a functional failure on average every 100 seconds.

The found flux will be used for all other tests. The processor shall be running a space representative

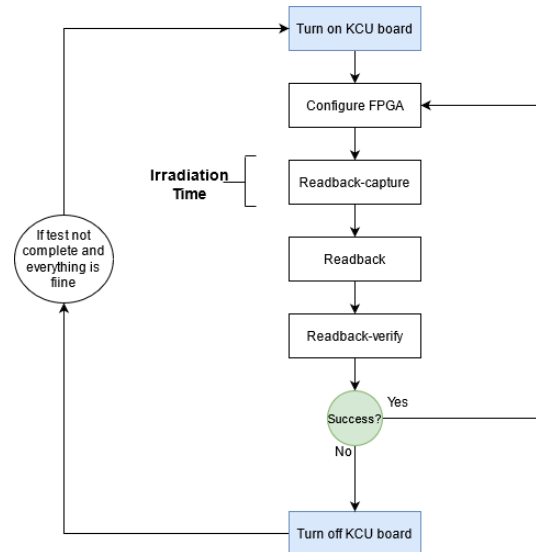


Figure 4.1: Test flow employed during test 1.

program and radiation effects can be observed by running the processor multiple times to failure. For this test, use was made of Dhrystone benchmark due to availability and simplicity. An example of the benchmark for NOEL-V is provided by Cobham Gaisler and was synthesised using the gcc compiler of the riscv-gnu-toolchain.

Data gathered during this test are the error rates of the CRAM, common failure modes of the internal logic, and initial susceptibility of the NOEL-V processor.

### Dhrystone

Dhrystone is a benchmark for testing the integer performance of a processor. It was developed in 1984. Dhrystone scores are available for many different processors and therefore correct operation can be verified and compared. Dhrystone specifically focused on integer and string handling, no floating point operations are performed.

## 4.2. Particle test 2

In order to set the next step in extracting the data needed as outlined in subsection 1.4.2, a number of requirements are drafted. These in addition to the requirements of test 1, like data acquisition (*DAQ*) and mounting (*MO*) requirements.

Table 4.2: Requirements for the second facility test

Code	Requirement	Rationale
FT2-SE-1	The test shall show the influence of software running on the processor.	Differences in influence of software can show in which use cases the processor performs best.
FT2-DL-1	The test shall show the least vulnerable debug link.	Multiple debug links are available, but little is known about the relative radiation susceptibility between them.
FT2-EN-1	The effect of energy on upset rates shall be investigated.	Finding the energy dependence of cross-sections will allow the researcher to create a cross-section-energy graph.
FT2-EN-2	The effect of energy on upset severity shall be investigated.	Upset dependency on particle energy will be helpful to get in-orbit upset rates.
FT2-SEM-1	The inclusion of the SEM-IP core shall be investigated.	Possible inclusion of the SEM IP core will mask numerous errors, and therefore influence found cross sections.
FT2-L1C-1	Extraction of L1Cache upset rates shall be provided.	The individual susceptibilities of components is the NOEL-V can be helpful when using targeted TMR.

### 4.2.1. Developed test

These requirements will be satisfied by drafting a test plan which addresses all requirements. Starting off with the test procedures during the test to extract data, the nominal test flow during the test is shown on the left in Figure 4.2. A similar flow is followed compared to test 1, with the inclusion of a more detailed procedure to handle the processor after termination of the program (be it by automatic or manual termination), as shown on the right in Figure 4.2.

To succeed in the goal of providing clarity about masking effects present in the NOEL-V processor, the processor is synthesised with the SEM IP core performing both mitigation and detection (default setting) of SEU, and only detection of SEU mode. After no differences were found in operation running the test setup at home, it was clear that a test in the proton beam was needed to provide a decisive answer to the question. In case active error mitigation was present on the NOEL-V processor, a significant difference in CRAM upset rates should be observed between the two different processor versions. This difference would be about 1.77x as observed by D. Hiemstra et al. [18, 34]

Furthermore, during the second test the two debug links are tested, in combination with a test of the

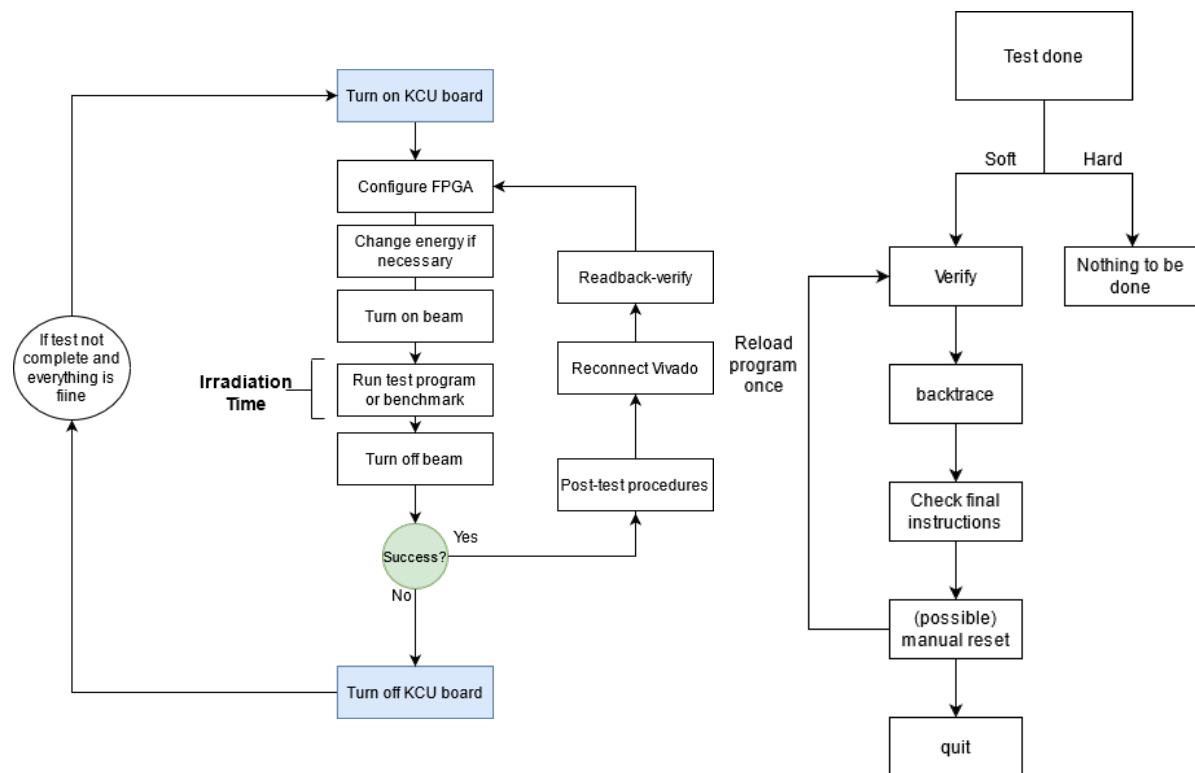


Figure 4.2: Test flow employed during test 2 (left) and post-test procedures employed (right).

influence of software running on the soft processor. Pre-test testing already revealed that the UART debug link is considerably slower. Due to this slow operation of the link, the expectation of the test is that the UART connection is indeed less reliable. Both debug links will be used separately on the same FPGA configuration and any difference in error rate would indicate which debug link is more robust to radiation.

Lastly, as the facility provides multiple available proton energies, a test was developed to test whether the energy has an impact on error rate. Three energies are available at the facility; 70, 120 and 250 *MeV* (at the DUT). The plan is to test all 3 energies, starting with the two lowest energies, with the board running the DhryStone benchmark. Expectations are that the lowest energy level leads to the lowest energy rate, as can be observed in energy-cross section graphs constructed for proton tests on other FPGA. When the 'knee' region of the energy-cross section graph is already reached at 70 *MeV* (as it is already a quite high energy), no difference in upset rate will be observed.

In order to cut time spend testing, a number of these tests can be combined. The tests to determine the influence of debug link will be running the CoreMark benchmark until failure, as opposed to the Dhrystone benchmark that will be running during the latter tests. The energy and mitigation tests will be combined in the following way:

- High energy (120 *MeV*) with the SEM IP core set to detection only.
- Low energy (70 *MeV*) with the SEM IP core set to detection only.
- Low energy (70 *MeV*) with the SEM IP core set to detection&mitigation.

As noted, during these tests the Dhrystone benchmark will be run until failure. The Dhrystone benchmark is used as this allows for the new data to be compared to data obtained during the first test, warranting the exclusion of the "High energy (120 *MeV*) with the SEM IP core set to detection&mitigation" as that has already been performed during the first particle test. Fluence-To-Error(FTE) and CRAM upset rates will be compared to observe any differences between the runs.

In addition to these tests, L1cache susceptibility will be measured. There are multiple options to measure this sensitivity, being to load the dCache with a checkerboard, all 0's or all 1's and check all



these values after irradiation. Checking for upsets after irradiation can be done by using the grmon commands *dCache* and *iCache*, to read the values stored in the L1cache.

During at home testing it already became evident that Dhrystone uses 6.88% of the dCache during operation, where CoreMark uses 55.8%. This means that CoreMark stresses the processor much more, but will also be more heavily impacted by any upsets in the L1Cache or L1Cache logic.

### CoreMark

CoreMark is a benchmark that measures integer performance, developed in 2009 and thus being much more recent than Dhrystone. The benchmark was indeed intended as a replacement for Dhrystone, and is thus comparable to Dhrystone in the operations it performs. Four different subprograms are executed in CoreMark: list searching and sorting, matrix operations, a state machine on a series of numerical inputs, and a cyclic redundancy check. In comparison, Dhrystone mainly focuses on string operations.

## 4.3. Particle test 3

Using the insight gained during the first two tests, a more detailed radiation characterisation of the NOEL-V soft processor could now be performed.

As was the case for test 2, test 3 also inherits certain requirements. All requirements of test 1 will be adhered to with the exception of *FT1-DAQ-2* & *FT1-BE-1*. Requirements *FT2-SEM-1* & *FT-L1C-1* will be inherited from test 2. One additional requirements will be added to be able to fulfill the high level test requirement of test 3:

*FT3-NV-1*: The test shall investigate the susceptibility of different parts of the NOEL-V.

### 4.3.1. Developed test

For this test 3, different test programs are developed of which one or multiple are always applicable to one of the NOEL-V configurations discussed in section 3.3. The used test programs will be discussed hereafter with its possible combinations and conclusions.

