UNIVERSITY OF PIRAEUS

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATICS

Ph.D. Thesis

Commercial Off-The-Shelf Programmable System-on-Chip Devices for Space Applications

Εμπορικά Διαθέσιμες Συσκευές Προγραμματιζόμενου Συστήματος-σε-chip για Διαστημικές Εφαρμογές

Student's name-surname:	Vassilis Vlagkoulis
Father's name:	Dionissis
Student's ID No:	ΠΛΔ1612
Supervisor:	Mihalis Psarakis, Professor, University of Piraeus

July 2026

Ph.D. Thesis
was prepared during the Programme of Doctoral Studies
of the Department of Informatics
of the School of Information and Communication Technologies
of the University of Piraeus
for the degree of Doctor of Philosophy

3-Member Counseling Committee

Mihalis Psarakis	**Antonios Paschalis**	**Cristiana Bolchini**
Professor	Professor	Professor
University of Piraeus	National and Kapodistrian University of Athens	Politecnico di Milano
School of Information and Communication Technologies	School of Science	School of Industrial and Information Engineering
Department of Informatics	Department of Informatics and Telecommunications	Department of Electronics, Information and Bioengineering
(Supervisor)	(Member)	(Member)

Ph.D. Thesis
was presented before the 7-Member Examination Committee
and approved on 13/07/2026

7-Member Examination Committee

Mihalis Psarakis
Professor
University of Piraeus
School of Information and Communication Technologies
Department of Informatics

Antonios Paschalis
Professor
National and Kapodistrian University of Athens
School of Science
Department of Informatics and Telecommunications

Cristiana Bolchini
Professor
Politecnico di Milano
School of Industrial and Information Engineering
Department of Electronics, Information and Bioengineering

Panayiotis Kotzanikolaou
Professor
University of Piraeus
School of Information and Communication Technologies
Department of Informatics

Aggelos Pikrakis
Associate Professor
University of Piraeus
School of Information and Communication Technologies
Department of Informatics

Apostolos Meliones
Professor
University of Piraeus
School of Information and Communication Technologies
Department of Digital Systems

Nektarios Kranitis
Professor
National and Kapodistrian University of Athens
School of Science
Department of Aerospace Science and Technology

ABSTRACT

The recent advances of the semiconductor industry have enabled the integration of complex components and system architectures into a single silicon die. Nowadays, state-of-the-art programmable circuit devices, such as field programmable gate arrays (FPGAs), include not only programmable logic fabric but also hardwired components, such as hard-core general-purpose processors, dedicated processing blocks, interfaces to various peripherals, on-chip bus structures, and analog blocks. These new heterogeneous devices are commonly referred to as all programmable system-on-chip (APSoC) devices.

One of the major concerns about radiation effects on APSoCs is that radiation-induced errors may have different probabilities and criticalities in their heterogeneous hardware parts at both device and design levels. For this reason, this work performs a deep investigation of the radiation effects on APSoCs and the correlation between hardware and software resource sensitivity in the overall system performance. Several static and dynamic experiments were conducted on various hardware components of an APSoC.

This thesis presents a comprehensive study on the single event effect (SEE) susceptibility of all major components in a commercial off-the-shelf (COTS) AMD Zynq-7000 APSoC. The work encompasses detailed radiation experiments, in-depth analysis of observed SEE phenomena, the SEE upset rate estimation for three different orbits, and the development of an efficient mitigation strategy to meet the high-reliability demands of space environments.

Initial efforts targeted the characterization of the programmable logic (PL) of the Zynq-7000 under ultra-high-energy and very-high-energy heavy ions at CERN Super Proton Synchrotron North Area (SPS-NA) in Geneva, Switzerland and GSI Helmholtz Centre for Heavy Ion Research in Darmstadt, Germany respectively, and high-energy protons at Paul Scherrer Institute (PSI) – Proton Irradiation Facility (PIF) in Villigen, Switzerland. Memory elements within both the design and configuration layers were evaluated, with calculated cross sections for each memory type. Novel or under-investigated SEE behaviors, such as single-event resets and functional interrupts, were revealed, which can evade traditional redundancy-based protection methods like triple modular redundancy (TMR). Multi-bit upset (MBU) and multi-cell upset (MCU) patterns were characterized across memory types, highlighting limitations in built-in error correction and informing the design of more robust scrubbing and interleaving schemes.

The study was extended to the processing system (PS) of the Zynq-7000 using proton at PSI-PFI and heavy-ion irradiation at GSI. SEE sensitivity of SRAM-based arrays (e.g., OCM, L1/L2 caches) was assessed under static and dynamic workloads, and application-level effects were quantified by executing processor benchmarks under varied cache configurations. These findings enabled accurate modeling of real-world error impact at the system level.

A key outcome is the introduction of a robust configuration memory scrubbing technique. The proposed mixed 2-D coding scheme integrates internal error correction code (ECC) with an interleaved parity code to detect and correct both single bit upsets (SBUs) and multiple bit upsets (MBUs) with minimal latency and overhead. Two implementations, an external microcontroller-based scrubber and an on-chip hardware scrubber, were validated under irradiation and compared favorably to state-of-the-art methods.

Finally, upset rate predictions were performed for typical space orbits using a dedicated software tool for space environment and radiation effects on electronic devices such as OMERE, demonstrating the applicability of the Zynq-7000 APSoC in radiation-prone missions. This work lays the groundwork for future APSoC reliability strategies and advances the state of knowledge in SEE effects and mitigation across heterogeneous SoC architectures.

Keywords: Programmable System-on-Chip (SoC), Field Programmable Gate Array (FPGA); Space Applications; Single Event Effect (SEE); Radiation Effects; Reliability; Memory Scrubbing; Heavy-Ion Irradiation; Proton Irradiation; Fault Tolerance; Fault Mitigation; Space Electronics.

ΠΕΡΙΛΗΨΗ

Οι πρόσφατες εξελίξεις στη βιομηχανία των ημιαγωγών επιτρέπουν πλέον την ενσωμάτωση πολύπλοκων υποσυστημάτων και ολόκληρων αρχιτεκτονικών σε ένα μόνο ολοκληρωμένο κύκλωμα. Σήμερα, οι σύγχρονες προγραμματιζόμενες διατάξεις, όπως τα FPGAs, δεν περιορίζονται απλώς σε προγραμματιζόμενη λογική, αλλά ενσωματώνουν πολλαπλές ηλεκτρονικές διατάξεις, όπως επεξεργαστές γενικού σκοπού, εξειδικευμένες μονάδες επεξεργασίας, εσωτερικούς διαύλους επικοινωνίας, διεπαφές με περιφερειακά κυκλώματα, καθώς και αναλογικά υποκυκλώματα. Οι νέες αυτές ετερογενείς συσκευές αναφέρονται πλέον ως πλήρως προγραμματιζόμενα συστήματα σε ολοκληρωμένο κύκλωμα (all programmable system-on-chip – APSoC).

Ένα από τα κύρια ζητήματα που τίθενται για τα APSoCs είναι η συμπεριφορά τους απέναντι στην ακτινοβολία. Τα σφάλματα που προκαλούνται από την ακτινοβολία μπορεί να εμφανίζονται με διαφορετική πιθανότητα στα διάφορα μέρη του ολοκληρωμένου κυκλώματος ή μπορεί να έχουν διαφορετική κρισιμότητα ανάλογα με το υποσύστημα που επηρεάζουν. Για τον λόγο αυτό, η παρούσα εργασία μελετά σε βάθος τις επιπτώσεις της ακτινοβολίας στα APSoCs και τη συσχέτιση μεταξύ της ευαισθησίας υλικού και λογισμικού με τη συνολική αξιοπιστία του συστήματος. Η ανάλυση βασίστηκε σε εκτενή πειραματική αξιολόγηση στα διαφορετικά μέρη ενός APSoC.

Η διατριβή εστιάζει στη μελέτη της ευπάθειας όλων των βασικών υποσυστημάτων μίας εμπορικά διαθέσιμης (commercial off-the-shelf – COTS) AMD Zynq-7000 APSoC συσκευής σε επιδράσεις μεμονωμένων σωματιδίων (single event effects – SEE). Η εργασία περιλαμβάνει λεπτομερή πειράματα ακτινοβολίας (radiation experiments), εις βάθος ανάλυση των παρατηρούμενων φαινομένων SEE, την ακριβής πρόβλεψη του ρυθμού λαθών (SEE upset rate) για τρεις διαφορετικές διαστημικές τροχιές, καθώς και την ανάπτυξη μιας αποτελεσματικής μεθοδολογίας μετριασμού σφαλμάτων (mitigation strategy) σχεδιασμένης να καλύψει τις υψηλές απαιτήσεις αξιοπιστίας των διαστημικών εφαρμογών.

Το πρώτο μέρος της έρευνας επικεντρώθηκε στην προγραμματιζόμενη λογική (programmable logic - PL) του Zynq-7000. Πραγματοποιήθηκαν τρία πειράματα ακτινοβολίας σε διεθνή ερευνητικά κέντρα, το πρώτο πείραμα με χρήση βαρέων ιόντων εξαιρετικά υψηλής ενέργειας στο CERN (Conseil Européen pour la Recherche Nucléaire) στην Ελβετία, το δεύτερο με χρήση βαρέων ιόντων πολύ υψηλής ενέργειας στο GSI (Helmholtz Centre for Heavy Ion Research) στη Γερμανία και το τρίτο πείραμα με χρήση προτωνίων υψηλής ενέργειας στο Paul Scherrer Institute (PSI) – Proton Irradiation Facility (PIF) στην Ελβετία. Εξετάστηκαν διαφορετικά είδη μνήμης, τόσο σε επίπεδο σχεδίασης, όπως οι καταχωρητές (flip-flops – FFs), οι καταχωρητές ολίσθησης (shift registers look-up tables – SRLs) και τα μπλοκ μνήμης (block RAM – BRAM) όσο και σε επίπεδο διαμόρφωσης όπως η μνήμη διαμόρφωσης (configuration memory – CRAM), δηλαδή η μνήμη μέσα στην οποία αποθηκεύεται η πληροφορία που καθορίζει

πώς θα συνδεθούν τα ανωτέρω λογικά στοιχεία ώστε να υλοποιούν το επιθυμητό ψηφιακό κύκλωμα. Υπολογίστηκε η πιθανότητα εμφάνισης διαταραχών που προκαλούνται από ένα συμβάν (SEU cross sections) για κάθε τύπο μνήμης, ενώ εντοπίστηκαν νέα φαινόμενα SEE που δεν είχαν μελετηθεί επαρκώς στη βιβλιογραφία, όπως σφάλματα επαναφοράς (single-event resets) ή διακοπές λειτουργίας (functional interrupts), μερικά εκ των οποίων δεν μπορούν να διορθωθούν από παραδοσιακές τεχνικές ανοχής σε σφάλματα, όπως η τριπλή πλεοναστική πλειοψηφία (triple modular redundancy – TMR). Η σε βάθος ανάλυση των πολλαπλών διαταραχών που προκαλούνται σε όλους τους τύπους μνήμης (multi-bit upsets – MBUs και multi-cell upsets – MCUs) αναδεικνύει τους περιορισμούς της ενσωματωμένης διόρθωσης σφαλμάτων στο Zynq-7000 και καθοδηγεί τον σχεδιασμό πιο αξιόπιστων σχημάτων διόρθωσης σφαλμάτων.

Η μελέτη επεκτάθηκε στο σύστημα επεξεργασίας (processing system – PS) του Zynq-7000 μέσω δοκιμών ακτινοβολίας με τη χρήση πρωτονίων στο PSI-PFI και βαρέων ιόντων στο GSI. Η ευαισθησία σε SEE των SRAM μνημών, όπως είναι η on-chip memory (OCM) και οι κρυφές μνήμες (L1/L2 caches), αξιολογήθηκε υπό στατικά (η μνήμη είναι στατική κατά τη διάρκεια της ακτινοβολίας) και δυναμικά φορτία (πρόσβαση στη μνήμη κατά τη διάρκεια της ακτινοβολίας), και οι επιπτώσεις σε επίπεδο εφαρμογής ποσοτικοποιήθηκαν μέσω εκτέλεσης δοκιμαστικών προγραμμάτων για διαφορετικές παραμετροποιήσεις των κρυφών μνημών. Τα ευρήματα αυτά επέτρεψαν την ακριβή μοντελοποίηση της πραγματικής επίδρασης των σφαλμάτων σε επίπεδο συστήματος.

Η πρόταση μιας νέας προσέγγισης για την αποσφαλμάτωση της μνήμης διαμόρφωσης (memory scrubbing) αποτελεί βασικό μέρος της διατριβής. Η προτεινόμενη μικτή δισδιάστατη τεχνική κωδικοποίησης (mixed 2-D coding scheme) συνδυάζει τον εσωτερικό κώδικα διόρθωσης σφαλμάτων (ECC) ενός APSoC με έναν διεμπλεκόμενο κώδικα ισοτιμίας (interleaved parity code), ώστε να ανιχνεύει και να διορθώνει τόσο μεμονωμένες ανατροπές bit (single bit upsets – SBUs) όσο και πολλαπλές ανατροπές bit (multiple bit upsets – MBUs), με μικρή χρονική καθυστέρηση και περιορισμένο υπολογιστική επιβάρυνση. Η μέθοδος υλοποιήθηκε με δύο τρόπους, είτε με τη χρήση ενός εξωτερικού μικροελεγκτή (external microcontroller-based scrubber) είτε ως ενσωματωμένο κύκλωμα στην συσκευή υπό δοκιμή (on-chip hardware scrubber), συγκρίθηκε με τις πλέον προηγμένες μεθόδους και δοκιμάστηκε σε συνθήκες ακτινοβολίας.

Τέλος, παρουσιάστηκε εκτίμηση του ρυθμού σφαλμάτων για τυπικές διαστημικές τροχιές χρησιμοποιώντας εξειδικευμένο λογισμικό που μοντελοποιεί το διαστημικό περιβάλλον και τις επιδράσεις του στα ηλεκτρονικά κυκλώματα, όπως είναι το OMERE, αποδεικνύοντας την καταλληλότητα του Zynq-7000 APSoC για αποστολές σε περιβάλλοντα με έντονη ακτινοβολία. Η εργασία αυτή συμβάλλει στη θεμελίωση νέων στρατηγικών για την αξιοπιστία των APSoCs και εμπλουτίζει την επιστημονική γνώση

σχετικά με τα SEE φαινόμενα και τις μεθόδους μετριασμού τους σε ετερογενείς αρχιτεκτονικές SoC.

Λέξεις-κλειδιά: Programmable System-on-Chip (SoC), Field Programmable Gate Array (FPGA), Διαστημικές Εφαρμογές, Single Event Effect (SEE), Επιδράσεις Ακτινοβολίας, Αξιοπιστία, Memory Scrubbing, Βαρέα Ιόντα, Πρωτόνια, Ανοχή σε Σφάλματα, Μετριασμός Σφαλμάτων, Διαστημικά Ηλεκτρονικά.

ACKNOWLEDGEMENTS

With the completion of this dissertation, I have reached the end of a long and demanding journey, supported and assisted by many people over the past years.

I would like to express my sincere and profound gratitude to my supervisor, Professor Mihalis Psarakis, for his unwavering support, academic guidance, and pivotal contribution throughout my research journey. His commitment to scientific quality, constant availability for discussion and feedback, and the trust he placed in me have been the solid foundation upon which this dissertation was built. He encouraged me and believed in me even during the most demanding phases of the research. More than that, he has been a true collaborator; someone who inspired me, strengthened me in moments of challenge, and stood by me every step of the way with integrity, consistency, kindness, and honesty. I am deeply grateful for our meaningful, creative, and truly human collaboration and for the mutual trust that developed between us.

A significant figure in both my professional and research journey over the years has been Dr. Antonis Tavoularis. He was the one who gave me my very first opportunity to work. His trust in my potential and his support during my early steps laid the foundation for my subsequent development and progress. He believed in me and was the one who encouraged me to embark on this PhD. Throughout the course of my research, especially during the challenging early stages, his contribution was invaluable. Our fruitful scientific discussions, his insightfulness, openness to the exchange of ideas, and the directness of our communication contributed not only to the advancement of this dissertation but also to my personal and scientific growth.