#### Using the modularity of NOEL-V

In order to fulfil this test goal, the NOEL-V in different configurations will be leveraged as noted before. The configurations of interest have already been determined, meaning the test programs to enable extraction from these configurations are to be developed.

Due to differences in the configurations, multiple layers of test programs have been developed. Most importantly the inclusion of the FPU and L2cache leads to the use of incremental layers.

In Figure 4.3 the general flow of the tests is depicted. This flow is identical for all test layers but differs within the individual blocks.

The developed tcl script that makes sure all tests are started in an identical manner, is changed to disable the l2cache for certain tests. The data of tests with disabled l2cache can be compared to data with enabled l2cache to identify the susceptibility of the MIG core. L2cache cannot be disabled using a grmon command, it has to be disabled when the debug link is established.

#### Developed software

The program run is very different for the different

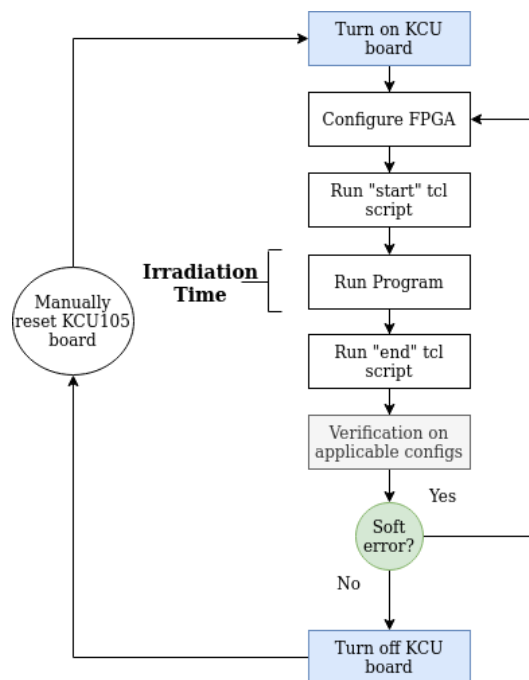


Figure 4.3: General test flow for all tests.

levels. A general overview of the functions of every level is covered below, and the ran programs are developed to fulfil these functions.

After the program has been running continuously for five minutes or a hang in the program is noticed, the program will be stopped and automatically a number of functions will be invoked. These grmon commands will show the latest instructions ran by the processor (command *inst 1000*), the backtrace (command *bt*), dCache, iCache and the verification of the program.

Afterwards, Vivado verification is used to show the amount of upsets in CRAM if it is possible (not possible for EX1 and EX2 as discussed).

In case there is no hard error, the board will be programmed again and the sequence will be performed again, possibly with other combinations of "start" scripts and software running.

## Software levels

**Level 1, called INT hereafter:** L1 caches, SRAM, I/O blocks, ROM

- L1 caches will be compared to a golden checksum to check for possible upsets. As the iCache is the same for every run of the same type, this is an easy compare to the golden run. dCache, however, is dynamically used and allocated.
- CRAM will be checked using Vivado verification function.
- I/O blocks will have their state constantly alternated, which can be visually checked on the cameras as the I/O block will output to the LEDs.
- ROM will constantly be read during operation and alert the tester via JTAG if any upsets occur and will save the error location.

**Level 2, called L2 hereafter:** L2 Cache

The L2 cache will be written with a known checkerboard pattern and disabled, the same will be done for the L1 cache. This way the L2cache can be statically tested. To dynamically test the L2 cache, the L2cache will be enabled. Known results (golden run without radiation) will be compared to the state of the L2 cache after irradiation and any discrepancies will be noted.

**Level 3, called FPU hereafter:** Floating Point Unit

Testing radiation hardness in a floating point capable processor will be checked by running the Whetstone benchmark [41]. The Whetstone benchmark is known to stress floating point operations. As floating point operations are performed, the FPU in the NOEL-V processor will be activated. Both single and double precision floating point operations will be used to get an understanding of the (possible) difference in susceptibility in the single and double precision floating point handling capabilities of the processor.

### Whetstone

Whetstone is a benchmark specifically targeted at stressing the processors FPU, it primarily measures the floating point arithmetic performance. Both single and double precision Whetstone tests are available and used in FPU tests.

## 4.3.2. Combinations and expected results

The possible configurations are noted in Table 4.3.

Table 4.3: Combinations of configuration and test level used during the test.

Configuration	INT	L2	FPU
<b>Tiny</b>	x		
<b>Minimal (EX1)</b>	x	x	
<b>GPP (Single)</b>	x		x
<b>GPP (Dual)</b>	x		x
<b>HPP (EX2)</b>	x	x	x

The expected results from the combinations are depicted in Figure 4.4. These expected results should be kept in mind when analysing the data as it will lead to the correct conclusions.

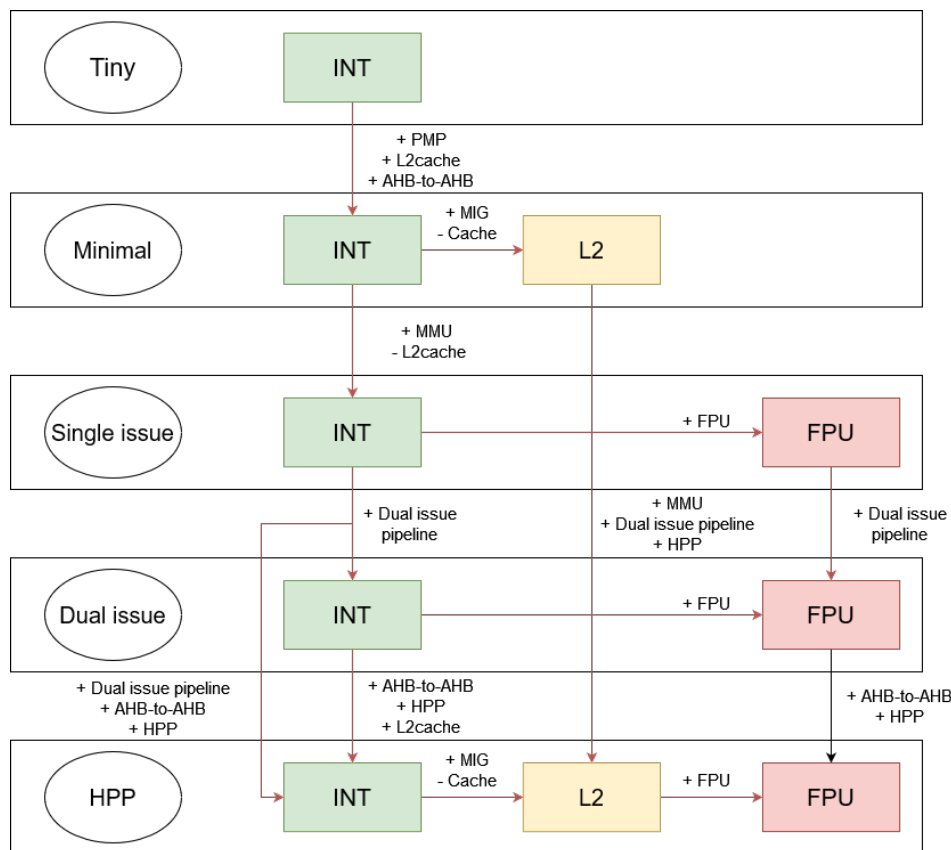


Figure 4.4: Possible finding from the different test levels combined with different configurations.

In this figure, one can see that comparing the test results of all the different tests will result in data about numerous important components of the NOEL-V processor. For example, as the tiny configuration does not employ PMP, has a smaller L1 cache and the AHB-to-AHB bridge is not present, any new upset states will possibly be due to these added components. Using other data, for example, the data from dual issue to HPP on added upsets due to the AHB-to-AHB bridge, the error modes can be narrowed down to a single component.

### Internal operating conditions

As internal operating conditions have a big impact on the susceptibility of components to radiation effects, differences in operating conditions between the configurations have been monitored beforehand. It was found that no differences exist for the configurations, including when running different benchmarks. This means that the possibly found error rate differences will be due to other factors.

The configurations all drawing the same power is also important for the performance comparison. Because it is shown that there is no difference, the performance will only be compared based on area and CoreMark score.



# 5

## Results

During the three tests a combined test time of 174 minutes is reached, in which  $2.14 \cdot 10^{11} p/cm^2$  were fired at the DUT.

### 5.1. Particle test 1

During test one, the FPGA was irradiated over a total time of 59 minutes (beam on time). During the test this meant a total fluence of  $7.93 \cdot 10^{10} p/cm^2$  was reached.

Test one started off with an initial empirical test to determine the optimal flux level to use. This level was found to be  $5 \cdot 10^7 p/cm^2 \cdot s$ . The test results of the tests conducted using this flux level are shown in the following sections.

#### 5.1.1. Errors in CRAM

A plot of the error rates in the configuration memory during the first test is made and is depicted in Figure 5.1. The first four runs depicted are in the absence of software running, the bitstream was thus loaded, left idle and the bitstream was checked for upsets. After these four initial runs the Dhrystone benchmark was running during irradiation.

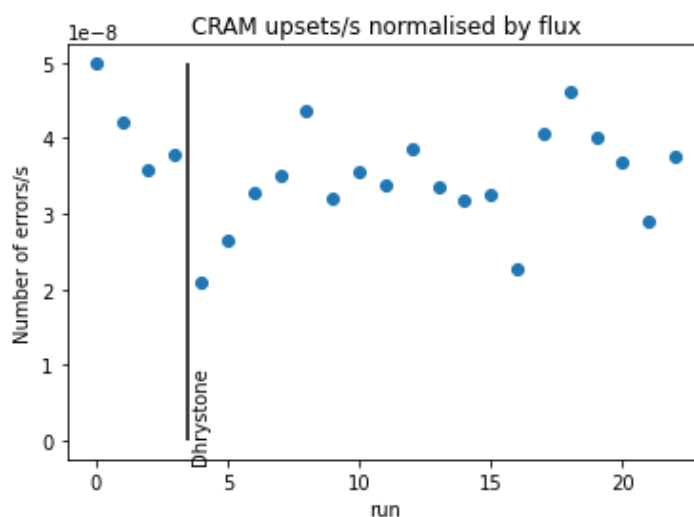


Figure 5.1: Graphical depiction of CRAM cross section during test 1.

As can be seen in Figure 5.1, the CRAM error rate is not influenced by activity of the processor. Dividing by the amount of bits in the bitstream of the processor, the average cross section of errors in CRAM reported is  $2.78 \cdot 10^{-16} cm^2/bit$ .