I owe special thanks to my colleagues at the Embedded Systems Laboratory (ESLab) of the University of Piraeus, Mr. Aitzan Sari, Dr. Dimitrios Agiakatsikas, Mr. Georgios Antonopoulos and Ms. Ioanna Souvatzoglou, for their generosity in sharing resources, knowledge, and time, as well as for their invaluable contribution to data collection and analysis, and their practical assistance at various stages of the research process.

I would also like to thank my parents, Dionissis and Paraskevi, for the support they provided during my early years. I feel the need to express heartfelt gratitude to my father, who, although he passed away before I began this journey, showed me the way and gave me the foundation to follow it in my own time and in my own way.

Finally, I would like to extend my warmest thanks to my family for their patience and understanding throughout this demanding path.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Space and COTS Programmable Devices

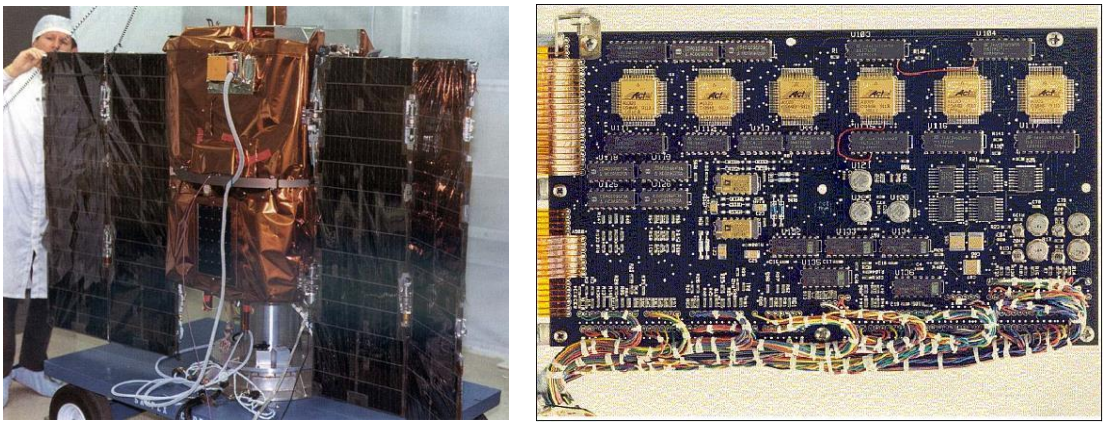
Since ancient times, humanity's interest in outer space has been lively. This innate desire to explore the unknown and explain the (probably) known has led human to launch missions, crewed and uncrewed, into space for more than 65 years, and, among others, to use space science and technology in various areas, e.g., aviation, maritime and land transportation, human health, disaster management, environmental monitoring and natural resources management. But the cost and complexity of space technology is such that hundreds of millions of euros can be spent, mainly due to the need to use expensive space-grade components, capable of operating for many years in extremely harsh conditions. For example, a conventional satellite, as a reference, can cost between 150 and 300 million euros [37] and can have a lifespan of several years to several decades in low orbits, i.e., at an altitude of a few hundred kilometers from the ground of the Earth, where atmospheric drag may produce unacceptable decay rates, or more than 100 years at higher altitudes, such as in geostationary orbit (GEO) positions [59].

However, in recent years, the space has become more accessible with the advent of what's termed "NewSpace" and the launch of commercial spaceflights, which has pushed the science community to focus on reducing costs, using commercial off-the-shelf (COTS) products and ensuring that low costs will pay off. Typically, NewSpace satellites, operating in low earth orbit (LEO) up to 2,000 km above the Earth's surface, can be built for less than one million euro each [37] and designed to last up to 25 years [101]. Their cost is a small fraction (less than 1%) of the price of a traditional spacecraft, while their lifetime is considerable compared to the lifetime of the latter.

Programmable circuit devices, such as field programmable gate arrays (FPGAs), have made a significant contribution to advancing the state-of-the-art for the emerging space industry. FPGAs featuring high flexibility, combined with high performance and complexity are increasingly being used for space applications far beyond one decade. Reprogrammability in flight becomes a strict requirement for a satellite given that the more technology grows, the longer the satellite lives and, in parallel, the space standards change over its lifetime. On the other hand, reprogrammability can lead to multitasking

satellites capable of switching, for example, from earth observation to searching for earthlike planets.

The Solar Anomalous and Magnetospheric Particle Explorer (SAMPEX) [28], also known as Explorer 68, presented in Figure 1.1(a), was the first spacecraft globally to utilize the first FPGA. It was a National Aeronautics and Space Administration (NASA) solar and magnetospheric observatory, the first of a series of spacecraft under the Small Explorer (SMEX) program for low-cost spacecraft, successfully launched into LEO on 3rd of July, 1992, from Vandenberg Air Force Base. The SAMPEX data processing unit (DPU), which provided overall control of the scientific payload while acquiring, combining, and compressing science data to produce the science telemetry for the mission, made extensive use of Actel FPGAs to implement the hardware system, as shown in Figure 1.1 (b).



(a) (b)

Figure 1.1: NASA's SAMPEX spacecraft - First FPGA in space [110], [122]

Since then, mainly space-grade FPGA devices have been used in many space missions, commonly in data processing. Space-grade products are built, tested, characterized, and specified to have a known performance in a space radiation environment. The dominant space-grade categories launched consist of the antifuse-, flash-, and static random-access memory (SRAM)- based FPGAs. Antifuse-based devices are one-time programmable (OTP) while the other two are reprogrammable. In addition, antifuse- and flash-based are non-volatile, i.e., their configuration data remains when the system is powered down. As a result, these devices do not require external memory for configuration storage, offering advantages in terms of system simplicity and startup latency [102].

The internal construction of antifuse structure is basically impossible to dissolve without the damage of the device, making the technology highly reliable. For this reason, they are perfect for a lot of critical space applications like command and data handling, altitude

and orbit control, and spacecraft environmental and power control. Following antifuse technology, flash-based FPGAs combine the advantages of the antifuse technology with the ability to be reprogrammed in-orbit.

However, SRAM-based FPGAs have taken advantage of the continuous development of complementary metal oxide semiconductor (CMOS) technology, its high performance and increased logic density. Along with the fact that they can be reprogrammed during system development and also in-orbit, SRAM-based has been becoming the state-of-the-art space technology used in multiple space programs. For example, NASA's Mars 2020, launched on 30 July 2020, on which the AMD Xilinx Virtex-5QV [14], [66], integrated in the Mars rover "Perseverance" (Figure 1.2), is first responsible for rover entry, descent and landing on Mars and after landing, it accelerates certain tasks, including image detection, matching, and rectification and also analyzes and filters out useless data that it has captured before sending it to Earth [84]. Other units on Perseverance such as UHF transceivers, radar and X-ray (identifying chemicals) are controlled with AMD Xilinx Virtex-4QV [13] and AMD Xilinx QPro Virtex-II [11] FPGAs.

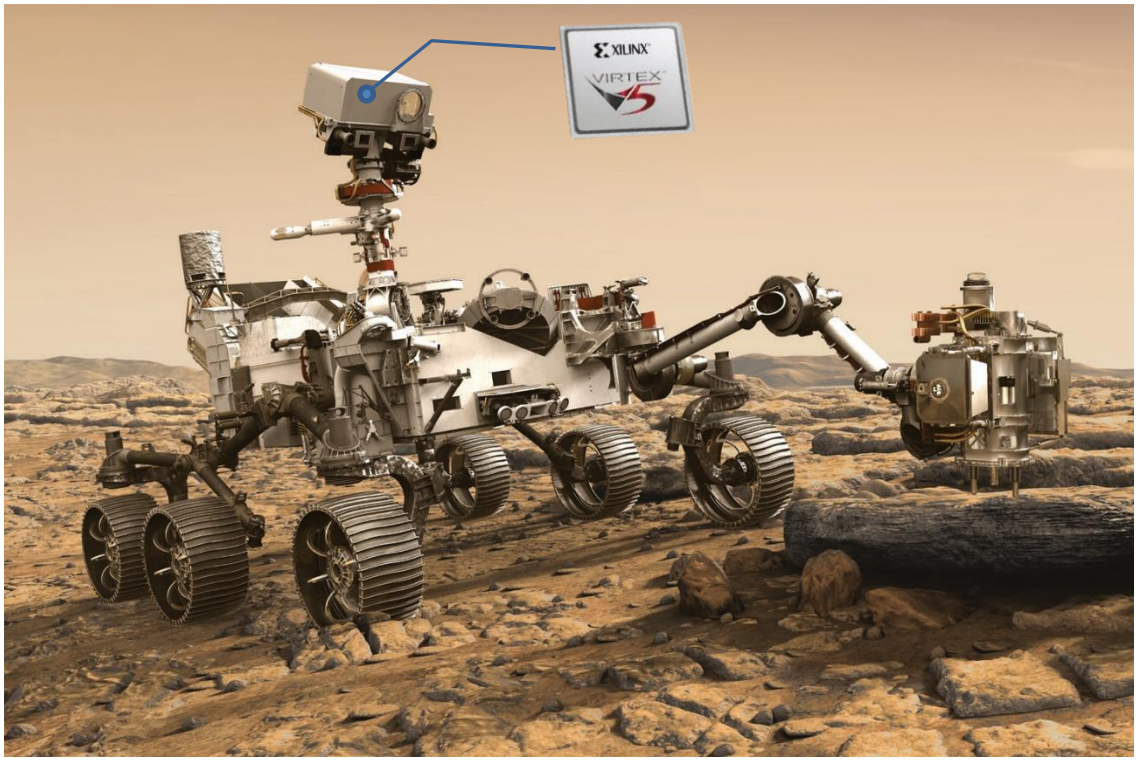


Figure 1.2: The Mars rover Perseverance [109]

In space mission set-ups, in addition to FPGAs, performing data processing, microprocessors are integrated operating as the "brain" and implementing the command and control applications through specific interfaces to various peripherals. However, the

rapid growth of silicon technology in recent years promoted the integration of single or multicore processors with many of the elements of a traditional computer system, such as on-chip peripherals, memory controllers and high speed input/output (IO). These devices, commonly called system on a chip devices (SoCs), achieve performance and miniaturization goals [53]. State-of-the-art computing systems rely on the combination of the advantages of both FPGAs and SoCs, i.e., the higher integration, low power, real-time operations, high bandwidth communication between the processor and FPGA, into a single device which forms an even more powerful embedded computing heterogeneous platform, referred to as all programmable SoC (APSoC) device. In general, APSoCs consist of two main parts, the programmable logic (PL), which is basically an embedded FPGA, and the processing system (PS), which is based on a single or multicore processor. The heterogeneity of the APSoC architectures offers more than just an FPGA accelerator, including a powerful processor, which allows applications to be run from a convenient operating system (OS) environment.

The integration of SoC and FPGA fabric into an APSoC device magnifies benefits such as cost, performance and miniaturization at system level, especially for the development of on-board computers (OBC) where the SoC flexibility is highly desirable [94]. OBC refers to as the brain of the satellite system, i.e., the unit which controls all the subsystems and monitors the health of each subsystem implementing functions such as attitude and orbit control, telecommands execution or dispatching, housekeeping telemetry gathering and formatting, on board time synchronization and distribution, failure detection, isolation and recovery, etc. The programmable part of APSoC, based in FPGA, implements high-speed arithmetic and data processing of the OBC, while the processing part, based in SoC, supports software routines and operating system.

Given that the spacecraft development landscape faces a substantial shift from traditional large satellites to SmallSats and CubeSats, the space design engineers recommend using APSoC devices to satisfy the performance requirements and the resource constraints imposed by the limited SWaP (size, weight, and power) budget of the OBC of small satellites [55]. The size and cost of spacecraft vary depending on the application; some you can hold in your hand while others are as big as a rail wagon. SmallSats focus on spacecraft with a mass less than 200 kilograms and smaller than the size of a large kitchen fridge. CubeSats are a subclass of SmallSats. They shall measure 10x10x10 cm^3 and shall weight no more than 1.33 Kg. Typical power consumption is on the order of a few Watts, and available data rates do not exceed 1 Mbps [127]. CubeSats now provide a cost effective platform, especially using COTS SRAM-based FPGAs and APSoC devices, for science investigations, new technology demonstrations and advanced mission concepts using constellations, swarms disaggregated systems.

Despite miniaturized APSoC technology is very convenient for space, where every gram launched carries huge costs, it is currently used primarily by consumer terrestrial applications, where a single powerful device with very low power consumption is the best solution for the network and telecommunication systems. COTS APSoCs have

developed very advanced computation capabilities in consumer electronics that is continuously demanding more powerful devices and applications. Some recent examples of commercially available APSoCs are AMD Xilinx Zynq-7000 [15], Zynq UltraScale+ [18] and Versal [19], Microchip (Microsemi) PolarFire [103] and SmartFusion2 [106], and Intel (Altera) Agilex [69], Stratix10 [74] and Cyclone V [71] series. Current APSoC devices available in the market typically include at least one processor and an FPGA fabric. Since Advanced RISC Machines (ARM) cores are the standard processors that all APSoCs use, APSoCs are differentiated by the FPGA fabric they use. This fabric embeds routing resources, programmable logic, e.g., flip-flops (FF) and lookup tables (LUTs), digital signal processing (DSP) units and random-access memory (RAM) blocks together with the memory cells to store their configuration.

However, high susceptibility in transient effects, such as the single event effects (SEEs) caused by the space ionizing radiation, is a blocking factor to employ COTS SRAM-based FPGAs and APSoCs in dependable embedded systems [118]. An SEE affecting the sequential logic of a processor system executing an application can result in wrong outcomes or even in the crash of the system with a dramatic effect on the application. In the case of FPGAs, SEEs can lead to functional change in a logic module or misconnected or misrouted signals, resulting eventually in system failure. SEEs occur when charged particles collide with the silicon of an electronic component transferring enough energy in order to provoke a fault in the electronic device [88]. Energetic charged particles, including electrons, protons and heavy ions, are prevalent in the Earth's magnetosphere, interplanetary space and the magnetospheres of other planets. When a single ionizing particle interacts with an electronic device, it can deposit a localized charge that induces a wide range of SEEs which can have a destructive or transient effect. The nature and severity of these effects depend on both the energy transferred and the specific location of the interaction within the device. Some of them, such as the temporary voltage spike at a node in an integrated circuit known as single-event transient (SET), the change of state of a storage element called single-event upset (SEU) and the soft error that causes an electronic component to reset or lock-up known as single-event functional interrupt (SEFI), are temporary and can be recovered. Others, such as single-event latch-up (SEL), i.e., a type of short circuit which can occur in an integrated circuit, can lead to permanent damage [47]. Thus, SEE ground testing for candidate spacecraft electronics has become a key in many spaceflight programs. To SEE characterize a COTS SRAM-based device for a space mission, it is essential to perform test campaigns irradiating the device with both heavy ions and protons.

As a charged particle traverses a material, it undergoes energy loss due to its interactions with the material. The amount of energy lost by the particle per unit length of track is called linear energy transfer (LET) which depends on the nature of the radiation as well as on the material traversed. By counting the number of SEEs and knowing how many particles passed through an electronic part, we can calculate the probability of a particular particle causing a SEE. This resultant number, which is the number of upsets divided by

the number of particles per cm^2 causing the upsets, is called the cross section of the part. A device's susceptibility to SEEs is commonly characterized by its cross section as a function of the LET. The cross section quantitatively reflects the likelihood that an incident energetic particle induces a single event. But not all the single events will cause an error in a FPGA design or an application running on a processor. It is important to account for the specific design or application to be used because the sensitivity is very design- or application- dependent and can even vary amongst different implementations. The vulnerability factor of a device that expresses this, is the error rate (i.e., the number of errors per device per day). The error rate can be estimated from LET, cross section and parameters of the operational orbit.

Error rate evaluation is typically conducted using methodologies based on radiation ground testing, commonly referred to as accelerated testing. These experiments are carried out in specialized facilities, such as cyclotrons or linear accelerators, where the target circuits are exposed to controlled particle beams while executing representative workloads that mimic their intended operational behavior in the final application environment. The outcomes of these radiation tests are then analyzed to estimate the likelihood of device degradation, damage, and functional error rates. However, because radiation testing must be repeated whenever modifications are made to the application, it can result in substantial development costs. As an alternative, off-beam fault injection campaigns offer a cost-effective means of predicting error rates without the need for physical irradiation. This approach provides a practical solution for assessing reliability, particularly in scenarios where frequent design iterations or budget constraints make repeated radiation testing impractical [146].

The primary approaches to mitigate or eliminate the radiation effects of microelectronics operating in radiation environments are radiation-hardened, often referred to as rad-hard, and radiation-tolerant, referred to as rad-tolerant. Rad-hard devices are electronics specifically designed and produced to be less susceptible to damage from exposure to radiation and extreme temperatures. While they perform similar functions to their commercial counterparts, their design, materials, and manufacturing processes are optimized to withstand various forms of radiation encountered in space, high-altitude flight, and research facilities. This often includes the use of insulating substrates rather than conventional semiconductor wafers, significantly increasing their radiation endurance. On the other hand, rad-tolerant devices offer a compromise between commercial-grade and fully hardened components. Although not capable of enduring the same radiation levels or prolonged exposure as rad-hard systems, they provide sufficient robustness for applications involving moderate radiation levels or shorter mission durations. As such, rad-tolerant components are frequently employed in cost-sensitive or less demanding environments where some level of radiation resistance is necessary, but full hardening is not justified.

Although both space-grade and COTS devices can be used in space, space-grade devices are often necessary, depending on the mission's orbit or location, planned lifetime, and

requirements for reliability and accessibility. However, rad-hard and rad-tolerant devices often comes with associated costs [57], including slower operating frequencies, decreased numbers of processor cores or computational units, increased power dissipation, and less efficient than the COTS devices [97]. Traditionally, space-grade processors have come in the form of single-core CPUs. However, in recent years, development has occurred on space-grade processors with more advanced architectures such as multicore CPUs and FPGAs. Most existing space-grade processors, such as BAE Systems RAD750 [27], which is single-core CPU, and Gaisler GR716 [83] and Gaisler GR740 [54] which are multicore CPUs provide performance levels up to 2.0 Dhrystone MIPS/MHz per core while AMD Xilinx Zynq-7000 single-core version reaches performance 2.5 Dhrystone MIPS/MHz and AMD Xilinx Versal (dual-core) achieves more than 15.5K Dhrystone MIPS/MHz.

However, some promising attempts are currently being made to develop potent radiation-hardened SoCs for space. BRAVE NG-Ultra [107], which is developed by NanoXplore and will go through a qualification process for flight applications, is the world's first radiation hardened by design (RHBD) SoC integrating a quad-core ARM running at 600MHz and FPGA fabric. In addition, NASA's high performance spaceflight computing (HPSC) project [108] is focused on advancing onboard computing capabilities by developing next-generation flight computing technology that offers a minimum of 100-fold improvement in computational performance over existing spaceflight processors. The initiative aims to produce newly architected multicore processor chips, each integrating multiple processing cores, accompanied by the necessary operating software to efficiently manage and utilize the enhanced hardware resources.

An alternative solution is to study efficient error mitigation techniques to enable the deployment of COTS devices in radiation environments and meet the high-reliability requirements of space applications. Several fault tolerance methods have been applied to COTS devices, such as 1) the redundancy techniques for error detection and masking, where the resources are replicated in order to process the same task in parallel and a voting circuitry is in charge of error detection and correction, 2) the memory scrubbing to correct memory upsets and avoid fault accumulation by reading the current state of the memory and writing back into the memory correct value usually stored in an external rad-hard memory, 3) the error correction code (ECC) algorithms to support the ability for detecting and correcting errors within memories by adding additional check data to the original data [47], [88], [156].

Mitigation techniques have been used in many space missions. Venus Express was European Space Agency's (ESA) first spacecraft to voyage to our nearest planet, to study its complex dynamics and chemistry, the interactions between the atmosphere and the surface, and especially the atmosphere and clouds in unprecedented detail and accuracy. It uses an AMD Xilinx Virtex SRAM-based FPGA in its micro Venus Express monitoring camera (VMC) to implement a microprocessor using Gaisler LEON2 core as

well as peripheral logic and interfaces to various sensors and communication units. Partial hardware redundancy, reconfigurable subunits, majority voting, and graceful degradation of performance are implemented to get required reliability [50]. For a timely protection of all FPGA areas and to eliminate SEU effects before they are accumulating, additional supervisor logic is implemented. Another radiation hardened and TMR by design, one-time programmable FPGA is used as supervisor circuit to achieve radiation tolerance. VMC was successfully verified in space during Venus orbit insertion and routine science operations, which had been started mid. of May 2006 and completed in 2014.

The NASA's SpaceCube v2.0 high performance data processing system for space applications also uses in-flight reconfigurable SRAM-based FPGA which has been tested on a Hubble Space Telescope servicing mission of the Space Shuttle Atlantis [113], [114]. It supports various mitigation techniques for protecting against radiation effects that can be tailored to a specific project's requirements. The FPGA configuration can be scrubbed by either an external radiation hardened FPGA or by using a self-scrubber internal to the FPGA. The internal memory attached to the SRAM-based FPGA contains internal error detection and correction (EDAC) and scrubbing capability.

Data acquisition, synchronization and processing of the space payload systems are currently implemented by either a commercial flash-based FPGA (e.g., Microchip ProASIC3 [104]) or a space-grade SRAM-based FPGA (e.g., AMD Xilinx Virtex5QV [14]), while an antifuse-based FPGA (e.g., Microchip RTAX [105]) or a rad-hard processor (e.g., BAE Systems RAD750 [27]) is used for spacecraft guidance, navigation and control (GNC), data communication with the spacecraft computer, data collection and configuration of the SRAM-based FPGA. This architecture has been encountered in various instruments, such as the NASA's Mars Curiosity rover and the SpaceCube, and the ESA's VMC, as shown above. Thus, an APSoC that includes these two parts, processor and programmable logic, into a single chip is perfectly suited for space instrument payload systems.

AMD Xilinx is one of the vendors with the most advanced APSoC technology roadmap [79]. The AMD Xilinx SoC portfolio is comprised of Zynq-7000, Zynq UltraScale+ and Versal series devices integrating the software programmability of a processor with the hardware programmability of an FPGA (Table 1.1). In addition, the space programs using AMD Xilinx SoCs have increased in recent years. Such cases are the Space Test Program-Houston-ISS-5 SpaceCube experiment [154] and the Compact Radiation bElt Explorer (CeREs) CubeSat mission [86] which use the space-flight-proven CubeSat Space Processor (CSP) based on a Zynq-7000 and the LinkStar-X, an integrated Zynq UltraScale+ based flight computer [124]. In 2023, LEONIDAS and ASTRID are the first missions to fly a Versal in space that serves as pathfinder for the ISS payload capability, providing easy and quick access to space for low-rigor research experiments [31].

	Zynq-7000	Zynq UltraScale+	Versal
Application processing unit	Single or Dual ARM Cortex-A9	Dual or Quad ARM Cortex-A53	Dual ARM Cortex-A72
Real-Time processing unit	-	Dual ARM Cortex-R5F	Dual ARM Cortex-R5F
Programmable logic process technology	28nm 7-Series	16nm FinFET+	7nm FinFET
Proton SEL immunity	No SEL	SEL at 64 MeV	-
Heavy Ion SEL immunity	LET < 16 MeV·cm^2/mg	LET < 3.3 to 5.7 MeV·cm^2/mg	No SEL at 69 MeV·cm2/mg

Table 1.1: AMD Xilinx SoC platforms

The main inhibitory factor for a device to be considered acceptable for space missions is its susceptibility to proton SEL. Several experiments performed on Zynq-7000 have shown that no SEL due to proton irradiation was observed [64]. Other works have presented that the lowest LET threshold that any latch-up happened was 16 MeV·cm^2/mg under heavy ion irradiation [22], [133]. Note that only about 10 particles pass through a 1 cm^2 device in a year that have LET of 16 MeV·cm^2/mg or greater [111] and thus, the events will not occur more than about once a month in the LEO environment. Therefore, Zynq-7000 can be used on space stations [e.g., the international space station (ISS) and China's tiangong space station (TSS)] or any other LEO short duration mission for a significant amount of time without reset or reconfiguration and without resulting in destruction of the device. No reconfiguration defines the availability of the device and the fact that the effect is not destructive defines its reliability. Various radiation tests have also been carried out for the Zynq UltraScale+ in recent years. No proton SEL events were observed in [36] and [65], however other works did report them for proton test at 64 MeV [87]. In addition, the SEL threshold was found between LET of 3.3 and 5.7 MeV·cm^2/mg under heavy ion irradiation [58], which means that 100 - 300 particles pass through a 1 cm^2 device in a year, i.e., about up to one event per day will happen in low orbits. In [92], a high-current anomaly was observed on a voltage supply rail across all LETs in heavy ions. This anomaly resulted in a destructive event when the event was allowed to remain in the device and when no current limit was set on the voltage rail. Finally, the Versal series were commercially available in the second half of 2019. In 2021, the first SEL test was performed on this technology [41] and Versal was found to be immune to latch-up up to an effective LET in the active area of 69 MeV·cm2/mg.

From the above it follows that the Zynq-7000 is more suitable for the space domain than the Zynq UltraScale+. Thus, in this thesis, the Zynq-7000 APsoC was selected for study, as Versal is a recent architecture that was not available in the early years of the thesis.

AMD Xilinx Zynq-7000 APSoC is a COTS device which includes ARM Cortex-A9 (single- or dual-core) high-performance processor (PS) integrated with 28-nm AMD Xilinx 7-Series (Artix-7 or Kintex-7) based FPGA architecture (PL). The chip includes abundant on-chip AXI interfaces with low power rails to communicate the PS with the PL, which results in substantially less power consumption, considerably higher bandwidth and lower latency. This means that the processor and programmable logic can each be used for what they do best, without the overhead of interfacing between two physically separate devices. The PS has a Level 1 cache for local storage of frequently required data (L1-D) and instructions (L1-I) for fast access times and optimal processor performance, as well as a larger Level 2 (L2) cache for instructions and data and a further on-chip integrated memory (OCM). It also features a variety of external peripherals and high-speed communications interfaces. The PL, the second principal part of the Zynq architecture, is predominantly composed of general purpose FPGA logic fabric, consisting of configurable logic blocks (CLBs), DSPs and block random-access memories (BRAMs), and there are also input/output blocks (IOBs) for interfacing. CLBs are contains 8 LUTs, 16 FFs, and other logic, while LUTs can be configured as either distributed RAM (DRAM) or shift register look-up table (SRL).

An important issue regarding radiation effects on Zynq-7000 is the different vulnerability to radiation-induced errors in the PL and PS parts. Zynq-7000 is programmed by configuring a SRAM memory, called configuration RAM (CRAM), which stores the information about the user circuit of the PL. These configuration memory cells are very susceptible to SEE and reconfiguration is needed to correct them. When an application is running on the PS, SEEs may cause malfunction according to the application and the resources in use. Memories of the PS part such as caches and OCM and BRAMs of the PL part are also very sensitive to SEE, each with a distinct sensitivity, contributing differently to the overall system degradation.

Several radiation experimental results have been recently presented for the Zynq-7000. Some approaches investigate the radiation-induced soft errors in various memory structures located either in the FPGA fabric, such as the CRAM and the BRAM, or the SoC area of the Zynq-7000 device, such as the OCM and the L1 and L2 caches, under heavy ion and/or proton [22], [40], [96], [130], [135], [136], [137], [158], [159]. Other approaches focus on the impact of SEEs in the processing system and its subsystems [95], e.g., arithmetic logic unit (ALU), floating point unit (FPU), direct memory access (DMA), of the Zynq-7000 SoC. In [22], the SEE susceptibility of CRAM and BRAM was investigated using heavy ions with LETs from about 2 to 22 MeV·cm^2/mg and OCM and L1 cache using an 18 MeV proton beam. Similarly, in [135] the authors were tested the CRAM, BRAM and OCM for SEUs using heavy ions with effective LETs varying from about 2.6 to 17 MeV·cm^2/mg. In [136], the same authors extended the heavy ions tests for the CRAM and BRAM considering variations in the nominal supply voltage and temperature of the chip. They also performed experiments with the full range of proton energies, from 50 to 250 MeV, but they presented results only for the CRAM and the L2

cache. These researchers examined the impact of using different memory cache organizations and communication schemes on the Zynq-7000 in the system failure rate using a 7.3 MeV·cm^2/mg heavy ion beam [137]. In [40], proton irradiation experiments were performed to characterize the SEE sensitivity of both the memories (CRAM, D-cache) and the functional components (ALU, FPU, DMA) of the Zynq-7000 device but with a low-energy proton beam (3 to 10 MeV). In [158] and [159], the authors performed heavy-ion microbeam experiments with an CI beam of 13.59 MeV·cm^2/mg LET or an Ni beam of 25 MeV·cm^2/mg, respectively to calculate the SEE sensitivity distributions of various SoC components (OCM, D-cache, ALU, FPU and peripherals) and identify the SEE sensitive spots in different chip regions. In [96], the SEE sensitivity of the SIMD coprocessor of the ARM core (NEON) with low energy protons (15.4 MeV) was studied. Finally, in [130], the researchers presented a comprehensive analysis of the SEEs observed on the PS OCM using proton beam with several energies, from 16 to 200 MeV.

Component	Heavy ions	Protons
CRAM	2 – 22 MeV·cm^2/mg [22] 2.6 – 17 MeV·cm^2/mg [135]	3 – 10 MeV [40] 50 – 250 MeV [136]
BRAM	2 – 22 MeV·cm^2/mg [22] 2.6 – 17 MeV·cm^2/mg [135]	-
LUTs	-	-
FFs	-	-
OCM	2.6 – 17 MeV·cm^2/mg [135] 13 – 25 MeV·cm^2/mg [158], [159]	18 MeV [22] 16 – 200 MeV [130]
L1 cache	13 – 25 MeV·cm^2/mg [158], [159]	18 MeV [22] 3 – 10 MeV [40]
L2 cache	-	50 – 250 MeV [136]
CPU SDCs	-	-

Table 1.2: Existing Radiation Tests for Zynq-7000

The experimental data shall cover a range of heavy ion LETs and a range of proton energies to guarantee accurate prediction of the SEE rates. These ranges should span from the minimum LET/energy threshold required to trigger single events in the device under study up to the maximum particle energy predicted for the target radiation environment or up to the point that the SEE cross section of the device saturates. However, as presented in Table 1.2, all of the above work does not provide a complete picture of the susceptibility for all the components of Zynq-7000, i.e., CRAM, BRAM, FFs, LUTs, OCM, L1/L2 caches in order to provide the means for an accurate prediction of the system error rates in a space radiation environment. Additionally, no work presents

the impact of radiation-induced errors at the application-level by executing software benchmarks on the ARM processor and measuring silent data corruptions (SDC) (i.e., errors in system data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data) and system crashes. In addition, to guide the design of effective SEE mitigation approaches, such as ECCs able to correct multiple-cell upsets (MCUs) (i.e., the phenomenon of a single particle strike affecting multiple logic cells) in memories or multiple errors in state registers caused by SETs on global signals, a more thorough investigation of the radiation effects is required. In this thesis work, all the above was achieved by performing several experiments in various irradiation environments, using heavy ions with LETs from 2.29 to 12.45 MeV·cm^2/mg and protons with energy range 30 to 200 MeV, followed by an in-depth analysis of the SEE vulnerability of all memories and system reliability of Zynq-7000 family.

Based on this detailed study, the design and implementation of a new proposed efficient hardening technique, which was validated under irradiation and compared with the state-of-the-art, was also developed in this thesis work. Specifically, a mitigation approach, that improves the MCU correction capabilities of the on-chip ECC schemes in SRAM-based memories (e.g., FPGA configuration memory) while keeping low the error correction latency and the hardware cost, is proposed.