Unfortunately, Vivado readback was unsuccessful due to the inability of the researcher to handle the mask file. The golden readback compared to the readback of a run did not compare with the error count reported by Vivado itself in the verification file. The verification file makes use of the mask file to determine which bits of the bitstream are essential. As the mask file is not readable and no additional information about it is provided by Xilinx, the readback data will not be used for further analysis. Verification could be used as a black box however as it is included in the Vivado, where readback can not easily be performed.

### 5.1.2. Error modes

Although the benchmark running on the processor did not influence CRAM error rates, Dhrystone did lead to a more representative operation. Every Safe Failure of Dhrystone could only be resolved by reprogramming the FPGA. The differences between errors was the usage of GRMON operations after SF, and the ability of the processor to handle the failure safely. In 2 out of 19 runs the debugger failed to connect to the processor and no additional information could be gathered, thus deemed a FF (fatal in Figure 5.2). A hard reset to allow the FPGA to be reprogrammed again (power cycling the KCU105 board), was needed once, this was the case for one of the FFs.

In 6 out of 19 runs the program exited safely, the last instruction being "*ebreak <tohost\_exit+52>*". This shows that the processor handled an error and exited automatically. This is deemed a SF as discussed in subsection 1.4.3. An unsafe failure occurred when the processor was not able to handle the error itself but could remain communication with the host, this was the case in 10 out of 19 runs. Graphically this is depicted in Figure 5.2.

The other 3 runs are shown in the figure as one SDC and 2 fatal failures. The former being a timing violation and the latter resulting from a lost connection due to JTAG error.

The average fluence to error over 19 runs was  $4.17 \cdot 10^9 \text{ p/cm}^2$ , computed by multiplying test duration by measured flux.

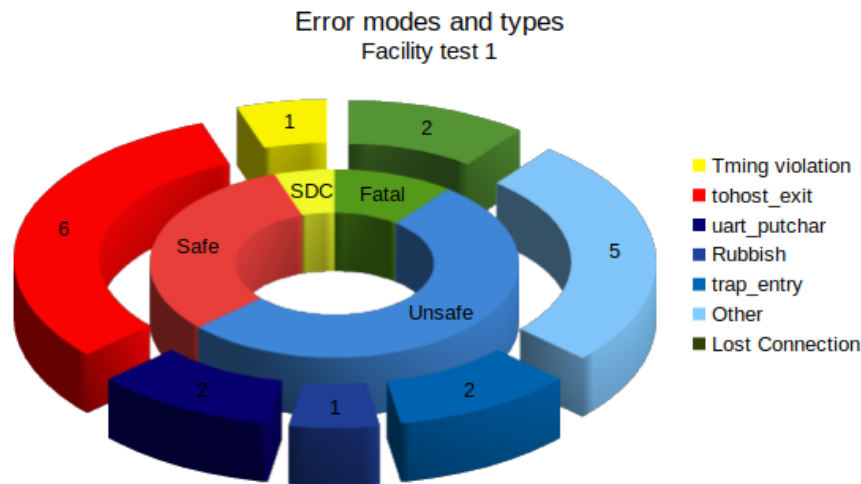


Figure 5.2: Error modes and types grouped.

### 5.1.3. Particle test 1 discussion

The first conclusion from test 1 is the flux to be used during the following tests. The flux level was found to be  $5 \cdot 10^7 \text{ p/cm}^2 \cdot \text{s}$ . This flux is a good trade-off between test time required and statistic relevance.

No conclusions could be drawn from the obtained readback data as the mask file could not be used to identify CRAM critical bits.

The found CRAM cross section of  $2.78 \cdot 10^{-16} \text{ cm}^2/\text{bit}$  is a factor 10 lower opposed to other reported upset rates when evaluating susceptibility of the Kintex Ultrascale FPGA [18, 31]. Where a CRAM cross section of  $1.89 \cdot 10^{-15} \text{ cm}^2/\text{bit}$  and  $2.5 \cdot 10^{-15} \text{ cm}^2/\text{bit}$  are reported for 105 and 64 MeV proton irradiation respectively. A possible reason for this order of magnitude decrease is the extraction method. Aforementioned researchers have checked all CRAM for upsets, where in this thesis the

Vivado Verification tool is used, which already takes into account that some bits are meant to change and will thus not report any upsets in these bits. As aforementioned, the tool will only take into account upsets in the essential bits, which are 10 – 35% of a > 80% full bitstream<sup>1</sup>. In this case the bitstream does not describe that many resources so the share will be on the lower end (configuration used in this test uses ~25% of available resources).

Using the average time to failure and using the FOM technique as described in E.L. Petersen, the average failure rate of the NOEL-V processor in orbit can be calculated [42]. The influence of both proton and heavy-ion irradiation is taken into account in this calculation. The found rate corresponding the 19 Dhrystone tests to failure of the first test is 0.007 *upset/day* for a 51.6 ° circular orbit at 420 km (ISS orbit), employing 5 mm of shielding, leading to a used rate coefficient of 40. This corresponds to values found in [18], where the upset rate is a factor 10 higher but also the used cross section is a factor 10 higher.

This upset rate of the essential CRAM bits is translated to one upset every 143 days in orbit. When looking at functional errors, the same procedure can be used to find that a failure of the processor, running bare-metal Dhrystone, is expected every 669 days. This shows that the processor can be used for in-space operations for short, low Earth orbit missions.

It should be kept in mind that an increase of just 80 km would decrease this time by about two thirds to 267 days. The upset rate is thus highly orbit dependent, therefore the CRAM cross section and FEC will be leading hereafter. Furthermore, the FOM technique makes use of the limiting proton cross section, which is the highest cross section reached by the device, which is at the plateau of the cross section-energy graph. By using this technique it is thus assumed that the limiting cross section was reached at 150 MeV

## 5.2. Particle test 2

The total time tested during test 2 was similar to test 1, in this case 58.5 minutes, in total only 25 seconds less. With the fluxes used in this test this accounted for a total fluence of  $8.93 \cdot 10^{10} \text{ p/cm}^2$ .

As discussed in chapter 4, the second test aimed at getting a better view of the best debug link and energy to use. It also served as a way to gain insight into the possible error mitigation and dCache test methods.

The results of these tests are described below.

### 5.2.1. Total CRAM error rates

As for the first test, CRAM error rates are reported for all examples applicable. In this test this was the case for all but the first 10 runs.

An average cross section of  $2.56 \cdot 10^{-16} \text{ cm}^2/\text{bit}$  is found, the individual cross sections for the runs are shown in Figure 5.3, where it can be seen that the cross section is relatively constant between runs. One run got interrupted by the collimator going into an error state, as the time reported is now unreliable, this measurement was discarded and only 4 runs were performed for the 120 MeV No SEM case.

Differences shown in this figure are highlighted on the right in Figure 5.6.

### 5.2.2. Debug link & Benchmark

As the tests for debug link and benchmark were executed in a combined fashion, with the JTAG debug link tests also being useful as a benchmark test, this section evaluated the outcomes of these tests.

#### Debug link

UART and JTAG debug links were both tested by for their reliability. Running until failure and reconnecting again five times for each link. The result was an average JTAG debug link run time of twice the run time of UART debug link. For the JTAG link the error message had a relation to the debug link at one of the five errors (Debug support unit (DSU) error), whereas the UART debug link failed due to the debug link three times. The errors being twice a Host Target interface (HTIF) error and once a UART communication error. Plots comparing the functional error and CRAM error cross sections of the two debug links are shown in Figure 5.4.

<sup>1</sup>[https://support.xilinx.com/s/question/0D52E00006hpjri/number-of-configuration-bits-used-by-design-in-vivado?language=en\\_US](https://support.xilinx.com/s/question/0D52E00006hpjri/number-of-configuration-bits-used-by-design-in-vivado?language=en_US)

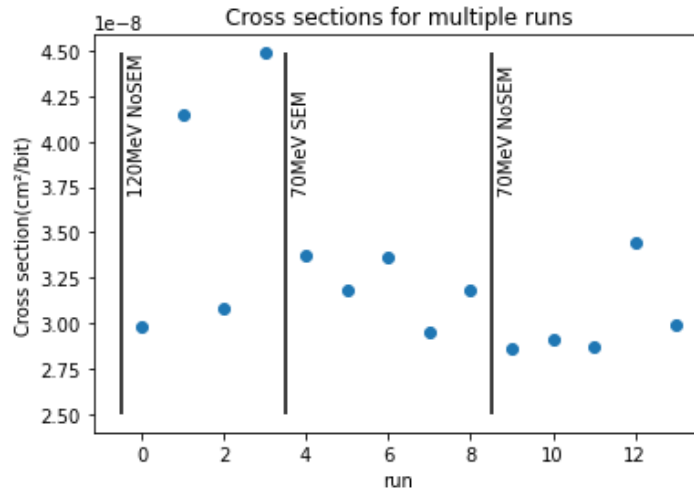


Figure 5.3: Graphical depiction of CRAM cross section during test 2.

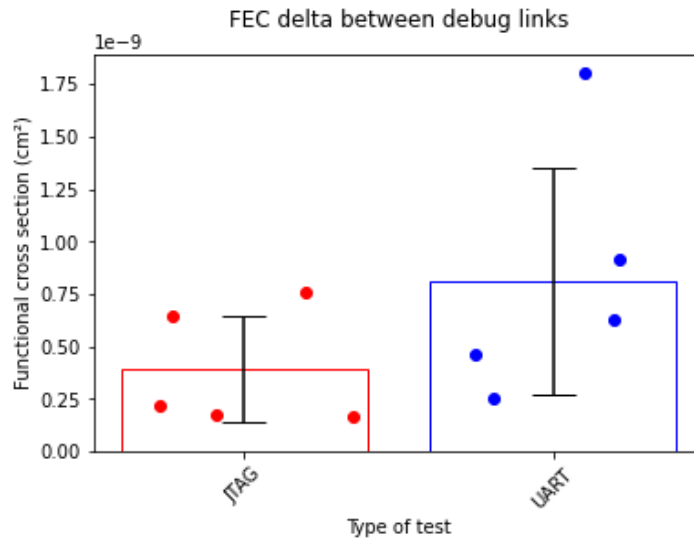


Figure 5.4: FEC of runs for different debug links.

And in absolute values depicted in Table 5.1, where the JTAG measurement is taken as a reference and the delta is computed against this reference.

This result in addition to the slower nature of UART lead to the use of the JTAG debug link for the following tests.

### Benchmark used

During the first test the Dhrystone benchmark was used during 19 runs, during the second test for 14 runs. The CoreMark benchmark was run 5 times using the JTAG debug link during test 2, the 5 times it was ran using UART are not included as all Dhrystone runs also make use of JTAG debug link.

Comparison of FEC for runs running different benchmarks is plotted Figure 5.5, For initial comparison the runs of the first test and second test are kept separated, as there could possibly be an effect of the changing energy/SEM. CRAM cross sections are not plotted here as the CoreMark tests was performed with a unverifiable configuration.

Absolute values are shown in Table 5.2. For the absolute value comparison the Dhrystone runs are joined together, warranted by the small differences observed during the energy and mitigation tests, covered in subsection 5.2.3. Dhrystone runs are taken as the reference.



Table 5.1: Absolute values of the FEC and CRAM cross sections for debug link usage.

	FEC ( $cm^2$ )
JTAG	$3.92 \cdot 10^{-10}$
UART	$8.11 \cdot 10^9$
Delta	0.48x

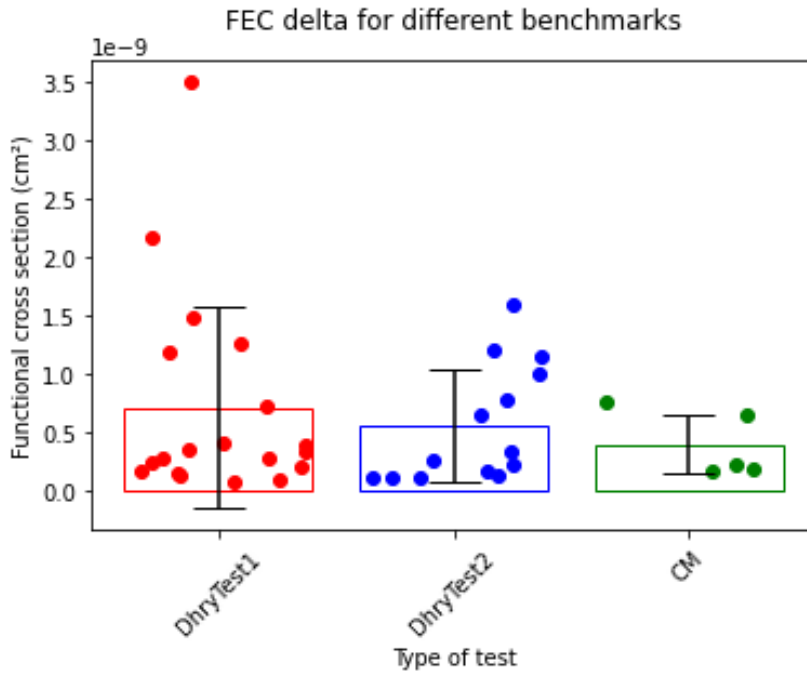


Figure 5.5: FEC for different benchmarks.

### 5.2.3. Error mitigation and Energy

The CRAM error rates of the error mitigation and energy tests are depicted together in Table 5.3. The first column, in this case 120 MeV with SEM cross-sections are taken as the reference, all other tests are compared to this reference and the difference is indicated between brackets.

Table 5.3: CRAM error rates for test with different energies and SEM IP inclusion. 120 MeV SEM data is obtained during the first test.

Error rate(error/s)	120 MeV SEM	120 MeV no SEM	70 MeV SEM	70 MeV no SEM
FEC ( $cm^2$ )	$6.99 \cdot 10^{-10}$	$7.89 \cdot 10^{-10}$ (0.89x)	$4.57 \cdot 10^{-10}$ (1.53x)	$4.57 \cdot 10^{-10}$ (1.53x)
CRAM ( $cm^2/bit$ )	$2.77 \cdot 10^{-16}$	$2.87 \cdot 10^{-16}$ (0.96x)	$2.51 \cdot 10^{-16}$ (1.10x)	$2.36 \cdot 10^{-16}$ (1.18x)

The FEC of all tests is visually depicted and compared to FEC values obtained during the first test in Figure 5.6.

### 5.2.4. dCache

The dCache dynamic test led to no errors. In the 4 minutes that the static dCache test was run a single data corruption occurred, in which 1 out of every 64 bits changed from a 0 to an 8, all in the same position. This meant that out of the 16 KiB dCache, 260 bytes contained an upset. As this is too much for a single particle to have caused, the error is deemed not due to a/multiple bit flips in the dCache, rather an error in the cache controller, possibly being a SEFI.

Table 5.2: Absolute values of the FEC and CRAM cross sections for different benchmarks.

	FEC ( $cm^2$ )	CRAM ( $cm^2/bit$ )
Dhrystone	$6.37 \cdot 10^{-10}$	$2.72 \cdot 10^{-16}$
CoreMark	$3.92 \cdot 10^{-10}$	$2.82 \cdot 10^{-16}$
Delta	1.63x	0.96x

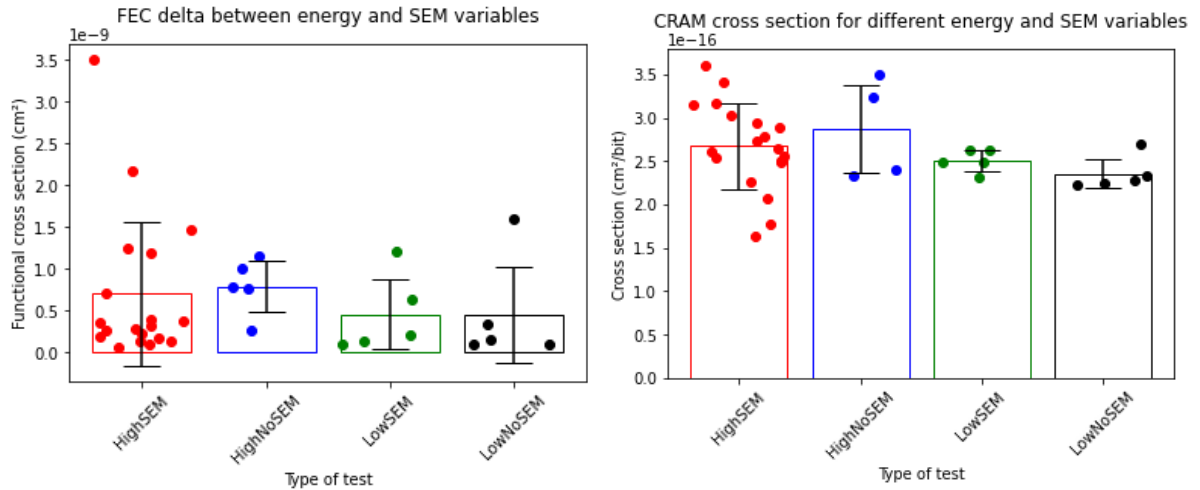


Figure 5.6: Functional error (left) and CRAM cross-section (right) at different energy levels and mitigation measures employed.

### 5.2.5. Error modes

Error modes are depicted in Figure 5.7. Notable differences with the first test is that two runs of Dhrystone fully finished, resulting in no error mode. The rest of the error modes are similar, with the processor taking care of 3 errors, and the processor not managing to exit gracefully in 5 runs. Lost connection runs are unfortunately less helpful, two were due to JTAG connection failure, and are thus not presentable of space operation.

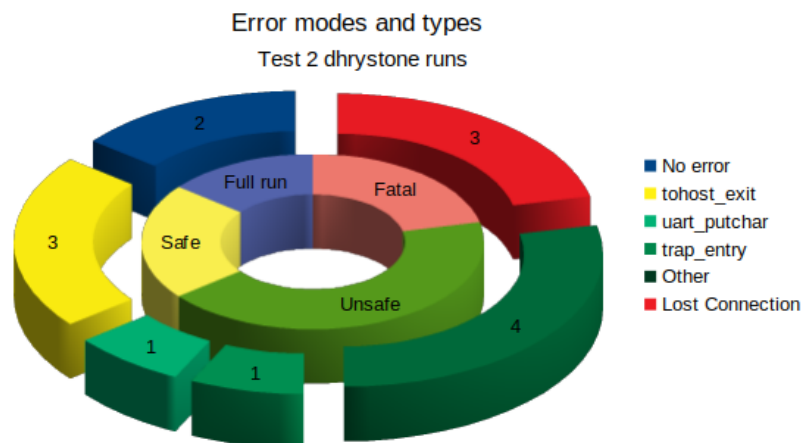


Figure 5.7: Error modes and types grouped.

### 5.2.6. Particle test 2 discussion

The average CRAM cross section observed during test 2 is very similar to the found value during test 1,  $2.56 \cdot 10^{-16} cm^2/bit$  and  $2.78 \cdot 10^{-16} cm^2/bit$  respectively. The small decrease can be accounted to the

use of lower energy particles during part of the test. The found cross section however still compares to the cross section values found in literature.

Looking at Figure 5.6 it is seen that SEM is not enabled on the CRAM. It is interesting to see that the CRAM cross section increases for high energy but decreases for low energy. This is an indicator that SEM is not enabled.

### Energy dependency

With the conclusion that SEM is not present, the four tests can be grouped based on energy. The resulting cross sections are shown in Table 5.4

Table 5.4: Absolute values of the FEC and CRAM cross sections for debug link usage.

	FEC ( $cm^2$ )	CRAM ( $cm^2/bit$ )
70 MeV	$1.81 \cdot 10^{-10}$	$2.43 \cdot 10^{-16}$
120 MeV	$2.74 \cdot 10^{-10}$	$2.70 \cdot 10^{-16}$
Delta	0.66x	0.90x

The small dependency of the CRAM cross section on energy of  $\sim 10\%$  is expected compared to literature, where this data shows that after 70 MeV the plateau is reached. The cross section still rises, but not as quickly as before the "knee" region. This shows that the limiting proton cross section has been reached at 150 MeV and the use of the FOM technique with the cross section is warranted.

A bigger difference is found comparing the FER of the configurations. This indicates that the user logic upsets increase more rapidly with increasing energy than the CRAM upsets.

### Debug link

It is clear from Figure 5.4 that the JTAG debug link is much more reliable compared to the UART debug link. This probably due to it being much slower.

### Benchmark

The plots show that the benchmark employed has no influence on the error rate in CRAM. This is in line with findings in A. Harvard et al. [43]. Where it was also found that Dhrystone and CoreMark lead to a similar amount of sensitive bits.

For the functional error rate however, the cross section for CoreMark is found to be much lower than for Dhrystone. CoreMark thus handles error in the user logic better than Dhrystone, which is unexpected as CoreMark uses more resources, but it has been shown in literature that error masking does depend on the software employed [11]. Found here is thus that CoreMark masks errors more than Dhrystone.