Various mitigation approaches have been proposed in the past especially for SRAM-based FPGAs [128]. Typical methods combine redundancy techniques for error detection with memory scrubbing for correction of the configuration memory upsets. In most SRAM-based FPGA families, as well as Zynq-7000, ECCs are embedded in the configuration frames to support the scrubbing mechanism. However, these frame-level ECCs are typically single-error correction, double error detection (SECDED) Hamming codes, and thus inadequate to correct multiple upsets in the configuration frames, as these COTS devices are designed for terrestrial applications. When two or more affected memory cells by a single event are located in a single configuration frame, so-called multiple-bit upsets (MBUs) (i.e., MBU is a subset of MCU, as MCU can expand in more than one adjacent frames), the SECDED code fails to correct them. For example, AMD Xilinx soft error mitigation (SEM) intellectual property (IP) [12] incorporates a frame-level cyclic redundancy check (CRC) scheme that enables only the correction of double-bit adjacent errors in the same frame for the Zynq-7000 families. The hybrid scrubbing approach proposed in [1] and [132] combines the aforementioned internal ECC-based scrubbing mechanism with an external scrubber. The internal scrubber continuously scans the configuration frames and corrects single-bit upsets (SBUs) or detects MBUs. The external scrubbing mechanism identifies the frames with MBUs by communicating with the internal scrubber, derives the golden frame data from an external storage medium and rewrites these frames. These approaches rely on an external radhard memory to store the entire golden configuration bitstream, which must be accessed in the case of

multibit errors, increasing both the memory storage requirements and the error correction latency.

Alternatively, other researchers have proposed frame-level redundant configuration scrubbing techniques [56], [139] or more sophisticated ECC schemes [42], [43], [98], [99], [112], [120], [147] to improve the FPGA correction capabilities for multibit errors. The redundancy techniques replicate the targeted design, thus adding multiple copies of the configuration frames, and check the memory applying majority voting between these redundant configuration frames to correct the upsets. These techniques can only be applied in designs with low programmable resource utilization. On the other hand, most of the ECC schemes consider the FPGA configuration memory as a two-dimensional (2-D) memory array and employ more complex codes, e.g., 2-D Hamming codes or bit-interleaved parity codes combined with erasure codes to protect the configuration memory from MBUs. All these approaches do not consider the ECC schemes embedded in the modern SRAM-based FPGAs and, thus, require extra hardware resources and memory storage to be implemented. Therefore, it is important to reduce these overheads to be tolerable in space applications with limited SWaP parameters.

In this thesis work, the idea of hybrid scrubbing [1], [132] which exploits the on-chip ECC logic with the concept of 2-D coding [42], [43], [98], [99], [112], [120], [147], that provides better error correction features, are combined to build a mixed 2-D coding technique. It combines the embedded ECC scheme with an interframe, interleaved parity code in a 2-D fashion. Assuming a 2-D arrangement of the FPGA configuration memory, the embedded ECC detects SBUs and MBUs per frame, while the combination of the ECC code (vertical direction) and the parity code (horizontal direction) achieves to correct the vast majority of single and multiple upsets affecting one or more frames. The proposed 2-D coding scheme's efficiency was demonstrated through heavy-ion radiation experiments. Additionally, to compare the approach with the state-of-the-art, two versions of the scrubbing scheme was implemented: a) an external scrubber that consists of a microcontroller, which runs the correction algorithm and communicates with the FPGA device through the "Joint Test Action Group" (JTAG) port, and b) an internal scrubber that integrates the entire solution on-chip.

1.2 Motivation

In summary, the following factors contributed to the initiation and implementation of this research work:

- Modern COTS SRAM-based FPGAs have recently emerged as a remarkable factor for the space avionics design providing a powerful computing platform for building low-cost, high-performance, miniaturized systems.

- The new high-performance processors architectures and silicon processes have offered the capability to integrate SoC and FPGA fabric into a single device, forming an even more powerful embedded computing heterogeneous APSoC platform due to the performance, power consumption, weight and volume benefits, hoping to pave the way for future space exploration missions that are becoming ever more demanding.
- The space industry has turned to miniature satellites, as especially in large numbers they may be more useful than fewer, larger ones for specific applications such as the collection of scientific data.
- The high-reliability requirements and the criticality of space applications, which are becoming increasingly demanding in computing power, make imperative to investigate in depth the vulnerability to radiation effects and to provide a more thorough reliability analysis for COTS APSoC devices since the majority of vendors has developed commercial powerful devices with very advanced computation capabilities in consumer electronics. The research so far has not performed the required experiments covering a range of heavy ion and proton energies to SEE characterize COTS SRAM-based devices, such as Zynq-7000 APSoC, in order to accurately predict error rates in a space radiation environment.
- The substantial shift of the space industry from space-grade devices, which are extremely expensive and less efficient than the COTS devices, to commercial powerful devices makes important to advance the state-of-the-art fault tolerant techniques by deploying more efficient hardening techniques. State-of-the-art approaches either do not use the ECC schemes integrated in the COTS devices, requiring additional hardware resources and memory storage for its implementation, or use them without considering that they are inadequate to correct multiple upsets in the configuration frames, as these COTS devices are designed for terrestrial applications.

1.3 Thesis Contributions

1.3.1 Contributions

The thesis performs an in-depth investigation of the SEE susceptibility under irradiation of all parts of a COTS APSoC, such as the AMD Xilinx Zynq-7000 APSoC, a detailed analysis of the radiation experimental results that produce a deep understanding for the SEE phenomena and a presentation of the development of an effective SEE mitigation approach to enable its deployment in radiation environments and meet the high-reliability requirements of space environments.

The work, published in "IEEE Transactions on Nuclear Science (TNS)" in 2020 [151] which is an extension of the work presented at the "Single Event Effects (SEE) Symposium coupled with the Military and Aerospace Programmable Logic Devices (MAPLD) Workshop" in 2019 [149], studies the impact of radiation-induced SEEs in the Zynq-7000 FPGA and presents a detailed analysis of the SEE vulnerability of the PL part of the Zynq-7000 architecture under very- and ultra- high-energy heavy ions by performing radiation experiments in two campaigns in different test facilities, CERN North Area (ultra-high energy ion beam) and GSI Helmholtz Centre for Heavy Ion Research (very-high energy ion beam), and for different heavy ions, energies, and LETs.

The major contributions of this work are the following:

a) The PL part of the Zynq-7000 device, which is based on 7-Series FPGA, can be conceptually split in two layers, namely design layer which includes the user logic and memory elements (e.g., FFs, SRLs and BRAM) and configuration layer which includes the logic and memory elements (e.g., CRAM) that allow configuring the logic and routing resources in the design layer. Thus, the PL integrate several different memories which contribute to the SEE vulnerability of the device. In this thesis work, the SEU cross sections for all these different memory types were calculated and analyzed.

b) The detailed analysis revealed various SEE phenomena such as SET-induced events in FFs and SRLs, single-event resets in FFs and BRAMs and SEFIs, which were either not observed in high-energy heavy-ion radiation experiments previously presented in the literature or not investigated in depth. Disclosure of these events is very important as they cannot be masked by conventional mitigation approaches such as triple modular redundancy (TMR) technique in which the resources are triplicated and a voting circuitry detects a potential fault. One of the above phenomena can affect more than one redundant replicas at the same time, thus the redundancy technique fails.

c) When a design is implemented on an FPGA, only a subset of the total available configuration memory cells is utilized. The memory bits corresponding to unused resources do not influence the functional behavior of the design and are thus considered non-essential. Consequently, soft error detection and correction should be concentrated on the subset of configuration essential bits, those that directly impact the implemented circuitry. An upset in one of these essential bits can alter the intended hardware functionality, potentially compromising system behavior. In this thesis work, the CRAM essential bit cross section was calculated, taking into account only the upsets in the essential configuration bits. Higher cross sections for the essential bits were reported, which may result to higher failure rate of the system and can be taken into account in the reliability analysis of the system given that the essential bit cross section provides a more realistic probability metric of SEUs affecting the programmable logic behavior.

d) A detailed analysis of the single-event multiple upsets (MBUs and MCUs) observed during radiation experiments in the various memory categories (e.g., CRAM, FFs, SRLs, BRAM) was made. This analysis may be beneficial for the design of effective SEU mitigation techniques. For example, the MBUs in the CRAM cannot be always corrected by the embedded ECC of the Zynq-7000 FPGAs and require a more sophisticated scrubbing scheme able to correct multiple errors. In addition, several critical observations from these should be considered in future designs of interleaving schemes, as they relate to device architecture and memory type.

Following the above work on the PL part of the Zynq-7000 under heavy ion irradiation [151], additional accelerated radiation experiments were carried out on both the Zynq-7000 PS and PL, with protons at Paul Scherrer Institute (PSI) and with heavy ions at GSI, to widen the range of tested particle energies and LETs and thus be able to achieve the full SEE characterization report, examine the radiation effects in all the SRAM arrays of the chip and predict upset and error rates more precisely providing useful reliability data for missions using Zynq-7000 device. The results of these experiments are presented at the "5th Space FPGA Users Workshop (SEFUW)" in 2023 [153].

Below are the main contributions of this work:

a) The first goal is to characterize the PL part for SEEs caused by protons, testing the chip for the full range of proton energies (30 to 250 MeV). Together with the results of the previous work, the results of this work cover a wide range of heavy ion LETs and the full range of proton energies to guarantee accurate prediction of the PL SEE rates.

b) The characterization of the PS part for SEEs caused by protons and heavy ions was performed. The SEU cross sections of the PS SRAM arrays (OCM, L1/L2 caches) by running static and dynamic memory tests were calculated.

c) The impact of the full range of proton energies and high-energy heavy ion radiation at the application level by running various processor benchmarks on the PS part for different cache configurations was studied. Different embedded microprocessor benchmarks with different memory footprints were executed, using four different cache configurations (i.e., all caches disabled, L1 only enabled, L2 only enabled, all caches enabled), to specify a representative picture of the errors that can occur in system data and system crashes in different conditions.

d) In this thesis work, the previous study of the MBU/MCU patterns (shapes) appearing on the PL under heavy ions [151] is extended for proton irradiation. Several works have been presented single-event multiple upsets for AMD Xilinx 7-Series devices (e.g., [39], [155]), but only for heavy ions. An analysis of the MBUs and MCUs observed for different energies calculating their number and frequency of occurrence per energy is made, using the results of the proton

radiation tests and providing further information for designing more comprehensive and thus more effective mitigation techniques.

e) Based on the radiation results, the error rates for three different orbits (e.g. LEO 800 km, 98°, LEO 400 km, 51.5° and MEO 23,222 km, 56°) were calculated using TRAD's OMERE software [141]. The predicted upset rates for different error categories for the PS and PL parts provide radiation constraints to use Zynq-7000 device in orbit.

The work, published in "IEEE Transactions on Nuclear Science" in 2022 [152] and the preliminary work presented at the "19th European Conference on Radiation and Its Effects on Components and Systems (RADECS)" in 2019 [150], proposes a configuration memory scrubbing approach for SRAM-based FPGA devices, which combines the embedded ECC logic with an interframe, interleaved parity code to build a mixed 2-D coding technique. The proposed technique improves the multiple-bit error correction capabilities of the on-chip ECC scheme while keeping the error correction latency and hardware cost low.

The mixed 2-D coding technique is based on the concept of the hybrid scrubbing approach presented in [1] and [132] which combines the internal ECC-based scrubbing mechanism of SRAM-based FPGAs with an external scrubber. The internal scrubber of the hybrid approach continuously scans the configuration frames, correcting SBUs or detecting MBUs, while its external scrubber identifies the frames with MBUs, derives the golden frame data from an external storage medium and corrects these frames by rewriting them. The innovation introduced by the mixed 2-D coding approach is that provides better error correction features. Its internal scrubber only performs fast error detection per frame, while its external scrubber identifies the frames with upsets and runs error correction algorithm, which relies on a 2-D coding concept presented in [42], [43], [98], [99], [112], [120], [147], using the ECC code as the vertical dimension and the interframe, interleaved parity code as the horizontal dimension, correcting all SBUs and the vast majority of MBUs.

Thus, the proposed scrubbing approach:

a) It eliminates the need for holding the golden bitstream externally as only the parity bits should be stored in a separate memory.

b) It keeps the error correction latency delimited as the correction process depends only on the execution of the correction algorithm.

c) It avoids wrong corrections, as the internal scrubber of the hybrid approach incorrectly repair an odd-numbered MBU interpreted as SBU since the embedded ECC used in correction is a SECDEC code. In addition, the external scrubber of the proposed approach does not need extra time to identify this situation and scrub the entire frame like the hybrid approach.

In addition to the above advantages of the proposed technique, an extra benefit is that it is an idea that can be applied in all the SRAM-based FPGA families featuring on-chip ECC (e.g., AMD Xilinx Virtex-5, Virtex-6, 7-Series and UtraScale, and Intel Stratix and Arria series). The scrubber design, with the appropriate adaptation to the interface and features of the ECC module of the FPGA family, can operate as a complete mitigation mechanism. The only thing to consider is the error detection and correction capabilities of each ECC and the different ECC results that must be accessed and processed by the scrubber as well as the access interface to the FPGA configuration memory to read the erroneous frames and write back the corrected ones.

In this thesis work, AMD Xilinx Zynq-7000 FPGA, based on the AMD Xilinx 7-Series FPGA architecture, is selected for demonstration purposes. Its built-in frame-level ECC module uses syndrome of 13 bits and generates the information related to the detected errors (frame address, syndrome, and error type) to feed the scrubber. In turn, the scrubber must have access to the FPGA configuration memory to correct the erroneous frames either through the internal configuration access port (ICAP) or the external JTAG configuration interface. On the contrary, Intel Stratix 10 FPGA family uses a 32-bit frame-level CRC, stores the error messages, including sector address, error location and error type, in a 32-bit eight-entry error message queue and can be configured only through its JTAG port. Thus, the scrubber should be modified in order to access the error message queue and the JTAG interface of the Stratix 10 FPGA.

The scrubbing concept has been validated under heavy-ion irradiation using a Zynq-7000 FPGA device at CERN, where all the SBUs and MBUs observed in the configuration memory were corrected. Moreover, to compare the approach with the state-of-the-art in terms of area overhead and error correction latency, two different versions of the scrubbing approach were implemented for the Zynq-7000 FPGA architecture.

The first one uses an external microcontroller in the role of scrubber running the error detection and correction algorithm in software, which communicates with the ECC logic and performs access to the configuration memory of the target FPGA through its JTAG port (external scrubber version). This solution combines the popularity of the JTAG interface with the software versatility provided by the external microcontroller. In this thesis, the low-cost Digilent Zybo board [38] which integrates an AMD Xilinx Zynq-7010 APSoC device is chosen to demonstrate the external scrubbing solution and measure its efficiency with respect to the state-of-the-art approaches. However, a space-grade microcontroller, e.g., the Gaisler GR716 [83] or the BAE Systems RAD750 [27], could also play this role with slightly performance degradation due to only a small portion of the error correction latency is consumed in the ECC algorithm run in the processor. Note that if a non-space-grade microcontroller is selected, e.g., the Zynq-7000 APSoC, radiation hardening techniques should be considered. The second version is a full-hardware solution that integrates the 2-D error detection and correction (EDC) algorithm on-chip (internal scrubber). The main advantage of the internal scrubber, which eliminates the need for an external controller, is the faster execution of the EDC

algorithm, resulting in higher system availability. It must be noted that all hardware logic of the internal scrubber is protected against soft errors using the TMR technique, and the built-in ECC feature of the BRAM is activated. The internal scrubber version outperforms the external version in terms of error correction latency, thus the internal scrubber solution would be selected for applications with high availability requirements. On the other hand, for applications where system reliability is more valuable than fast error correction, the external version can be adopted.

Compared to all the state-of-the-art approaches, both implementations outweigh in terms of correction latency or area overhead or both. Both external and internal scrubbers have an almost constant error correction latency compared with the approaches that use the internal ECC for error correction (e.g., [1], [12], [132]). External scrubber also eliminates the need for storing the entire golden bitstream in contrast to the hybrid scrubbing approaches [1] and [132]. On the other hand, the internal scrubber has lower error correction latency than SEM [12], as well as parity techniques [43] and [112]. Finally, although both proposed approaches may have higher error correction latency and memory overhead than the parity technique presented in [147], the additional circuit required by the latter is large enough.

For the radiation and fault-injection campaigns carried out during the above works, an open-source platform had been developed by the Embedded Systems Laboratory (ESLab) team [44], of which I am member, called FREtZ (FPGA Reliability Evaluation through JTAG) [145], which was presented at the “Workshop on Open Source Design Automation (OSDA)” in conjunction with “Design, Automation and Test in Europe Conference (DATE)”, in 2019 [125]. The open-source FREtZ tool provides a rich set of high-level Python application programming interfaces (API) and application examples to communicate with all AMD Xilinx 7-Series/Zynq-7000 and UltraScale/UltraScale+ FPGAs/APSoC devices via the JTAG protocol. FREtZ increases the productivity of performing fault injection and radiation experiments by hiding low-level AMD Xilinx Vivado TCL/JTAG commands that are executed behind the scenes to access the PS and PL memories of the target device.