## 5.3. Particle test 3

Effective proton beam time of test 3 was 56.5 minutes, accounting for for a total fluence of  $4.51 \cdot 10^{10} p/cm^2$ .

### 5.3.1. Cache, I/O blocks & ROM memory

#### Caches and I/O blocks

459 seconds of irradiation was performed on the processor when all caches were disabled, at the respective flux levels this means a subjected fluence of  $6.11 \cdot 10^9 p/cm^2$ . During this time, no upsets occurred within the iCache and L2Cache. One bit flip occurred in the dCache, where a 1 was flipped to a 0.

Taking into account the sizes of the caches, being 32 KiB(instruction and data) and 256 KB for the L1cache and L2cache respectively, the error rates can be estimated to be  $2.50 \cdot 10^{-15} err/bit-s$  and  $0 err/bit-s$  respectively.

During the test no invalid operation of the I/O blocks was observed. Such invalid operation could be the failure of a led to keep toggling due to an upset in the I/O block logic or the processor entering

a failure state or the led flickering irregularly indicating a transient.

### ROM and other silent data corruption

During all tests conducted the ROM was monitored, it thus received the same amount of fluence as earlier reported about the entire test. During irradiation, four bit flips in ROM are observed. SDC was also evident due to CoreMark calculations deviating from proper operation, and this being reported by the program, this happened three times. SDC, both in ROM and with CM computations was resolved by the processor in 3 cases, but persisted in the rest of the cases.

### 5.3.2. Influence of operating system

The influence of the employment of RTEMS is shown in Figure 5.8. As it is a comparison between processors running the EX2 configuration, no CRAM cross sections are available.

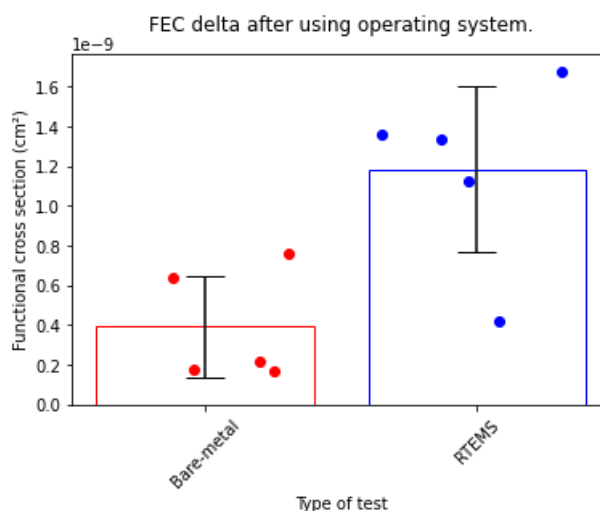


Figure 5.8: Influence of the use of RTEMS.

In absolute values the FEC running bare-metal CoreMark was  $1.18 \cdot 10^{-9} p/cm^2$  and when running CoreMark in RTEMS was  $0.39 \cdot 10^{-9} p/cm^2$ , showing a 3x improvement in cross-section by using JTAG link.

### 5.3.3. Error modes

A visual depiction of the error modes occurring during the third test are depicted in Figure 5.9.

The most common fatal source was RTEMS\_FATAL\_SOURCE\_EXCEPTION. Together with this error handle registers are printed which show the pointer value of the exception frame pointer.

A breakdown between the different configurations is shown in Figure 5.10.

### 5.3.4. Cross sections of different configurations

All different configurations and tests combinations were aimed to be tested five times to failure, with the exception of FPU test, which were performed four times.

#### Cross sections

CRAM cross sections for all applicable runs and grouped by configuration and test are shown in Figure 5.11. Absolute values, together with functional error cross sections are shown in Table 5.5. In Figure 5.11, tiny configuration without and with the inclusion of SEM IP are plotted (TINYN and TINYS respectively). It is here once again confirmed that the SEM IP is not included in the default version of NOEL-V. The cross sections of these individual tests are combined for the further analysis of the data.

Easy comparison between tests to see the influence of the floating point unit can be made by viewing plots of the different configuration and test combinations, which are shown below.

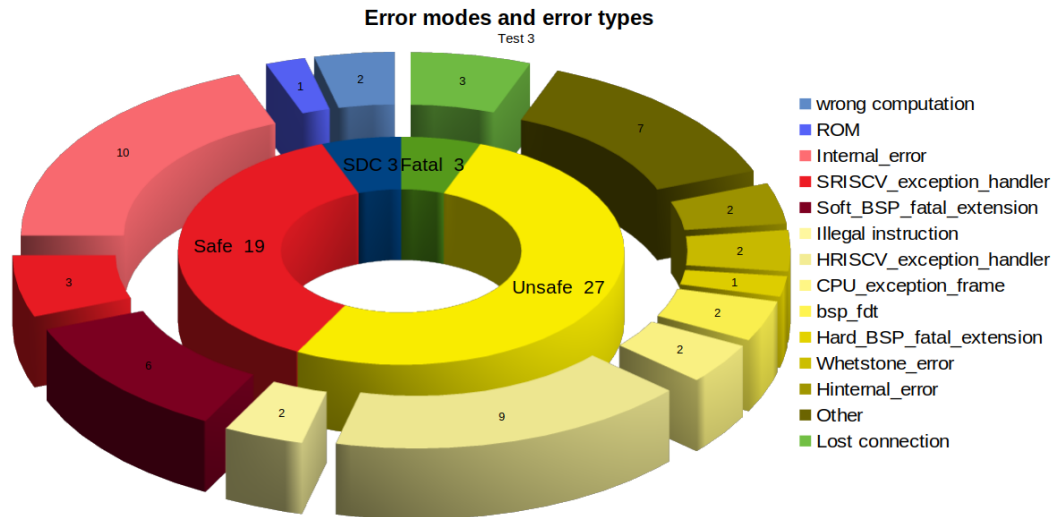


Figure 5.9: Error modes grouped.

Table 5.5: Soft error rates for the different configurations running different tests. All INT level comparison are compared to Tiny configuration, for FPU level the comparison is made with the Single configuration running this test.

Configuration	Level	CRAM $\sigma$ ( $cm^2/bit$ )	FEC ( $cm^2$ )
Tiny	INT	$7.99 \cdot 10^{-16}$ (-)	$1.33 \cdot 10^{-9}$ (-)
EX1	INT	-	$2.08 \cdot 10^{-9}$ (1.56x)
Single	INT	$6.16 \cdot 10^{-16}$ (0.77x)	$1.99 \cdot 10^{-9}$ (1.49x)
	FPU	$7.65 \cdot 10^{-16}$ (-)	$1.31 \cdot 10^{-9}$ (-)
Dual	INT	$8.82 \cdot 10^{-16}$ (1.10x)	$2.75 \cdot 10^{-9}$ (2.07x)
	FPU	$7.14 \cdot 10^{-16}$ (0.93x)	$1.16 \cdot 10^{-9}$ (0.89x)
EX2	INT	-	$9.35 \cdot 10^{-10}$ (0.70x)
	FPU	-	$1.07 \cdot 10^{-9}$ (0.82x)

### 5.3.5. Particle test 3 discussion

The overall error modes do not differ much from earlier tests, the relative amounts of the different error modes are similar. Within the error modes of the different tests notable differences exist.

It can be seen that the configurations employing MMU and FPU show considerable amounts of safe failures, whereas the other configurations (Tiny and EX1) show almost no SF but mostly FF. Comparing the tests running INT with the tests running FPU of these configurations it can be seen that the FPU is not of influence for the relative amount of SF, thus it must be resulting from the inclusion of the MMU.

#### Caches and ROM

Cross sections of the Cache and ROM memory show that mitigation measures will be necessary. Proton cross section for the L1cache being  $2.50 \cdot 10^{-15} \text{ err/bit}$ , compares to literature values for the CRAM cross section, but is a factor 10 higher than found CRAM cross section values in this research. This, again, due to the exclusion of non-essential bits by the Vivado verification tool. It shows that susceptibility of the caches can be compared to the susceptibility of the logic configuration SRAM.

It does mean, however, that the L1cache (and thus the L2cache) needs some kind of mitigation if no upsets, possibly leading to slower operation or to faulty computations, are tolerated by the mission. This is also the case for the ROM memory, where four upsets occurred during the irradiation time. With a size of 512 MB this leads to a cross section of  $3.9 \cdot 10^{19} \text{ err/bit}$ , which is much higher than the found values for caches and CRAM.

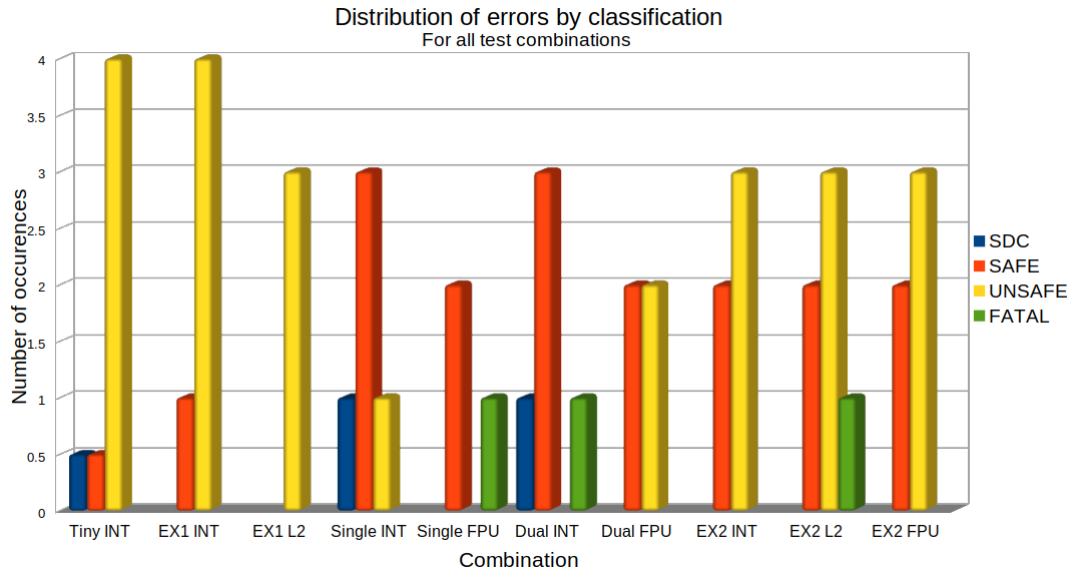


Figure 5.10: Error modes for the different conducted test. As 10 tests were conducted using the Tiny configuration INT test these values have been normalised by dividing the values by 2.