This open-source framework provides several functions to facilitate various reliability evaluation and improvement methodologies, such as fault injection, memory scrubbing, memory recording during radiation experiments, etc. It mainly consists of:

a) an on-chip logic in the device under test (DUT) to provide access to the FPGA configuration memory and its embedded ECC logic, and APSoC embedded memories using the advanced extensible interfaces (AXI) between the PS and the PL,

b) a software library of low-level JTAG functions in Vivado TCL commands using transmission control protocol (TCP) client-server scheme to access the DUT and

c) a set of high-level configuration functions (graphical user interface - GUI) to enable the development of target applications, consisting of the user application and the interface functionality with the low-level JTAG engine.

FREtZ, as an open-source framework, is low-cost, non-intrusive tool, like other approaches, since it requires neither specific hardware platform nor modifications to the DUT, except the minimum integrated logic to provide access to the target device. In addition, due to its generic and open architecture, the framework is extensible without depending on the DUT architecture being able to support new device families. It is also user friendly since its generic GUI allows easy development of the target reliability applications.

The client-server architecture of the proposed framework enables the development of parallel multi-server and distributed approaches in order to handle the concurrent activation of several host processes or the connection to multiple target devices. Thus, either an application can communicate with multiple servers targeting different DUTs or multiple user applications running in the same platform can communicate with one server targeting the same DUT. This idea allows the deployment of a parallel fault injection platform into multiple identical target devices to reduce testing time as proposed in [51].

To speed up the radiation and fault-injection campaigns, especially on devices with more resources in general, an additional embedded SoC platform was designed to replace the software library of JTAG functions that use Vivado commands. It is based on the low-cost commercial Digilent Zybo Z7 development board [38] and provides both high-speed JTAG communication and software programmability. It combines software modules running on the FreeRTOS embedded real-time operating system with a JTAG hardware controller both implemented in Zynq-7010 APSoC. Using FreeRTOS makes it easy to create software modules so that they can run in parallel and independently of each other, as well as to further develop the platform and add other modules in the future. A simple API provides a highly programmable interface for creating custom JTAG sequences, to implement applications such as full configuration, readback, scrubbing, fault injection, and design monitoring. The hardware controller module consists of a state machine, which optimizes the timing and speed of JTAG communication, and multiple buffers, which maximizes the bandwidth of large JTAG transfers. The high-speed JTAG signal generation is much faster and less processor intensive than it would be in software, allowing for a maximum JTAG bandwidth.

The contributions of this thesis have been published in two publications in a top-tier scientific journal, IEEE Transactions on Nuclear Science, in two conferences: RADECS, the leading annual European conference focusing on radiation effects and SEE/MAPLD, the premier US symposium focusing on Single Event Effects (SEE), and in two workshops (SEFUW, OSDA), as listed in 1.3.2, using peer review process. In addition, three test reports (1.3.3) have been approved and included in European Space Agency (ESA) Radiation Test Database. These reports constitute three of the four total reports

concerning AMD Xilinx devices in this Database. Finally, a number of publications, listed in 1.3.4, present work with collaborators which is related with the research in this thesis.

1.3.2 Main Publications

1. Vasileios Vlagkoulis, Dimitrios Agiakatsikas, Aitzan Sari, Ioanna Souvatzoglou, Georgios Antonopoulos, Mihalis Psarakis, Cesar Boatella-Polo, Alessandra Costantino, Viyas Gupta, Antonios Tavoularis, Gianluca Furano, Christian Poivey, Véronique Ferlet-Cavrois, Tim Wagner and Hajdas Wojciech, "Estimating the Error Rates of Xilinx Zynq-7000 APSoCs in Space Radiation Environments," in 5th SpacE FPGA Users Workshop (SEFUW), ESTEC, Noordwijk, The Netherlands, March 2023. [Online]. Available: https://indico.esa.int/event/439/contributions/7941/attachments/5183/8216/SEFUW_v1.1.pdf.
2. Vasileios Vlagkoulis, Aitzan Sari, Georgios Antonopoulos, Mihalis Psarakis, Antonios Tavoularis, Gianluca Furano, Cesar Boatella-Polo, Christian Poivey, Véronique Ferlet-Cavrois, Maria Kastriotou, Pablo Fernandez Martinez and Rubén García Alía, "Configuration Memory Scrubbing of SRAM-Based FPGAs Using a Mixed 2-D Coding Technique," in IEEE Transactions on Nuclear Science, vol. 69, no. 4, pp. 871-882, April 2022, doi: https://doi.org/10.1109/TNS.2022.3151977.
3. Vasileios Vlagkoulis, Aitzan Sari, John Vrachnis, Georgios Antonopoulos, Nikolaos Segkos, Mihalis Psarakis, Antonios Tavoularis, Gianluca Furano, Cesar Boatella-Polo, Christian Poivey, Véronique Ferlet-Cavrois, Maria Kastriotou, Pablo Fernandez Martinez, Rubén García Alía, Kay-Obbe Voss and Christoph Schuy, "Single Event Effects Characterization of the Programmable Logic of Xilinx Zynq-7000 FPGA Using Very/Ultra High-Energy Heavy Ions," in IEEE Transactions on Nuclear Science, vol. 68, no. 1, pp. 36-45, January 2021, doi: https://doi.org/10.1109/TNS.2020.3033188.
4. Vasileios Vlagkoulis, Aitzan Sari, Juljan Proko, Dimitrios Zografakis, Mihalis Psarakis, Antonios Tavoularis, Gianluca Furano, Cesar Boatella-Polo, Christian Poivey, Véronique Ferlet-Cavrois, Maria Kastriotou, Pablo Fernandez Martinez and Rubén García Alía, "Configuration Memory Scrubbing of the Xilinx Zynq-7000 FPGA using a Mixed 2-D Coding Technique," 2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Montpellier, France, September 2019, doi: https://doi.org/10.1109/RADECS47380.2019.9745693.
5. Vasileios Vlagkoulis, Aitzan Sari, John Vrachnis, Georgios Antonopoulos, Nikolaos Segkos, Mihalis Psarakis, Antonios Tavoularis, Gianluca Furano, Cesar

Boatella-Polo, Christian Poivey, Véronique Ferlet-Cavrois, Maria Kastriotou, Pablo Fernandez Martinez and Rubén García Alía, "Analysis of the Single Event Upsets in the Programmable Logic of 28 nm Xilinx Zynq-7000 FPGA due to Heavy Ion Irradiation", Single Event Effects (SEE) Symposium and Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, San Diego, USA, May 2019. [Online]. Available: https://www.seemapld.org/archive/2019/0522_WED/1040%20-%20SEE-MAPLD19_Psarakis_Zynq_Radiation.pdf.

6. Aitzan Sari, Vasileios Vlagkoulis and Mihalis Psarakis, "An Open-source Framework for Xilinx FPGA Reliability Evaluation," Workshop on Open Source Design Automation (OSDA) 2019, in conjunction with Design, Automation and Test in Europe Conference (DATE), pp 1-6, Florence, Italy, March 2019. [Online]. Available: https://osda.gitlab.io/19/3.2.pdf.

1.3.3 Test Reports

1. Mihalis Psarakis, Vasileios Vlagkoulis, "Radiation Test Report: Single-Event Effects (SEEs) characterization of the Xilinx Zynq-7000 APSoC device under proton irradiation," ESA Radiation Test Database, January 2022. [Online]. Available: https://esarad.esa.int .
2. Vasileios Vlagkoulis, Mihalis Psarakis, "Single Event Upsets (SEU) Test Report: Single-Event Upsets Characterization of the 28nm Artix-7-based Programmable Logic of Xilinx Zynq-7000 FPGA," ESA Radiation Test Database, July 2019. [Online]. Available: https://esarad.esa.int .
3. Vasileios Vlagkoulis, Mihalis Psarakis, "Single Event Upsets (SEU) Test Report: Single-Event Upsets Characterization of the 28nm Kintex-7-based Programmable Logic of Xilinx Zynq-7000 FPGA," ESA Radiation Test Database, July 2019. [Online]. Available: https://esarad.esa.int .

1.3.4 Ancillary Publications

1. Ioanna Souvatzoglou, Dimitris Agiakatsikas, Vasileios Vlagkoulis, Aitzan Sari, Mihalis Psarakis, Maria Kastriotou and Carlo Cazzaniga "Assessing the Reliability of FPGA-Based Quantized Neural Networks Under Neutron Irradiation," in IEEE Transactions on Nuclear Science, vol. 71, no. 12, pp. 2565-2577, December 2024, doi: https://doi.org/10.1109/TNS.2024.3491503.
2. Dimitris Agiakatsikas, Nikos Foutris, Aitzan Sari, Vasileios Vlagkoulis, Ioanna Souvatzoglou, Mihalis Psarakis, Ruiqi Ye, John Goodacre, Mikel Lujan, Maria Kastriotou, Carlo Cazzaniga and Chris Frost, "Single Event Effects Assessment of UltraScale+ MPSoC Systems under Atmospheric Radiation," in IEEE

Transactions on Reliability, vol. 73, no. 1, pp. 771-783, March 2024, doi: https://doi.org/10.1109/TR.2023.3312548.

3. Ioanna Souvatzoglou, Dimitris Agiakatsikas, Aitzan Sari, Vasileios Vlagkoulis, Antonios Tavoularis and Mihalis Psarakis, "Tradeoff Between Performance and Reliability in FPGA Accelerated DNNs for Space Applications," 2023 European Data Handling & Data Processing Conference (EDHPC), Juan Les Pins, France, October 2023, doi: https://doi.org/10.23919/EDHPC59100.2023.10396101.
4. Ioanna Souvatzoglou, Dimitris Agiakatsikas, George Antonopoulos, Vasileios Vlagkoulis, Aitzan Sari, Athanasios Papadimitriou and Mihalis Psarakis, "The Impact of Hardware Folding on Dependability in Spaceborne FPGA-based Neural Networks," 2022 International Conference on Field-Programmable Technology (ICFPT), Hong Kong, December 2022, doi: https://doi.org/10.1109/ICFPT56656.2022.9974551.
5. Ioanna Souvatzoglou, Athanasios Papadimitriou, Aitzan Sari, Vasileios Vlagkoulis and Mihalis Psarakis, "Analyzing the Single Event Upset Vulnerability of Binarized Neural Networks on SRAM FPGAs," 2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Athens, Greece, October 2021, doi: https://doi.org/10.1109/DFT52944.2021.9568280.
6. Dimitris Agiakatsikas, Nikos Foutris, Aitzan Sari, Vasileios Vlagkoulis, Ioanna Souvatzoglou, Mihalis Psarakis, Mikel Lujan, Maria Kastriotou and Carlo Cazzaniga, "Evaluation of Xilinx Deep Learning Processing Unit under Neutron Irradiation," 2021 European Conference on Radiation and Its Effects on Components and Systems (RADECS), Vienna, Austria, September 2021, doi: https://doi.org/10.1109/RADECS53308.2021.9954522.
7. Ioannis Tsounis, Antonis Tsigkanos, Vasileios Vlagkoulis, Mihalis Psarakis, Nektarios Kranitis and Antonios Paschalis, "Analyzing the Resilience to SEUs of an Image Data Compression Core in a COTS SRAM FPGA," 2019 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Colchester, United Kingdom, July 2019, doi: https://doi.org/10.1109/AHS.2019.000-5.

1.4 Thesis Organization

The remainder of this thesis is organized into six chapters, each addressing critical aspects of the study on the effects of radiation on COTS heterogeneous programmable devices, with a focus on the Zynq-7000 APSoC. The content progresses from foundational concepts to experimental evaluations and proposed mitigation techniques.

In chapter 2, a detailed overview of COTS programmable devices is provided, with a focus on FPGA and APSoC architectures. The Zynq-7000 APSoC architecture is

described in depth, including its processing system (PS) and programmable logic (PL) components. Chapter 3 reviews the space radiation environment and its impact on electronic components. It explains various radiation-induced effects, including cumulative and single-event effects, and presents common mitigation techniques. Validation methods such as fault injection and radiation testing are also discussed.

The following three chapters present the major contributions of this thesis. In chapters 4 and 5, the experimental setup and results of radiation testing performed on the PL and PS sections of the Zynq-7000 device, respectively. Reliability analysis is conducted under exposure to heavy ions and protons to characterize the device's vulnerability. Additionally, at the end of chapter 5, the upset rates for the device in the context of modeled space environments are estimated, taking into account the experimental results of both PL and PS.

In chapter 6, an error detection and correction mechanism is proposed and evaluated for the configuration memory. Both internal and external scrubber implementations are discussed. The chapter presents reliability analysis, compares the techniques with state-of-the-art methods, and evaluates their performance through radiation testing.

The last chapter concludes this thesis. It summarizes the main findings and contributions of the research. It discusses the implications of the results and outlines potential directions for future research, particularly in improving the resilience of COTS programmable devices for space missions.

2 COTS Programmable Devices

2.1 Field Programmable Gate Arrays

FPGAs are advanced digital integrated circuits that can be programmed or reprogrammed in the field after manufacturing. It consists of an array of configurable (programmable) logic blocks along with interconnects between these blocks. Designers can configure such devices to perform a variety of digital functions. They contain up to millions of logic gates, and thus, they can be used to implement even extremely large and complex functions that previously could be realized only using application-specific integrated circuits (ASICs). FPGA configuration is generally specified using a hardware description language (HDL), i.e., a modular programming code used to describe the structure and behaviour of digital logic circuits.

The first FPGAs appeared in the 1980s and were primarily used to implement small amounts of simple logic and medium-complexity state machines. Over the next decade, as the size and complexity of FPGAs began to increase, they were used primarily in telecommunications and networking applications, both of which involved processing large blocks of data. By the early 2000s, high-performance FPGAs led to an explosion of their use in consumer, automotive and industrial applications. Nowadays, FPGAs can be used to implement just about anything, especially in applications where flexibility, speed, and parallel processing capabilities are required, such as in telecommunications, industrial, automotive, defence and aerospace sectors.

The basic advantage of FPGAs is that they can perform calculations in parallel and, thus, faster than general-purpose central processing units (CPUs), which run software code in a sequential fashion. Although developers can write software to take advantage of multiple CPUs, the massive inherent parallelism of an FPGA can improve system performance even more since all the pieces are just hardware. Moreover, processing units that perform calculations in a sequential manner tend not to be used in time-critical applications. On the other hand, FPGAs can be designed to include multiple blocks processing data in parallel, offering much greater scalability, as well as adaptability to time-critical applications.

One of the most prominent advantages of FPGAs over other processing platforms lies in their inherent reprogrammability. This capability allows the device to be modified or entirely reconfigured even after deployment, enabling a change in functionality without altering the physical hardware. As a result, FPGAs offer a highly flexible solution that supports updates, upgrades, or functional repurposing, thereby significantly reducing the costs and efforts associated with long-term maintenance and hardware obsolescence. Instead of replacing outdated components or redesigning new hardware, the same FPGA can be reprogrammed in the field to meet evolving system requirements. Moreover, modern FPGAs support partial reconfiguration, allowing specific regions of the device to be reprogrammed dynamically while the remaining logic continues to operate uninterrupted, which is particularly advantageous in adaptive or resource-constrained applications.

Although FPGAs may have higher per-unit costs compared to other hardware platforms, their ability to be reconfigured multiple times using accessible development tools renders them highly cost-effective over the product lifecycle. This reprogrammability, combined with their reusability, makes FPGAs particularly well-suited for prototyping applications, especially in the context of ASIC validation. The flexibility offered by FPGAs enables iterative testing and design refinement, allowing engineers to evaluate various design configurations and optimize system performance prior to committing to a final hardware implementation. This iterative capability significantly accelerates the development cycle and reduces the risks and costs associated with ASIC fabrication.

2.1.1 FPGA Architecture

Each FPGA vendor has its own FPGA architecture, but in general terms, they are all variations of that shown in Figure 2.1. The architecture mainly consists of configurable logic blocks arranged in a two-dimensional grid, configurable input/output blocks arranged at the periphery of the grid to implement the connection with external components, and programmable routing resources which interconnect the logic blocks and input/output blocks creating a configurable logic mesh.

The configurable logic block is a basic component of an FPGA that provides the basic logic and storage functionality for a target application design, such as flip-flops (FF) and lookup tables (LUTs). FPGAs also contain clock circuitry for driving the clock signals to each logic block, including clock trees and clock management blocks, which provide clock frequency synthesis, deskew, and jitter filtering functionality. Additional logic resources such as DSPs, memories, decoders and gigabit transceivers may also be available.

FPGA logic is configured by loading application-specific configuration data into internal memory. This internal memory, called configuration memory, stores all the necessary information to program the device. The programmable blocks and interconnections are configured by the configuration file, i.e., a group of configuration bits loaded in the

configuration memory for defining a specific circuit previously described with an HDL model. A variety of tools and flows for implementing FPGA designs, created by each FPGA vendor, generate the configuration file.

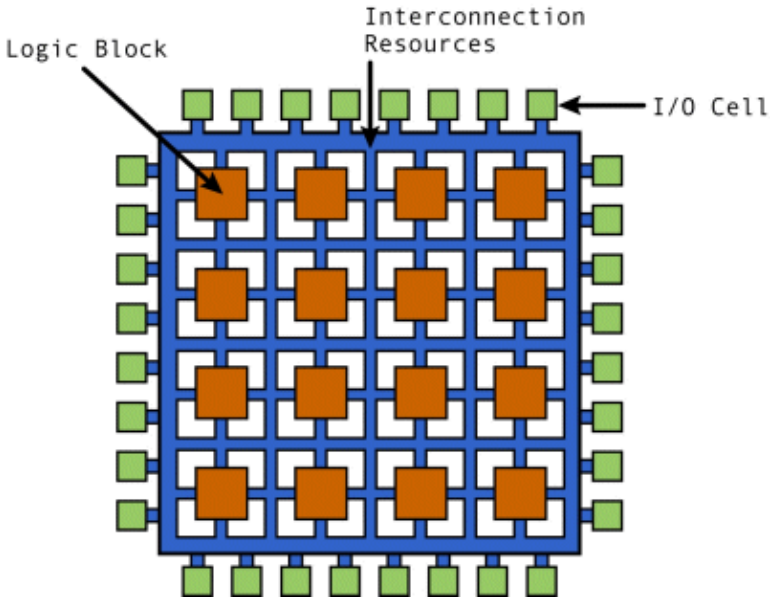


Figure 2.1: Generic FPGA logic architecture [47]

The configuration file comprises both configuration data, which defines the state of the programmable logic elements, and configuration commands, which provide the necessary instructions for interpreting and applying this data. When the configuration file is being loaded into the FPGA device, the information being transferred is referred to as the configuration bitstream. The mechanisms for programming FPGAs may vary by family.

2.1.2 FPGA Technologies

Various programming technologies have been used for reconfigurable architectures. Each of these technologies has different characteristics, which in turn have significant effects on the programmable architecture. The most well-known types of programmable elements for FPGAs used in the space environment are antifuse, flash, and SRAM.

Antifuse technology is the opposite of a regular fuse. Whereas a fuse starts with low resistance and is designed to permanently break or open an electrically conductive path, when the current through the path exceeds a specified limit, an antifuse is normally an open circuit until a programming current that exceeds a certain level converts it into a permanent electrically conductive path.

Antifuse-based FPGAs are one-time programmable (OTP) in contrast to other FPGA technologies, which may be reprogrammed to fix logic bugs or add new functions. Antifuse devices are non-volatile, i.e., they do not need to be configured each time the device is powered on, and they may be less susceptible to alpha particles, which can cause circuits to malfunction. The antifuse programming mechanism provides good data retention performance and ensures secure data storage against physical attacks, making this device inherently very secure. But, the complex, nonstandard fabrication process of an antifuse device has turned out to be a key disadvantage as antifuse FPGAs have lower yields and the technology has improved more slowly than, e.g., SRAM-based FPGAs.

Flash-based FPGAs are also non-volatile devices, meaning they don't require an external memory to store their configuration data. Their memory cells hold the configuration pattern on the chip, and even if power is removed the contents of the flash cells stay intact. Additionally, flash-based FPGAs, unlike antifuse-based, can be electrically erased and reprogrammed. Advances in process technology allow FPGA designers to shrink the flash configuration cells and integrate them into advanced logic processes. This evolution enables high-performance flash-based FPGAs to deliver a wealth of features and functions, often at a lower system cost. Thus, flash-based FPGAs combine the advantages of the antifuse technology with the ability to be reprogrammed.

In contrast, SRAM-based FPGAs are volatile and need to be reprogrammed at each power-up. The static qualifier associated with SRAM refers to the memory's ability to retain data as long as power is continuously supplied; however, once power is removed, the stored data is lost. Since the SRAMs are reprogrammable, these FPGAs can be reconfigured rapidly and repeatedly, including in-system reprogramming, in a manner analogous to standard SRAM write operation.

While SRAM-based FPGAs typically exhibit higher power consumption and pose greater security concerns, owing to the necessity of external configuration bitstream susceptible to interception, their advantages have led to widespread adoption, making them the predominant FPGA technology. This widespread usage is largely attributed to the continuous evolution of CMOS technology, enabling high performance and increased logic density. In addition, FPGA manufacturers benefit from the substantial research and development investments made by companies specializing in memory technologies. Since SRAM cells are fabricated using the same CMOS process as the rest of the FPGA, no additional manufacturing steps are needed, thereby simplifying production.

The six CMOS transistors SRAM cell design used by most conventional COTS devices is depicted in Figure 2.2. The bit information within a six-transistor SRAM cell is stored in two cross-coupled inverters composed of four transistors. Each inverter contributes to keeping the other inverter's value alive. The two additional transistors are required to provide access control via the word line (WL). This design does not make SRAM the densest storage architecture but allows fast read and write operations, i.e., of only a few nanoseconds.

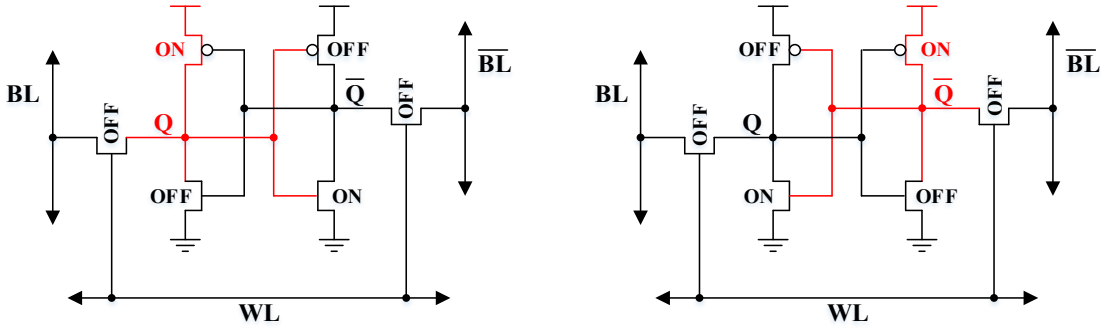


Figure 2.2: SRAM cell composed of six CMOS transistors

As long as the signal strength of both Q and $\overline{Q}$ (Figure 2.2) does not exceed the threshold limit of the inverter system, the stored configuration value of an SRAM cell remains untouched, assuming the system stays powered. If both exceed this limit, the cell inverters get programmed with the recently set values. This usually is the case when the word line is selected and both bit lines (BL) are driven strong with a value during a cell write request. In case of a read request, both bit lines will be pre-charged weakly to accept the SRAM cell's currently stored values Q and $\overline{Q}$ as soon as the word line is selected. The concept of cross-coupled inverters has been chosen to reduce the overall susceptibility to signal noise. A failure within this system may lead to incorrectly stored states and, therefore, upsets within the SRAM cell. More information regarding such effects can be found in section 3.2.

2.2 All Programmable SoC Devices

The explosion of silicon technology in recent years has allowed the integration of complex systems into a single chip. The classic design method is to make the various components needed for a system and then assemble and connect them at system level. The integration concept is that a single silicon chip can be used to implement the functionality of an entire system rather than requiring several different physical chips on a board.

A system-on-chip (SoC) integrates all essential components of a digital system, such as processing units, high-speed logic, interface circuitry, memory elements, and more, onto a single semiconductor substrate. Traditionally, these functionalities would be implemented using multiple discrete components, interconnected at the printed circuit board (PCB) level. In contrast, the SoC approach offers several significant advantages: it reduces overall system cost, enhances data transfer speed and security between

integrated modules, and delivers improved performance in terms of processing speed, power efficiency, physical footprint, and system reliability.

State-of-the-art SoCs are based on multicore technologies coupled with accelerators in a heterogeneous environment, offering a great potential of computing power for scientific and industrial applications. They combine the advantages of FPGAs and SoCs and high bandwidth communication between them into a single device that forms an even more powerful integrated heterogeneous device, called all programmable SoC (APSoC) device. In general, APSoCs consist of two main parts, the programmable logic (PL) which is equivalent to an FPGA and the processing system (PS), which includes a single or multicore processor. It also features integrated memory, a variety of peripherals, and high-speed communications interfaces (Figure 2.3).

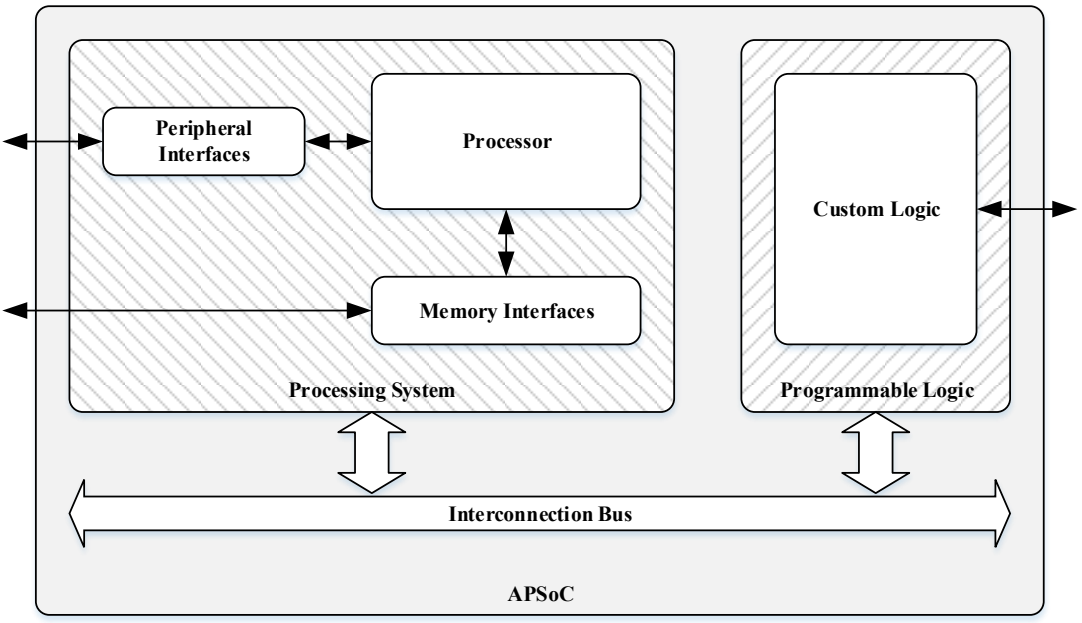


Figure 2.3: Generic APSoC architecture

The PL section is particularly well-suited for implementing high-speed logic, arithmetic operations, and data flow subsystems, whereas the PS is designed to support software execution, including both application code and operating systems. This separation of architecture facilitates an effective hardware/software co-design approach, enabling the deliberate distribution of system functionality across programmable hardware and software domains.

Software executed on the PS may operate in bare-metal mode, offering minimal abstraction for direct interaction with hardware, or within a full-featured OS environment. Communication between the PS and PL is facilitated through various interconnect architectures, which may include direct point-to-point links or shared bus

structures capable of supporting multiple components. PS can also feature on-chip memory and a variety of external memory and peripheral interfaces, e.g., industry-standard interfaces for external data communication.

These characteristics make APSoCs suitable and attractive for safety-critical applications like space and avionics. Especially, APSoCs with SRAM-based FPGA fabric are becoming the state-of-the-art space technology used in multiple space programs. Using run-time reconfiguration in the space environment allows designers to modify on-board hardware by replacing faulty/outdated designs at different stages of a mission. Some example applications are rectification of design faults, alteration of system functionality in response to changed mission requirements, support of multitasking satellites capable of switching, improvement of processing algorithms, etc. Reprogrammability in flight can even be used to debug FPGAs in orbit.

The CHREC Space Processor (CSPv1) [121] is an example of SRAM-based APSoCs employment in space applications. CSPv1 is a small computer designed to operate in low-cost space missions powered by an AMD Xilinx Zynq-7000 APSoC. CSPv1 relies on a combination of commercial and rad-hard components, in which the commercial components perform critical computations but are monitored by the rad-hard components. Additionally, in [78], a platform named Advanced Processor for space Exploration (APEX-SoC) is proposed which is aligned with the new trend for using state-of-the-art COTS SoC technology in space applications. APEX-SoC is a generic platform based on a Zynq-7000 device intended to control scientific instruments in future NASA missions.

2.3 Zynq-7000 APSoC Architecture

Zynq-7000 is a family of COTS SRAM-based APSoC devices designed by AMD which combines both FPGA fabric and a capable applications processor. It is built on 28 nm high-k metal-gate CMOS process technology [16], [35].

The general architecture of the Zynq-7000 comprises two sections (Figure 2.4), the PS and the PL. They can be used independently or together, as they afford separate power circuitries allowing either the PS or PL to be powered down if not in use. Nevertheless, the full potential of the Zynq-7000 is realized when both parts operate collaboratively. As such, understanding the internal structure of each section, as well as the interconnect mechanisms between them, is essential. PS contains a dual- or single-core ARM Cortex-A9 CPU and PL is based on the Artix-7 or Kintex-7 FPGA fabric. The devices of the Zynq-7000 series and their features are shown in Table 2.1.

The ARM Cortex-A9 cores [24] are the heart of the PS, which also includes on-chip memory, external memory interfaces, and a rich set of I/O peripherals. Apart from CLBs, DSPs and BRAMs, the PL also contains IOBs for interfacing with the external world,

gigabit transceivers and a flexible analog-to-digital converter with programmable logic to address a broad range of analog data acquisition and monitoring functions. The two parts of Zynq-7000 communicate using functional interfaces, including AXI interconnect, extended multiplexed I/O (EMIO) interfaces for most of the I/O peripherals, interrupts, DMA flow control, clocks, and debug interfaces.

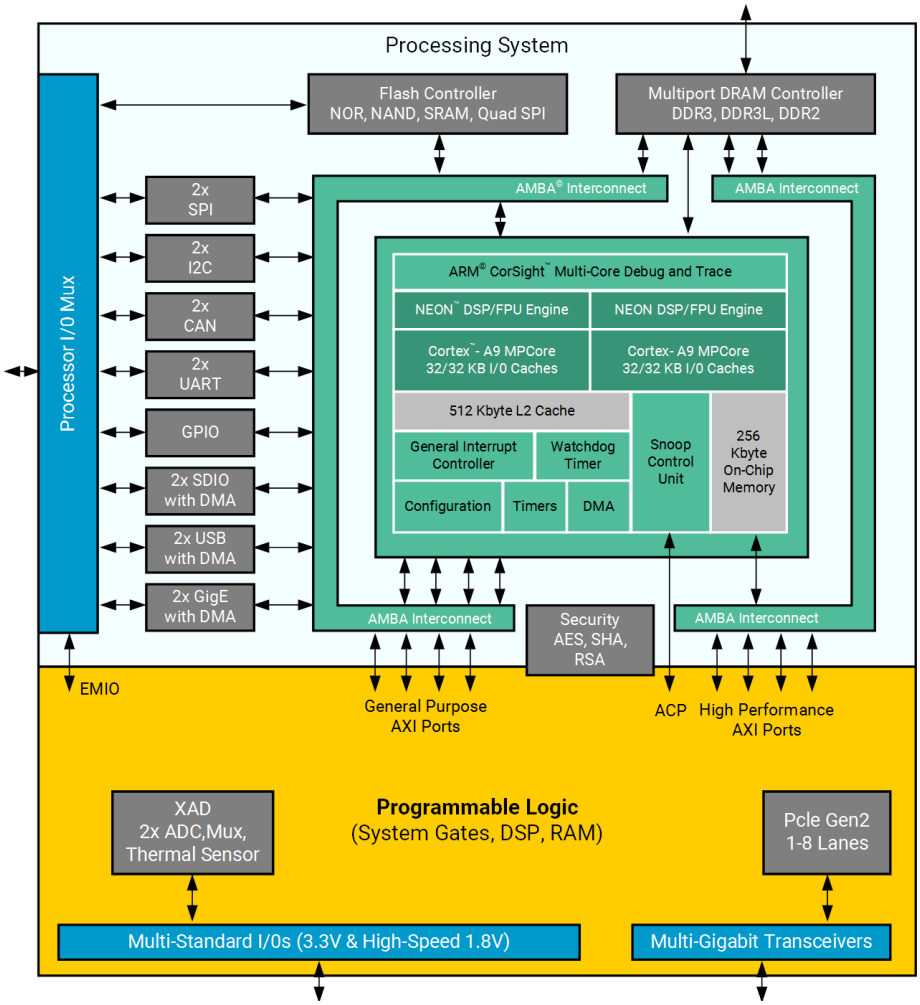


Figure 2.4: Zynq-7000 APSoC overview [15]

The processor(s) in the PS always boot first, allowing a software-centric approach for PL system boot and PL configuration. The PL can be configured either as part of the boot process or later. Additionally, the PL can operate uninterruptedly while a portion of it is being reconfigured, allowing dynamic partial reconfiguration.