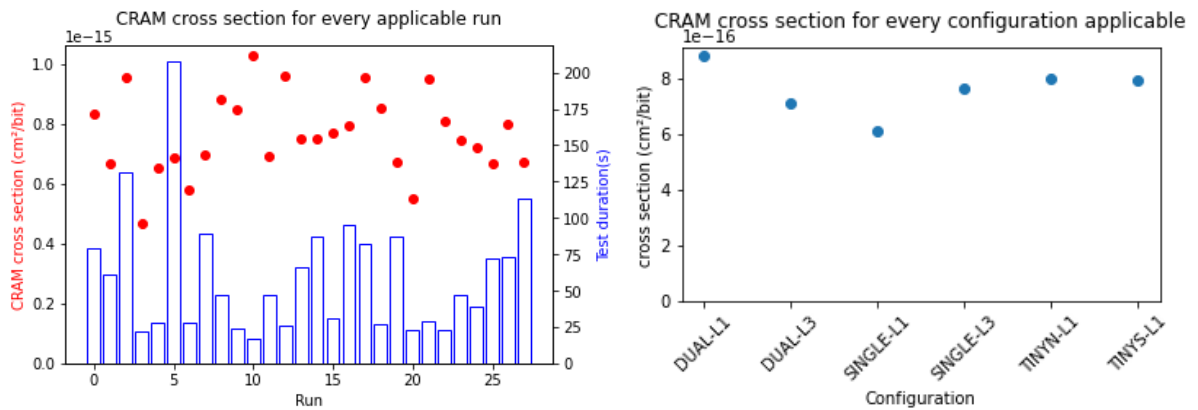


Figure 5.11: CRAM error rate and run duration for every run during test 3 (left) and average CRAM error rate for every combination (right).

**Differences between configurations/vulnerable parts**

Differences between the processor in different configurations running test 1 are depicted in Table 5.6. It has to be taken into account that the SER and FER values are highly orbit specific, with the orbit used here being the ISS orbit. A small alteration to the orbit can have substantial consequences for these rates. CoreMark allows several upsets to be set, use was made of the same set of options as used by Cobham Gaisler to increase comparison<sup>2</sup>

It can be seen that the expected failures are much sooner now than for the first test, this is mostly due to the FER increase due to use of RTEMS.

As described, it is expected that due to accumulating more CRAM errors, the CRAM upset rates of larger area configurations are higher. However, looking at the results this is not true. This indicates that the amount of essential bits in the bitstream does not linearly increase with resource usage. The inclusion of the dual-issue pipeline seems to be increasing the amount of essential bits.

In general, excluding EX2, the more performance the configuration has, the lower the FER. This

<sup>2</sup>Options set: HPP = high performance processor, GPP = general purpose processor. Coremark score generated -march=rv64im -mabi=lp64 -O2 -funroll-all-loops -funswitch-loops -fgcse-after-reload -fpredictive-commoning -mtune=sifive-7-series -finline-functions -fipa-cp-clone -falign-functions=8 -falign-loops=8 -falign-jumps=8 -param max-inline-insns-auto=20 using GCC 9.2.0 under RTEMS 5.

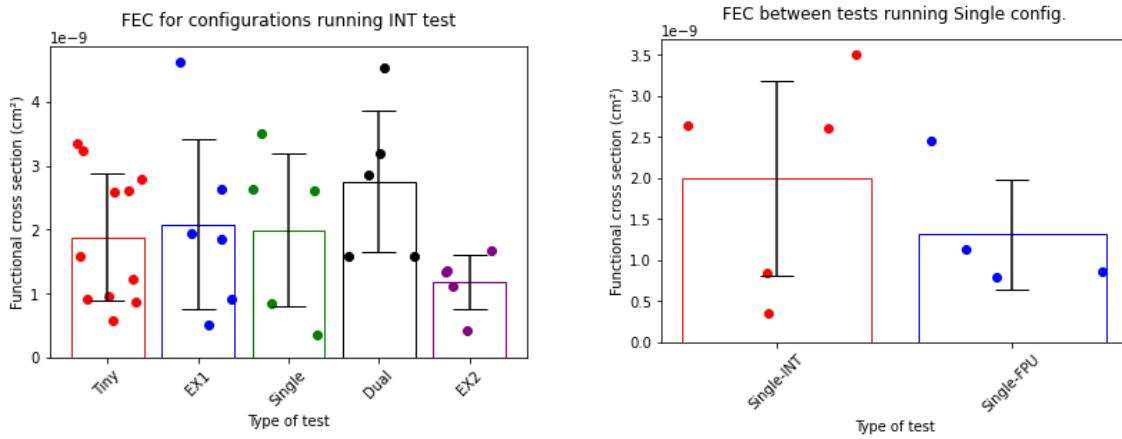


Figure 5.12: Functional error cross-section for all configurations running INT software (left) and on the right the single configuration running different software levels

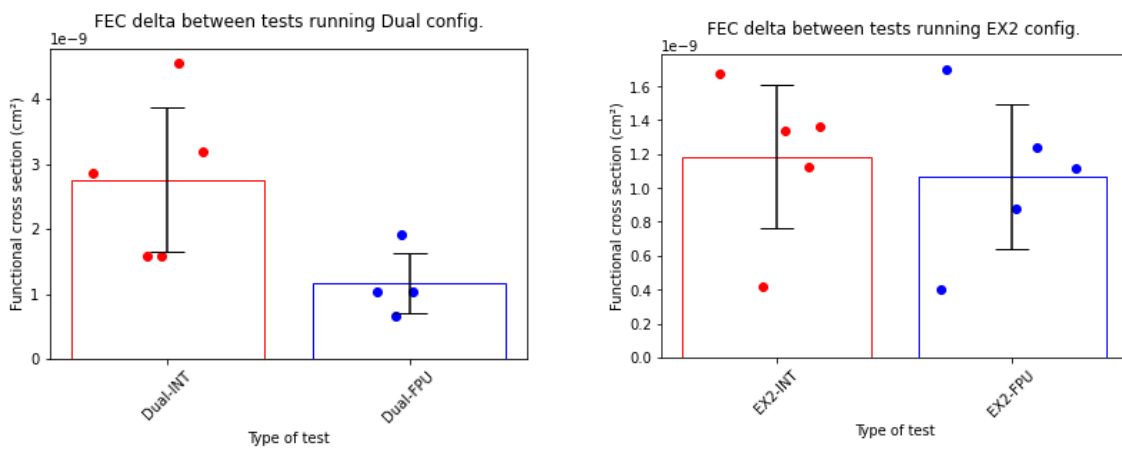


Figure 5.13: Functional error cross-section for INT and FPU software running on Dual (left) and EX2 (right) configurations

difference is marginal and does not warrant the decrease in performance available when the added area is not limiting.

To get a good grasp of performance increase of the processor in the user logic, the FEC will be scaled by the area of the respective configuration, thereby taking into account the expected effect increased resource usage will have, a plot is shown in Figure 5.14.

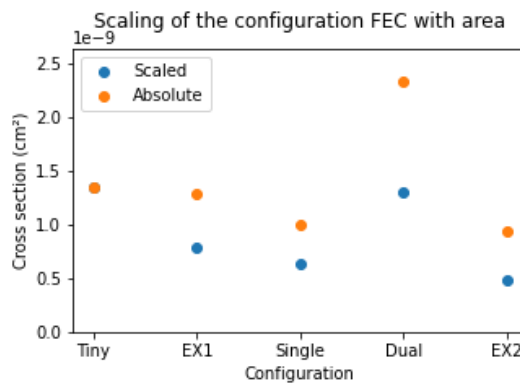


Figure 5.14: FEC for all configurations running INT software, absolute and scaled by area overhead.

Table 5.6: Comparison between the Tiny, Minimal, Single issue, Dual issue, and high performance Dual issue processor configuration based on radiation susceptibility (based on INT data), CM score, and overhead. All comparison values are compared to the Tiny configuration.

Configuration	SER/FER (days to upset)	CM Score ( $^{CM}/MHz$ )	Area%	Area overhead
Tiny	56.8 / 248.3	1.63 (-)	LUT: 15.98% FF: 7.41% BRAM: 6.08% DSP: 0.99%	-
Minimal(EX1)	-/ 224.6 (0.91x)	3.01 (1.85x)	LUT: 16.25% FF: 5.48% BRAM: 17.33% DSP: 0.83%	1.14x 0.78x 2.97x 0.84x
GPP(single issue)	73.7 (1.30x) / 234.6 (0.94x)	3.05 (1.87x)	LUT: 22.44% FF: 8.62% BRAM: 9.33% DSP: 1.09%	1.57x 1.23x 1.60x 1.10x
GPP(dual issue)	51.5 (0.91x) / 170.0 (0.69x)	4.38 (2.69x)	LUT: 25.83% FF: 8.92% BRAM: 10.33% DSP: 1.09%	1.81x 1.28x 1.77x 1.10x
HPP(EX2)	-/ 395.2 (1.59x)	4.46 (2.74x)	LUT: 20.25% FF: 6.10 % BRAM: 17.83% DSP: 0.94%	1.42x 0.87x 3.06x 0.95x

It can be seen that, the larger configurations show lower susceptibility compared to the Tiny configuration when normalised by their resource utilisation. Unfortunately, the differences between configurations do not lead to conclusive answers about the susceptibility of processor parts.

The minimal configuration shows higher susceptibility opposed to the tiny configuration. This could be due to the inclusion of the L2Cache, HPP-to-HPP bridge, increase in L1Cache or PMP. The higher speed can also be a possible reason due to increased usage of functional units. It is shown in [11] that the more frequently functional units are used, the more susceptible they are to radiation.

The slightly decreased susceptibility of the single configuration compared to the minimal configuration cannot be attributed to a speed increase, as there is none as shown in Table 5.6. The inclusion of the MMU, the FPU and a larger L1cache and the omission of the L2Cache being the most notable differences between these configurations. In this software layer the FPU is not used, and the increase in L1Cache leads to a higher susceptibility [11]. The conclusion is thus that the L2Cache makes the minimal configuration more susceptible and the MIG, which more often used by the single configuration, is less susceptible.

A surprising finding, which can also be seen very well in Figure 5.11, is the very high cross section of the Dual configuration. This would indicate high susceptibility of the dual-issue processor, possibly stemming from the inclusion of the dual-issue pipeline or the FPU. The former is unlikely as this would also mean that the EX2 configuration should show high susceptibility. The latter is nullified by the Single & EX2 configurations showing a very low cross-section and the cross section difference between single  $7.65 \cdot 10^{-16} \text{ err/bit}$  and dual ( $7.14 \cdot 10^{-16} \text{ err/bit}$ ) configurations running the FPU test, being a 1.07x decrease in cross section as opposed to the increase for INT level.

The speed increase (due to the inclusion of a dual issue processor) is the most likely reason for the excessive susceptibility of the dual issue configuration. However, this does not show at all when running the FPU software level.

The difference between the FEC of the applicable configurations running the FPU software is shown in more detail in Figure 5.15



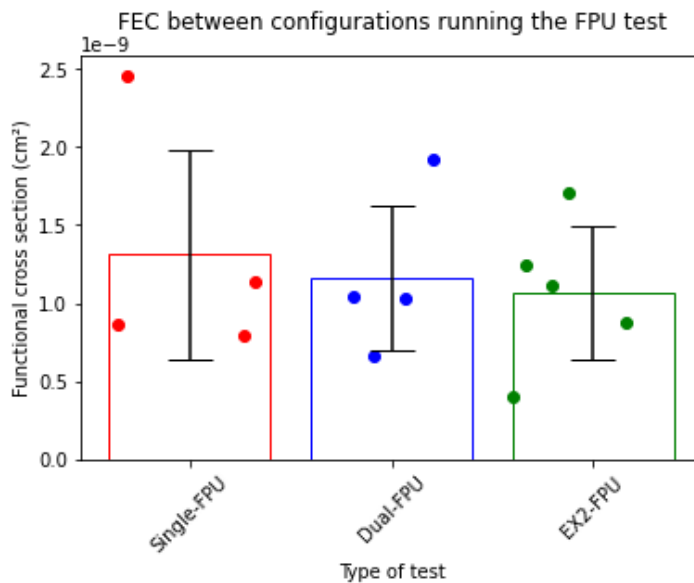


Figure 5.15: Differences in FEC for applicable configurations running FPU software.

With exact values being  $1.31 \cdot 10^{-9} \text{ cm}^2$ ,  $1.16 \cdot 10^{-9} \text{ cm}^2$  (0.88x) and  $1.07 \cdot 10^{-9} \text{ cm}^2$  (0.82x) for the single, dual and EX2 configurations respectively and compared to single.

It can be seen that the same decrease in susceptibility is observed for the EX2 configuration compared to the Single configuration. But the Dual configuration shows the opposite compared to when running the INT software.

For the EX2 configuration the increase in susceptibility by the dual-issue pipeline could be masked by a decrease in susceptibility in other parts of the processor. However, decreases due to for example inclusion of the L2cache, or HPP-to-HPP bridge would then also be present in the EX1 configuration cross section.

The most likely reason for the decrease is the addition of the HPP-to-HPP bridge and the dual-issue pipeline, but without the increase in L1cache size. It is evident that the EX2 configuration has the lowest failure rate of all processor configurations. As this is evident during both INT and FPU tests it is more likely that the dual-int data is faulty as Figure 5.15 also suggests this.

### Floating point unit

A detailed comparison between the The inclusion of the FPU decreases the cross section for all configurations. This is an indication that the FPU is much less susceptible to error than the integer unit.

This found decrease in susceptibility when exercising the floating point unit over the integer unit is in line with expectations. As discussed in chapter 1, the FPR is expected to be less vulnerable than the GPR and this is thus (part) of the reason, but other stages of the FPU can also be of influence.

Another possible explanation is the ability of the FPU to generate interrupts. As noted in [39], the only exception that can be thrown by the FPU is a faulty memory address exception.



# 6

## Conclusions and recommendations

In this chapter, conclusions are drawn based on the found data, and will be used to answer the research questions. Afterward, a discussion is included about potential limitations of the tests conducted and further steps to complete the radiation characterisation of the NOEL-V processor in the space environment.

### 6.1. Conclusions

From the discussions after the tests in chapter 5, a number of conclusions can be drawn. These are summarised below and thereafter used to answer the research questions.

#### 6.1.1. Conclusions

The most prominent conclusions that can be drawn from the particle tests are the use of debug link and the huge negative influence of the kernel used. Furthermore, the reduced susceptibility of the FPU compared to the integer unit states the findings in other research and has expanded the conclusion that not only is the FPR less susceptible than the GPR, but also the FPU as a whole shows decreased susceptibility.

Unfortunately, the error rate differences between different configurations did not lead to a conclusive answer on which part(s) of the processor have relatively high susceptibility, except for the aforementioned FPU. It seems that the inclusion of L2Cache does lead to higher susceptibility due to the increase in vulnerable memory and the resulting reduced use of the MIG. The MIG, contrary to that, is believed to be less vulnerable as the values it retrieves are not located in the beam area.

Employing a dual-issue is deemed beneficial for resilience to radiation effects, as is shown in three out of four comparisons with a single-issue processor. The remaining case, where the single-issue processor performs better, is deemed an anomaly.

The configuration one wants to use depends highly on mission requirements, but this research has proven that radiation susceptibility only increases slightly with processor resource usage, or not at all for configuration EX2. When resource usage does not limit one's ability to choose a better performing processor, a better performing one is always the superior choice.

In general, the radiation susceptibility between configurations does not differ much with microarchitectural differences, all being in the same order of magnitude for identical operating conditions.

#### 6.1.2. Research questions

*Is a (modified) version of the NOEL-V soft processor suited for space applications and what changes would lead to the highest improvement of the radiation susceptibility at the lowest cost?*

Using the earlier described FOM method it can be shown that the NOEL-V processor can be made suitable for in-space operations. Cache upsets are shown to occur but with the use of Error Detection And Correction(EDAC) methods such upsets can be avoided. Since no MBU occurred, EDAC methods with small overhead can be used.

It can, however, be seen that none of the configurations without any mitigation measures will last a satellite's lifetime. It is evident that fault tolerant measures should be taken. Without any area constraints, the user should always choose the best processor available as the susceptibility does not increase linearly with area used. The masking of the processor increases as the complexity increases.

Due to EDAC codes not posing an overhead on performance (if the cache can be enlarged), this would be a low cost alteration to improve the processor. Furthermore, the integer unit would be an improvement that would be more cost effective than the FPU.

During this research the sub-questions have also been answered:

#### **How to most ideally mimic in-space operation?**

(a) *What software can be used to mimic in-space operation?*

Satellites will be used to perform numerous different tasks so no definite answer can be given on which software best mimics space. dCache usage differs greatly between Dhrystone and Coremark. Dhrystone best mimics low performance operations, whereas CoreMark better mimics high performance operations. It is also shown that CoreMark handles user upsets better, showing that more complexity does not automatically mean an increase in user logic upsets.

(b) *What particles are best used?*

Out of the available energies, it is best to use the 150 MeV protons. As the energy has only marginal impact on cross section, a higher energy will result in less accumulation of ionising radiation [14].

Proton and heavy-ion tests both have their advantages and disadvantages. Statistical methods exist to use one of the two to estimate the contribution of the other, like the FOM method [42]. Choices on the extensiveness of the radiation characterisation will determine whether characterisation using both is needed or if such a conversion between the two is sufficient.

If only one test is sufficient, a proton test is beneficial over the heavy-ion test. The proton test is generally less expensive (material removal costs \$3.5k [44]) and the availability is higher. During heavy-ion testing, part of the PCB needs to be removed as the penetration depth is not sufficient. This is not the case for high-energy protons.

(c) *What duration should the test take?*

Test duration to achieve statistical relevance without being excessive determined the upper bound of the test time to be five minutes. This is achieved by using a flux that leads to around 1 CRAM error per second.

#### **How to extract information from the DUT?**

(a) *CRAM*

CRAM upset rates can best be extracted using Xilinx developed readback and verification operations. However, one should not use readback before verification, as it will render the verification useless with the introduction of numerous differences.

(b) *Cache memory*

Data from cache memories can be statically extracted using GRMON commands or loading and reading predefined data. The best way of extracting data has been proven to be to use GRMON commands.

(c) *I/O blocks*

The I/O blocks can individually be tested by operating them and checking if their operation is correct.

(d) *ROM*

The contents of ROM will be constant during operation, therefore values during/after irradiation can be compared to unupset data. Any changes in this data can be attributed to radiation effects.

(e) *Floating Point Unit*

Upset data from the FPU can be extracted using a benchmark that stresses operation of the FPU. Any deviations from the operation with the integer pipeline can be attributed to the FPU if the rest of the test is performed under identical circumstances.

#### **What parts are most vulnerable?**

(a) *How to best test per-part susceptibility?*

The best way to test per-part susceptibility is to use upset data on individual parts together with

upset data on the configurations as a whole. Using different software programs, additional information can be obtained. Comparing the upset and failure rates of all these different tests, certain parts can be isolated to show its individual vulnerability.

(b) *What is the cost of applying mitigation to a specific component?*

A definite answer to the question of the cost of mitigation can not be given due to the inconclusiveness of the per-part susceptibility. One can, however, look at the area of each part of the processor.

Keeping in mind that memories can be made fault tolerant by employing information redundancy. The cost of fault tolerance on memories will be less than for hardware redundancy, which will have to be applied to make microarchitectural elements such as FFs or ALUs fault tolerant.

### **Which configuration of the NOEL-V processor is best suited for specific tasks?**

Which configuration to use depends on a lot of characteristics about the requirements of the design. One certainty is that mitigation measures are needed to make any configuration viable for space operations, where not all mitigation measures are equal on overhead. It is shown that the highest performance configuration is also the most reliable configuration, thus being the optimal choice for both applications requiring dependability and applications requiring high performance. Lower performance configurations will be beneficial when area constraints are leading. Unfortunately, precise data about the resource usage of the example configurations is not available but it can be assumed that the Tiny configuration has the smallest area, therefore, this being the best choice for low to middle end microcontroller applications.

A framework on how to go about such fault tolerance adaptations has been proposed in [11]. More extensive research is needed to get a definite answer to the question about which fault tolerant measures are most beneficial to the NOEL-V, and these will also be FPGA specific.

It is proven that the caches are prone to upsets, and are, therefore, important to apply mitigation on. Mitigation in memory can be done by applying EDAC code, thus requiring no architectural changes but information redundancy instead. This leads to no penalties in speed as would be present when hardware redundancy is applied. It is shown that the caches are of importance to the speed of the processor (measured in CM score).

One has to keep in mind that mitigation measures will be more impactful on the bigger (area wise) configurations. The trade-off will thus shift towards the smaller (Tiny, EX1) configurations after mitigation has been applied and area overhead is a requirement.