Zynq-7000 device	PS core	PS frequency	PL 7-Series
7z007s, 7z012s, 7z014s	Single-core ARM Cortex-A9	667 - 766 MHz	Artix-7 FPGA
7z010, 7z015, 7z020	Dual-core ARM Cortex-A9	667 - 866 MHz	Artix-7 FPGA
7z030, 7z035, 7z045, 7z100	Dual-core ARM Cortex-A9	667 - 1000 MHz	Kintex-7 FPGA

Table 2.1: Zynq-7000 family

2.3.1 Zynq-7000 PS

All Zynq-7000 devices share a common architectural foundation, with each integrating either a single-core or dual-core ARM Cortex-A9 processor conforming to the ARMv7 architecture as the central component of the PS. However, the PS encompasses more than just the ARM processor; it includes a comprehensive suite of processing and support components collectively referred to as the application processing unit (APU). This unit is accompanied by various peripheral interfaces, cache memory, memory controllers, interconnect infrastructure, and clock generation modules. A detailed block diagram illustrating the structure of the PS is provided in Figure 2.5, with the APU highlighted in green.

The APU is primarily comprised of ARM processing core(s) with associated computational units, such as a NEON media processing engine and floating-point unit (FPU), a memory management unit (MMU) for virtual to physical address translation, and two Level 1 (L1) cache memories of 32KB each for local storage of frequently required data (D-cache) and instructions (I-cache).The APU also contains a larger Level 2 (L2) cache memory of 512KB for instructions and data, and a 256KB on-chip integrated memory (OCM) that provides a low-latency memory. Finally, a snoop control unit (SCU) forms a bridge between the ARM cores and the L2 and OCM memories.

Typically, every processor has inside its data path a set of general-purpose registers called register file. In case of Zynq-7000's ARM Cortex-A9 dual-core processor, each core has a register file consisting of 15 registers and the program counter (PC) [24]. The register file is the smallest addressable memory and it is located at the top of the memory hierarchy, allowing a register to drive its data onto an internal bus in a single clock cycle.

Each Cortex-A9 processor has separate L1 instruction and data caches, as shown in Figure 2.5. The L1 cache is the smallest and fastest cache memory of the Zynq-7000. It is built into the fabric of each ARM core and, therefore operates at the same clock frequency. Similarly, to the register file, L1 cache is not a general memory and is not accessible via any system bus. Each L1 cache can be enabled/disabled independently.

The L2 cache is external to the processing core(s), and in the case of dual-core APU, it is shared by the two cores. It is larger than L1 cache, but has slower access speeds. L2

cache is unified storing both instructions and data, unlike L1 which is split into separate instruction cache (I-cache) and data cache (D-cache). All accesses through the L2 cache controller can be routed to the DDR controller or can be sent to other slaves in the PS or PL depending on their address. To reduce the DDR memory access latency, there is a dedicated port from the L2 controller to the DDR controller. The L2 cache is disabled by default and can be enabled independently of the L1 caches.

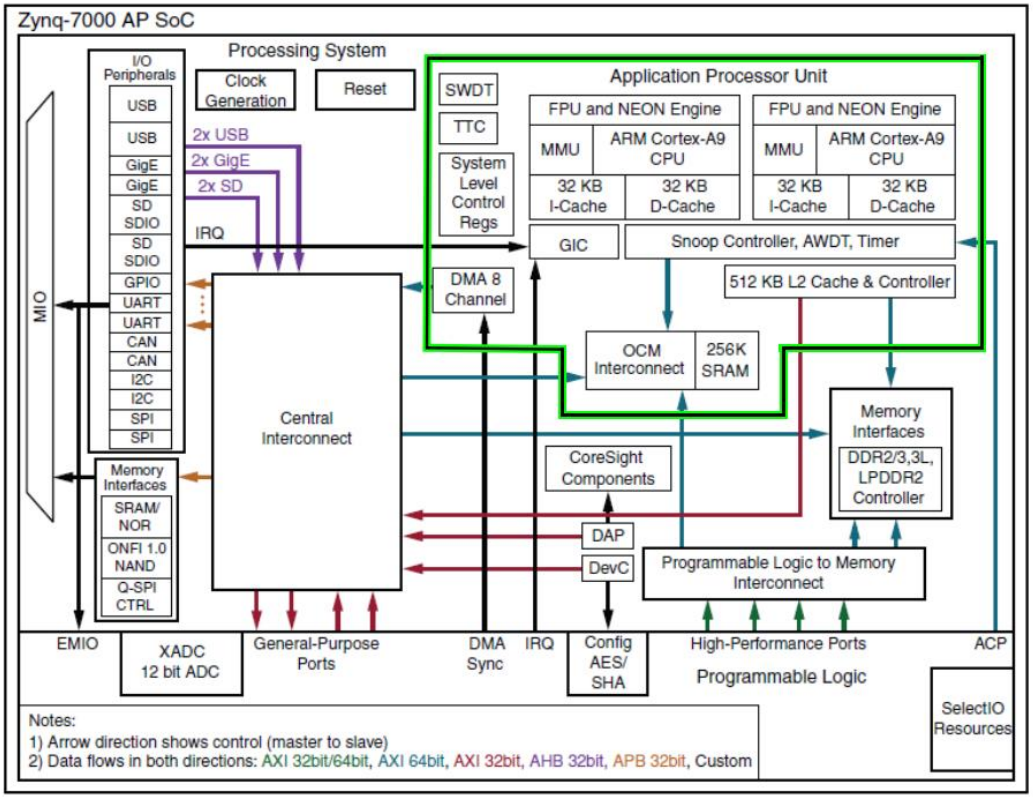


Figure 2.5: Zynq-7000 PS block diagram [16]

The OCM contains 256 KB of RAM and 128 KB of ROM (BootROM). The RAM part of the OCM can be considered a general memory, so it is accessible via system buses. It supports two AXI interface ports, one dedicated for processor access via the SCU and the other shared by all other bus masters within the PS and PL. The BootROM is reserved exclusively for use during the device's boot process and remains inaccessible to user applications. By default, the address space allocated to the OCM is treated as non-cacheable. However, it can be reconfigured as cacheable memory through appropriate settings in the processor's MMU. While enabling caching for the OCM may not yield significant performance gains over a standalone L2 cache, since both are effectively positioned at the same level in the memory hierarchy, it can be a valuable configuration

in systems where the OCM serves as a shared memory resource between the PS and PL, especially when cache coherence is required.

It is important to highlight that in the Zynq-7000 architecture, byte-parity error detection is supported in the L1 cache, L2 cache, and OCM. Parity represents a basic form of error detection code that functions by counting the number of logical ones within a data unit, such as a byte, and appending a parity bit to indicate whether the count is odd or even. This mechanism enables the detection of an odd number of bit errors, although it cannot reliably detect even numbers of erroneous bits. In the event that a parity error is encountered in cached data, the system treats it as a cache miss and requests data retrieval again. While parity checking in the L1 cache is permanently enabled and cannot be disabled, parity support for both the L2 cache and the OCM is configurable, allowing system designers to enable or disable it based on the requirements of the application.

The SCU plays a central role in interfacing the Cortex-A9 processor core(s) with the system's memory hierarchy. It is specifically designed to maintain data cache coherency between the processor cores and the L2 cache, ensuring consistent and synchronized data access across the multiprocessor environment. Beyond coherency, the SCU also handles interconnect arbitration, manages data transfers between caches and system memory, and facilitates communication across the processors. An additional critical function of the SCU is to manage transactions between the PS and PL via the accelerator coherency port (ACP). The ACP enables the PL to act as an AXI master, providing asynchronous cache-coherent access to the PS memory subsystem. Through this interface, the PL can directly access the L2 cache and OCM while maintaining coherence with the L1 caches of the processors. This system coherence not only increases the overall system performance but also reduces the software complexity and improves power consumption.

The primary interface between the PL and PS is via a set of AXI ports, each composed of multiple channels. These make dedicated connections between the PL fabric and interconnection blocks within the PS, such as General Interconnect, Memory Interconnect and SCU, as shown in Figure 2.5. There are four general-purpose AXI ports in total, which are suitable for low and medium-rate communications between the PL and PS. The interface is direct and does not include buffering. The PS is the master of two, and the PL is the master of the other two AXIs. In addition, four high-performance AXI interfaces provide PL bus masters with high bandwidth datapaths to the DDR and OCM memories. Each interface includes two first-in first-out (FIFO) buffers to accommodate burst read and write transactions, and support high-rate communications between the PL and memory elements in the PS.

The DDR memory controller supports access to a 1 GB address space with a single-rank dynamic RAM memory with optional ECC. The system accesses the DDR via its four AXI memory ports. One AXI port is dedicated to the L2 cache for the CPUs and ACP, two ports are dedicated to PL for access through the high-performance AXI interfaces, and the fourth port is shared by all the other masters on the central interconnect.

Communication between the PS and external interfaces is achieved primarily via the multiplexed I/O (MIO) pins. Configurable routing options are provided for these pins, allowing designers to map various peripheral interfaces to specific MIO lines as needed. In scenarios where the available MIO resources are insufficient, or when interaction with custom hardware is required, the EMIO interface is utilized. Unlike MIO, EMIO routes signals through the PL, enabling additional I/O expansion and offering a flexible means for the PS to communicate with user-defined logic implemented in the PL or with external devices via the PL's I/O fabric. This architectural feature enhances design versatility, especially in applications requiring tight integration between software-controlled peripherals and hardware accelerators.

2.3.2 Zynq-7000 PL

The second principal part of the Zynq-7000 architecture is the PL, which is based on the AMD Xilinx Artix-7 or Kintex-7 FPGA families. Thus, it has the same internal architecture as the 7-Series FPGA families. An FPGA can be considered a two-layer device consisting of the application and the configuration layers (Figure 2.6) [129]. The application layer includes all the programmable logic, while the configuration layer consists of the configuration memory that stores all the necessary information to program the device, and the logic and interfaces to access the configuration memory.

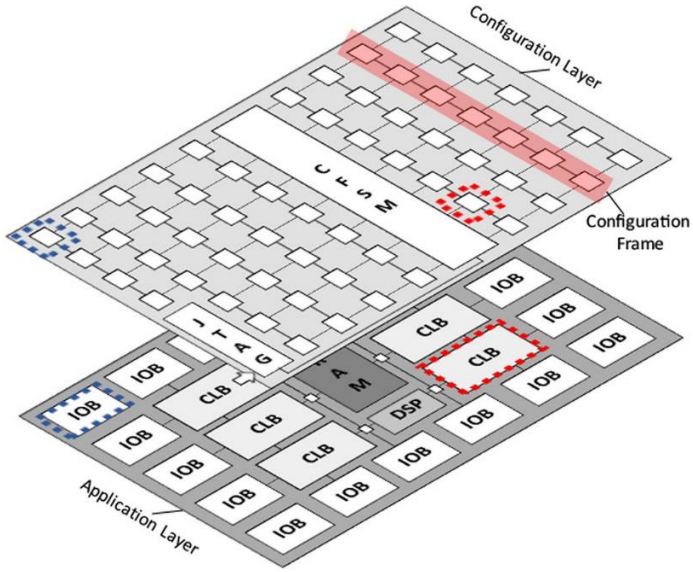


Figure 2.6: Conceptual view of SRAM-based FPGAs [129]

2.3.2.1 *Zynq-7000 PL Application Layer*

The application layer of the Zynq-7000 PL is depicted in Figure 2.7, with various features highlighted. It consists of general-purpose FPGA fabric, which is composed of slices and CLBs, programmable interconnect switch matrices, BRAMs, DSP blocks and IOBs for interfacing.

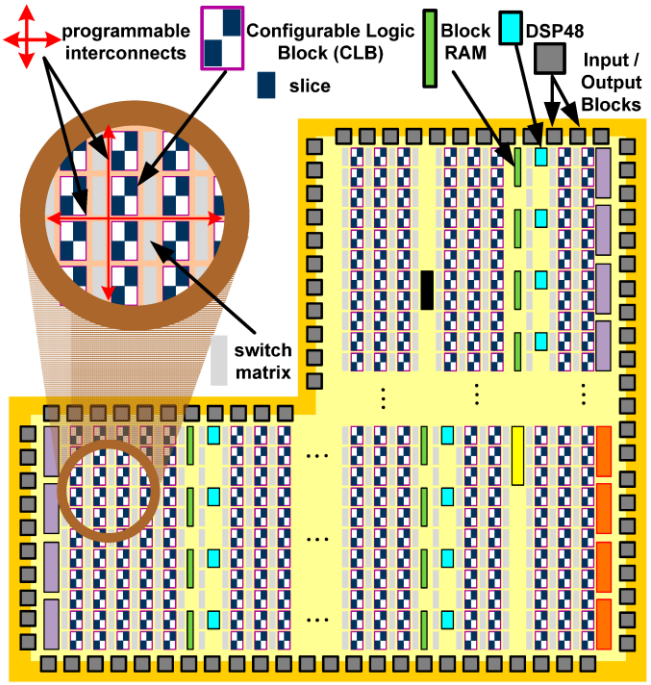


Figure 2.7: Zynq-7000 PL logic fabric [35]

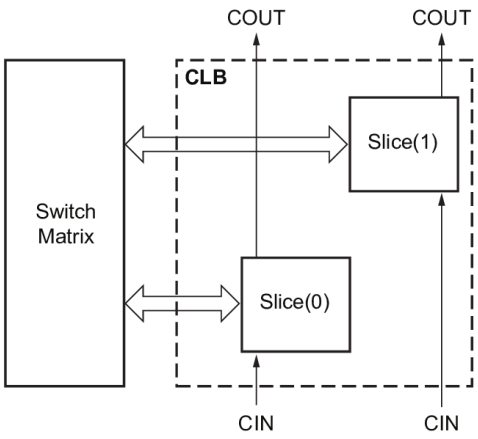


Figure 2.8: Zynq-7000 CLB architecture [6]

The basic logic blocks of the PL are summarised as follows:

- Configurable logic blocks (CLBs) are logic elements that are placed in a two-dimensional array in the PL, and they are connected to other similar resources via programmable interconnects. Each CLB is positioned next to a switch matrix and contains two logic slices (Figure 2.8).
- Slice is a sub-unit within the CLB, which contains resources for implementing combinatorial and sequential logic circuits. CLB includes a pair of slices, and each slice is composed of 4 LUTs, 8 FFs, and other logic, such as wide-function multiplexers and carry logic. These two slices do not have direct connections to each other, and each slice is organized as a column. Some slices support two additional functions: storing data using distributed RAM and shifting data with 32-bit registers. Slices that support these additional functions are called SLICEM, while the others are called SLICEL. Each CLB can contain either two SLICELs or a SLICEL and a SLICEM.
- Lookup table (LUT) is a flexible resource capable of implementing a logic function of up to six inputs, a small read-only memory (ROM), a small distributed RAM or a shift register. The Boolean function generators are implemented as six-input LUTs with two independent outputs. The propagation delay through a LUT is independent of the function implemented. In addition, each LUT can implement a 64 x 1-bit ROM. ROM contents are loaded at each device configuration. The LUTs only in SLICEMs (LUTRAMs) can be implemented as a synchronous RAM resource called a distributed RAM (DRAM) element. Multiple LUTs in a SLICEM can be combined in various ways to store more data. Single-port and dual-port RAMs are available in SLICEMs. A SLICEM LUT can also be configured as a 32-bit shift register (SRL) without using the FFs available in a slice. Used in this way, each LUT can delay serial data from 1 to 32 clock cycles. The 32-bit SRL occupies a 64-bit LUT, since there is a master-slave arrangement where two bits are required to make a register stage.
- Flip-flop (FF) is a sequential circuit element implementing a 1-bit register, with reset functionality. There are eight FF storage elements per slice. Four can be configured as either edge-triggered D-type FFs or level-sensitive latches.
- Carry logic comprises a chain of routes and multiplexers to link slices in a vertical column. Arithmetic circuits require intermediate signals to be propagated between adjacent slices, and this is achieved via carry logic.
- Adjacent to each CLB is a Switch Matrix, which offers a highly flexible routing mechanism. It enables signal connectivity both within the internal components of a CLB and between multiple CLBs, as well as with other programmable resources in the PL fabric. This interconnect structure is essential for establishing custom data paths and supporting complex logic implementations within the FPGA architecture.