## **6.2. Recommendations**

An interesting observation is that the irradiation time throughout the three tests is very similar. Although the time is only 4% less for the third test compared to the second, the total fluence is almost half. As known, the fluence data of test 3 are given with a possible error of  $\sim 10\%$ , so this difference lays beneath that. Investigation shows that this is due to a lower flux used to compute the test three fluence. This flux is about two third of that used during the second test, but only for the runs until the energy is changed. After the energy was changed back to  $150 \text{ MeV}$ , the reported flux is in the range of the now used flux to compute fluence values for test 3 (also the same intensity is used). Therefore, it was chosen not to calculate using the average flux of test two ( $2.18 \cdot 10^7 \text{ p/cm}^2 \cdot \text{s}$ ), but the calibration flux of  $1.33 \cdot 10^7 \text{ p/cm}^2 \cdot \text{s}$ .

The data could also benefit from updated and correct values for the resource utilisation of the example configurations. Using this data, a better trade-off can be made and a better comparison between the configurations can be performed.

In space, ionising protons will not be the only threat to the correct operation of the FPGA. Further investigation will be necessary with respect to, for instance, heavy-ions. The estimation of the time to upset performed here using [42] does try to take this into account, but is based on statistics. For proper characterisation, a heavy-ion test should be performed. It is known that high LET heavy-ion events are more prone to induce multiple bit upsets as opposed to protons and electrons, due to the high LET [14].

The processor is promising for use in space, however some mitigation measures should be taken to ensure this. The use of fault tolerant techniques will have to be investigated further. Inclusion of triple modular redundancy on the sequential elements can make sure no upsets occur as has been

observed for the unmitigated processor.

# Bibliography

- [1] Stefano Di Mascio et al. “Leveraging the Openness and Modularity of RISC-V in Space”. In: *Journal of Aerospace Information Systems* 16.11 (2019), pp. 454–472.
- [2] Hyungmin Cho. “Impact of microarchitectural differences of RISC-V processor cores on soft error effects”. In: *IEEE Access* 6 (2018), pp. 41302–41313.
- [3] Gianluca Furano and Alessandra Menicucci. “Roadmap for on-board processing and data handling systems in space”. In: *Dependable Multicore Architectures at Nanoscale*. Springer, 2018, pp. 253–281.
- [4] Michel Pignol. “COTS-based applications in space avionics”. In: *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*. IEEE. 2010, pp. 1213–1219.
- [5] Bruce Yost et al. “State-of-the-Art Small Spacecraft Technology”. In: (2021).
- [6] A Fernández León. “Trends and patterns of ASIC and FPGA use in European space missions”. PhD thesis. MS thesis, Delft Univ. of Technol., Delft, Netherlands, 2013.[Online ... , 2013.
- [7] Stefano Di Mascio et al. “The case for RISC-V in space”. In: *International Conference on Applications in Electronics Pervading Industry, Environment and Society*. Springer. 2018, pp. 319–325.
- [8] Krste Asanović and David A Patterson. “Instruction sets should be free: The case for risc-v”. In: *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146* (2014).
- [9] Xilinx. *UltraScale Architecture and Product Data Sheet: Overview*. 2021.
- [10] David S. Lee et al. “Single-Event Characterization of the 20 nm Xilinx Kintex UltraScale Field-Programmable Gate Array under Heavy Ion Irradiation”. In: *2015 IEEE Radiation Effects Data Workshop (REDW)*. 2015, pp. 1–6. DOI: 10.1109/REDW.2015.7336736.
- [11] Stefano Di Mascio et al. “Open-source IP cores for space: A processor-level perspective on soft errors in the RISC-V era”. In: *Computer Science Review* 39 (2021), p. 100349. ISSN: 1574-0137. DOI: <https://doi.org/10.1016/j.cosrev.2020.100349>. URL: <https://www.sciencedirect.com/science/article/pii/S1574013720304494>.
- [12] E. G. Stassinopoulos and J. P. Raymond. “The space radiation environment for electronics”. In: *Proceedings of the IEEE* 76.11 (1988), pp. 1423–1442. DOI: 10.1109/5.90113.
- [13] Robert C Baumann. “Radiation-induced soft errors in advanced semiconductor technologies”. In: *IEEE Transactions on Device and materials reliability* 5.3 (2005), pp. 305–316.
- [14] R. Baumann and Kirby Kruckmeyer. *Radiation handbook for electronics*. Jan. 2019.
- [15] T.S. Nidhin et al. “Understanding radiation effects in SRAM-based field programmable gate arrays for implementing instrumentation and control systems of nuclear power plants”. In: *Nuclear Engineering and Technology* 49.8 (2017), pp. 1589–1599. ISSN: 1738-5733. DOI: <https://doi.org/10.1016/j.net.2017.09.002>. URL: <http://www.sciencedirect.com/science/article/pii/S1738573317302723>.
- [16] Sophie Duzellier. “Radiation effects on electronic devices in space.” In: (2004).
- [17] Andrés Pérez-Celis, Corbin Thurlow, and Michael Wirthlin. “Identifying Radiation-Induced micro-SEFIs in SRAM FPGAs”. In: *IEEE Transactions on Nuclear Science* (2021).
- [18] David M. Hiemstra, Valeri Kirischian, and Jakub Brejski. “Single Event Upset Characterization of the Kintex UltraScale Field Programmable Gate Array Using Proton Irradiation”. In: *2016 IEEE Radiation Effects Data Workshop (REDW)*. 2016, pp. 1–5. DOI: 10.1109/NSREC.2016.7891743.

- [19] David S. Lee, Gary Swift, and Michael Wirthlin. "An Analysis of High-Current Events Observed on Xilinx 7-Series and Ultrascale Field-Programmable Gate Arrays". In: *2016 IEEE Radiation Effects Data Workshop (REDW)*. 2016, pp. 1–5. DOI: 10.1109/NSREC.2016.7891703.
- [20] L.Z.Scheick L.D.Edmonds C.E. Barnes. *An introduction to space radiation effects on microelectronics*. Pasadena, California: JPL, 2000.
- [21] Athanasios Chatzidimitriou et al. "Demystifying soft error assessment strategies on arm cpus: Microarchitectural fault injection vs. neutron beam experiments". In: *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2019, pp. 26–38.
- [22] JEDEC Government Liaison Committee et al. *TEST STANDARD FOR THE MEASUREMENT OF PROTON RADIATION SINGLE EVENT EFFECTS IN ELECTRONIC DEVICES*. JESD234. 2013.
- [23] JEDEC Government Liaison Committee et al. *Test Procedure for the Management of Single-Event Effects in Semiconductor Devices from Heavy Ion Irradiation*. JESD57. 2017.
- [24] Vahid Jamshidi. "NVRH-LUT: A nonvolatile radiation-hardened hybrid MTJ/CMOS-based look-up table for ultralow power and highly reliable FPGA designs". In: *Turkish Journal of Electrical Engineering and Computer Sciences* 27 (Nov. 2019), pp. 4486–4501. DOI: 10.3906/elk-1812-179.
- [25] Carl Carmichael et al. "Proton testing of SEU mitigation methods for the Virtex FPGA". In: *Proc. of Military and Aerospace Applications of Programmable Logic Devices MAPLD* (2001).
- [26] Aaron Gerald Stoddard. "Configuration scrubbing architectures for high-reliability FPGA systems". In: (2015).
- [27] Xilinx. *Xilinx XAPP532 - Soft Error Mitigation Using Prioritized Essential Bits*. 2012.
- [28] D. Curd and E. Crabill. *UltraScale Devices Maximize Design Integrity with Industry-Leading SEU Resilience and Mitigation*. 2015.
- [29] Xilinx. *SEM Error Mitigation Controller v4.1*. 2018.
- [30] Luca Sterpone Marco Desogus. *Analysis and Mitigation of SEUs on SRAM-based FPGAs using the VERI-Place tool*. Powerpoint.
- [31] Pierre Maillard et al. "Neutron, 64 MeV Proton, Thermal Neutron and Alpha Single-Event Upset Characterization of Xilinx 20nm UltraScale Kintex FPGA". In: *2015 IEEE Radiation Effects Data Workshop (REDW)*. 2015, pp. 1–5. DOI: 10.1109/REDW.2015.7336723.
- [32] Luis Alberto Aranda et al. "Analysis of the Critical Bits of a RISC-V Processor Implemented in an SRAM-Based FPGA for Space Applications". In: *Electronics* 9.1 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9010175. URL: <https://www.mdpi.com/2079-9292/9/1/175>.
- [33] Andrew Elbert Wilson and Michael Wirthlin. "Neutron Radiation Testing of Fault Tolerant RISC-V Soft Processor on Xilinx SRAM-based FPGAs". In: *2019 IEEE Space Computing Conference (SCC)*. 2019, pp. 25–32. DOI: 10.1109/SpaceComp.2019.00008.
- [34] David M Hiemstra and Valeri Kirischian. "Part II: Single Event Upset Characterization of the Kintex UltraScale Field Programmable Gate Array using Proton Irradiation". In: *2018 IEEE Radiation Effects Data Workshop (REDW)*. IEEE. 2018, pp. 1–4.
- [35] Xilinx. *KCU105 Board - User Guide UG917*. 2019.
- [36] Yu Cheng et al. "Accurate vulnerability estimation for cache hierarchy". In: *The 7th International Conference on Networked Computing and Advanced Information Management*. IEEE. 2011, pp. 7–14.
- [37] Chen-Yong Cher et al. "Understanding Soft Error Resiliency of Blue Gene/Q Compute Chip through Hardware Proton Irradiation and Software Fault Injection". In: *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2014, pp. 587–596. DOI: 10.1109/SC.2014.53.
- [38] Cobham Gailser AB. *GRMON3 User's Manual*. 2021.
- [39] Cobham Gailser AB. *GRLIB IP Core User's Manual*. 2021.
- [40] Cobham Gailser AB. *NOEL-XCKU-EX RISC-V Processor User's Manual*. 2021.



- 
- [41] Brian Randell and Lawford John Russell. "ALGOL 60 implementation: the translation and use of ALGOL 60 programs on a computer". In: *APIC Studies in Data Processing* (1964).
- [42] E. L. Petersen. "The SEU figure of merit and proton upset rate calculations". In: *IEEE Transactions on Nuclear Science* 45.6 (1998), pp. 2550–2562.
- [43] Nathan A. Harward et al. "Estimating Soft Processor Soft Error Sensitivity through Fault Injection". In: *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*. 2015, pp. 143–150. DOI: 10.1109/FCCM.2015.61.
- [44] Kenneth Label. *The mystery talk: Single Event Effect (SEE) Test Costs AND Selected GSFC Radiation Testing for NEPP/DTRA\**. 2006.