- Block RAMs (BRAMs) are embedded within the logic array in a column arrangement and serve as versatile memory resources. They can be configured to function as RAM, ROM, and FIFO buffers, with built-in support for ECC. Each BRAM has a maximum storage capacity of 36Kb, and can be utilized either as a single 36Kb memory block or divided into two independent 18Kb blocks. The default configuration typically employs a word width of 18 bits, yielding 2048 addressable memory locations. However, BRAMs are highly configurable and can be adapted to a range of data widths and depths, such as 1024 x 36 bits or 512 x 72 bits, depending on application requirements. When configured in the 72-bit wide mode, BRAMs can operate as 512 x 64-bit memories with built-in Hamming code for error correction. This configuration uses 8 additional bits to store ECC parity information. During write operations, these protection bits are generated and stored alongside the data. On read, the ECC logic is capable of correcting single-bit errors and detecting (but not correcting) double-bit errors, implementing a single error correction, double error detection (SECDED) mechanism to enhance memory reliability.
- Digital signal processor (DSP) slices are specialized, low-power, high-performance processing elements integrated within the FPGA fabric in vertically aligned columns. These custom-designed slices offer an optimal balance of speed, area efficiency, and design flexibility, making them ideal for implementing computationally intensive arithmetic operations. While originally developed for digital signal processing, the capabilities of DSP slices extend far beyond, benefiting a wide range of applications such as dynamic bus shifting, address generation, and wide-bus multiplexing. The basic functionality of the DSP slice is shown in Figure 2.9. DSP slice includes a 25-bit pre-adder/pre-subtracter prior to feeding a multiplier, a two's complement multiplier that accepts a 25-bit and an 18-bit two's complement inputs, a 48-bit adder/subtracter or logic unit and a pattern detector connected to the output of the add/subtract/logic unit. The architecture also supports cascading multiple DSP slices to form wide math functions, DSP filters, and complex arithmetic without the use of general FPGA logic.
- Input-output blocks (IOBs) serve as the interface between the internal logic resources of the PL and the external pins of the device that connect to surrounding hardware. Each IOB is designed to manage a single-bit signal, either as an input or an output, enabling communication between the FPGA fabric and external circuitry.
- The clocking resources manage complex and simple clocking requirements with dedicated global and regional I/O clock buffers and lines. Global clock trees allow clocking of synchronous elements across the device. I/O and regional clock trees allow clocking of up to three vertically adjacent clock regions. The clock management tiles (CMTs), each containing one mixed-mode clock manager

(MMCM) and one phase-locked loop (PLL), reside in the CMT column next to the I/O column. The MMCM and PLL can serve as frequency synthesizers for a wide range of frequencies, and filter jitters for either external or internal clocks or deskew clocks. The PLL is a subset of the MMCM functionality.

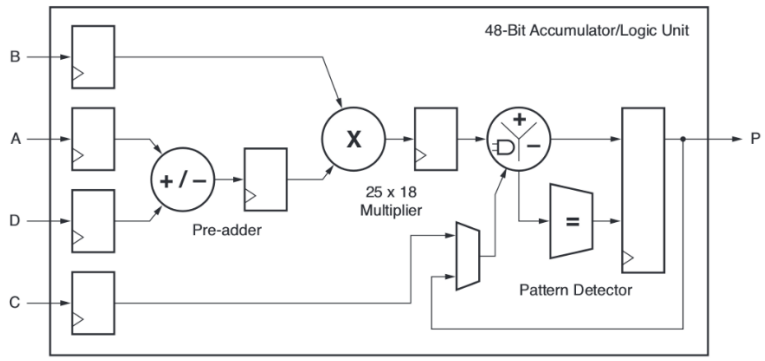


Figure 2.9: Zynq-7000 DSP slice functionality [7]

2.3.2.2 Zynq-7000 PL Configuration Layer

The Configuration Layer comprises all SRAM memory cells responsible to configure all the application layer's blocks. The programmable blocks are defined and controlled through a configuration bitstream, which consists of a structured set of configuration bits. These bits are loaded into the device's configuration memory, determining the specific functionality and interconnections required to implement a given circuit design. The configuration bitstream encapsulates all the essential data required to program the internal components of the FPGA, including CLBs, BRAMs, DSPs, IOBs, and the routing architecture that interconnects them. It defines the behaviour of user logic by specifying, for instance, which LUTs and FFs within the CLBs are activated, the logic functions implemented by the LUTs, and the specific programmable interconnect points within the switch matrices that must be enabled to establish the desired signal paths. Each configuration detail is governed by one or more individual bits within the bitstream, collectively determining the overall circuit structure and functionality. The bitstream can be generated by AMD software tools (e.g., Vivado) as a unique binary file.

The configuration memory is organized into frames, which represent the smallest individually addressable units within the configuration space. As a result, all configuration operations must be performed at the frame level, rather than on smaller portions of memory. The total size of the configuration bitstream is directly related to the number of frames in the device, which in turn depends on the amount of available logic resources; devices with greater resource density contain more frames and thus require larger bitstreams. Each frame typically consists of 101 words, with each word being 32 bits, totalling 3232 bits per frame [5]. Furthermore, every frame is assigned a unique

address corresponding to its physical location in the FPGA floorplanning. This frame address is structured into five distinct fields, as shown in Table 2.2.

Address field	Bit index	Description
Block type	[25:23]	0: CLB, BRAM, DSP, I/O, CLK configuration 1: BRAM content 2: CFG_CLB 3: Unknown type
Top/Bottom bit	[22]	0: Top-half rows 1: Bottom-half rows
Row address	[21:17]	Defines the row. The row addresses start at 0 on the center and increment to the top or bottom.
Column address	[16:7]	Defines the column, such as a column of CLBs, DSP, etc. The column addresses start at 0 on the left and increase to the right.
Minor address	[6:0]	Defines a frame within a column.

Table 2.2: Configuration frame address structure

The block type indicates the functionality that corresponds to a particular configuration frame. Block type 0 frames represent the configuration of the programmable logic components, while block type 1 frames reflect the BRAM contents. Both of these types account for the largest percentage of configuration memory blocks, as shown in Table 2.3, in the case of 7z020 device. AMD does not provide documentation for the block types 2 and 3. However, in [62], it is mentioned that the CFG_CLB block type 2 defines which part of the FPGA needs to be reset or reconfigured in case of partial reconfiguration and it only appears in a partial bitstream.

Block type	Count	Percentage
0	7692	75.15%
1	2304	22.51%
2	222	2.17%
3	18	0.17%

Table 2.3: Zynq 7z020 configuration frame types

The configuration bitstream includes the configuration frames with block types 0 and 1. The internal Readback CRC feature, mentioned below, scans the configuration frames with block types 0, 2 and 3. These configuration frames contain mostly static data, while

type 1 frames (i.e., BRAM content) are only dynamic, which means that their contents change during normal circuit operation. Type 0 frames, except configuration static bits, contain bits which correspond to FFs, DRAMs, SRLs.

In addition to generating the bitstream file, the AMD's development tools can also generate optional files that provide further information about configuration bits by classifying them as masked or essential. Masked bits are typically associated with user-defined memory elements, such as FFs, BRAM, DRAM, or SRLs. In contrast, essential bits represent the subset of configuration bits that directly influence the implementation of the user logic and are crucial for the proper functioning of the design.

The configuration memory is structurally divided into two principal regions: top and bottom. Each region is organized in rows and columns. Each frame spans the height of a row, while the columns are arranged according to the type of configurable resource they represent (e.g. CLB, BRAM, DSP, IOB, etc.). Each column contains a specific number of frames depending on the resource type it configures; CLB column contains 36 frames, BRAM and DSP columns 28 frames, IOB column contains 42 frames and CLK column 30 frames.

The actual memory contents of the BRAM are configured separately within a distinct segment of the configuration address space. This is because a) the access mechanism for programming BRAM contents differs from that used for standard configuration frames and b) this separation allows the BRAM contents to be optionally omitted if not required, offering a substantial reduction in the overall bitstream size. The BRAM content column contains 128 frames.

Each frame has an embedded 32-bit ECC word that is used to achieve SECDED functionality. A 13-bit syndrome value using all 100 data words of that frame corresponds to each frame. The ECC word resides in the 51^{st} word in block types 0, 2, and 3 frames, forming the 100 data + 1 ECC = 101 words per frame. The 13-bit syndrome represents $2^{13} = 8192$ possible unique error bit positions in a frame. Since the code is a SECDED code, two-bit errors can also be detected but not corrected. All odd-numbered multi-bit errors will be detected by the ECC, but the syndrome value will not actually indicate any of the erroneous bits. The vast majority of even-numbered errors larger than two bits will also be detected, but detection is not guaranteed.

To reduce the effects of multi-bit errors, which are usually localized to a very small area of the chip, Zynq-7000 configuration memory is interleaved such that logically adjacent bits are physically separated [12], [67]. Physically adjacent bits belong to different logical frames rendering multi-bit upsets correctable, as they appear as separate single-bit errors allowing them to be repaired by the built-in ECC. Thus, ECC can correct all single-bit errors and all double-bit (or multiple-bit) errors because upsets span across frames. In addition, a 32-bit CRC checker, also built into the device, detects 100% of multi-bit errors. In this case, reconfiguration of the Zynq-7000 PL must be performed.

The configuration memory of the device can be accessed via dedicated interfaces, allowing both external and internal configuration operations. Externally, the device is most commonly programmed through the widely used serial JTAG interface, which provides a straightforward and standardized configuration mechanism. Internally, configuration can be managed via software running on the PS, offering greater flexibility. The processor configuration access port (PCAP) is the primary internal interface employed for this purpose, as it enables configuration and reconfiguration without requiring the PL to be preloaded with a bitstream. A key advantage of PCAP is its ability to facilitate runtime access to the configuration module by applications executing on the embedded processors. In addition to PCAP, the internal configuration access port (ICAP) provides another internal method for modifying the device configuration. ICAP is typically utilized by logic within the PL itself to support partial or full reconfiguration after the initial configuration has been completed. However, it is important to note that ICAP cannot be used for the device's initial configuration, distinguishing it from the PCAP and external JTAG interfaces.

Zynq-7000 devices allow users to read configuration memory through the PCAP, ICAP, and JTAG interfaces. There are two styles of readback: readback and readback capture. During readback, the user can read all configuration memory cells, including the current values on all user memory elements (i.e., LUTRAM, SRL and BRAM) and the initial state programmed through bitstream of the internal CLB and IOB FFs. Readback capture is a superset of readback, also reading the current state of all FFs storing all register values in the same configuration memory cell on which the corresponding register initial values have been programmed. There is also the built-in Global LUT mask (GLUTMASK) feature which can be controlled by setting a specific configuration register bit [5]. When enabled, any dynamic bits of LUTRAM or SRLs are masked on all readback data, otherwise, the current state of these memories is stored into the configuration memory cells.

In addition, the Zynq-7000 architecture incorporates several mechanisms designed to enhance the robustness of the device against configuration memory upsets. Among these are the per-frame ECC, as described above, and a global CRC mechanism. Each configuration frame is equipped with its own ECC word, which facilitates the correction of single-bit errors and the detection of double-bit upsets. Complementing this, the CRC mechanism operates at the device level and is capable of detecting most multi-bit errors, thereby offering a comprehensive layer of protection against configuration data corruption. The feature built into recent AMD Xilinx FPGAs is the Readback CRC. It can continuously scan in the background all the configuration frame data of a user design and run the ECC check for each frame, and the CRC comparison for the entire device with the golden values. This feature aims to simplify the detection of SEUs that cause a configuration memory bit to flip and can be used in conjunction with the ECC feature for advanced operations such as SEU corrections.

Output Signal	Description
FAR[25:0]	It holds either the address of the frame where the ECC error was detected or the current address of the Readback CRC feature during the readback process. The functionality is selected by a control register [5].
SYNDROME[12:0]	It reflects the ECC calculation performed on all of the bits in the frame, and subsequent comparison to the golden ECC word stored in that frame. A non-zero value indicates that an error has been detected. No error: [12] = 0, [11:0] = 0 Single-bit error: [12] = 1, [11:0] ≠ 0 Double-bit errors: [12] = 0, [11:0] ≠ 0 Parity Bit Error: [12] = 1, [11:0] = 0
SYNWORD[6:0]	It holds the word location in the frame where an ECC error has been detected, if the upset is a single-bit error.
SYNBIT[4:0]	It holds the bit location in the word where an ECC error has been detected, if the upset is a single-bit error.
ECCERROR	It indicates that an ECC frame error has been detected. 1: when SYNDROME ≠ 0 0: when SYNDROME = 0
ECCERRORSINGLE	It indicates a single-bit frame ECC error detected.
SYNDROMEVALID	It indicates that all the above outputs are valid and that the ECC computation and the associated comparison process for the corresponding frame have been completed. This signal is asserted for precisely one clock cycle per frame.
CRCERROR	It indicates that the Readback CRC has detected a CRC error. During the Readback CRC operation, the system not only computes syndromes for each configuration frame, but also concurrently calculates a cumulative CRC value across all scanned frames. This resulting CRC is then compared against the golden CRC value to verify the correctness of the entire device configuration.

Table 2.4: FRAME_ECCE2 output signals

There is a dedicated hardware primitive supported by 7-Series architectures including Zynq-7000, called FRAME_ECCE2 [8]. This design element enables the dedicated, built-in ECC for the configuration memory. It contains outputs that allow monitoring of the status of the ECC circuitry and the status of the Readback CRC circuitry. Every readback, regardless of the configuration interface (i.e., JTAG, PCAP, ICAP and Readback CRC), passes through FRAME_ECCE2. Its outputs then indicate whether the

ECC calculation circuit found an error or not. These output signals (Table 2.4) export valuable information about the locations, as well as the types of upsets that have occurred.

3 Radiation Environment and Integrated Circuits

3.1 Radiation Environments

3.1.1 Space Radiation Environment

The space environment within our solar system is primarily shaped by three major sources of radiation [46], [85]:

- Galactic cosmic rays (GCR): These are high-energy particles, mainly protons, originating from outside the solar system. GCRs form an almost uniform flux, arriving isotropically from all directions and representing one of the most energetic components of space radiation.
- Solar radiation: Continuously emitted by the Sun, this radiation includes low-energy photons, plasma, and magnetic fields, collectively forming what is known as the solar wind, a constant stream of charged particles. Superimposed on this steady flow are transient, more intense emissions caused by solar events such as solar storms, flares and coronal mass ejections (CMEs), which can significantly increase particle energy and flux levels in localized regions.
- Radiation belts: These are zones of trapped energetic particles, formed when charged particles are captured by a planet's magnetic field. The Earth's Van Allen belts are a prominent example, where particles spiral along magnetic field lines and become concentrated in specific regions.

GCRs are high-energy charged particles (i.e., 40 to 1000 MeV) that enter the solar system from the outside, consisting primarily of protons (ionized hydrogen) and secondarily of alpha particles (ionized helium), heavy ions and electrons. The galactic magnetic field deflects the charged GCRs, thus accelerating them around circular paths, confining them to the disk of the galaxy, as shown in Figure 3.1. This figure also presents the heliopause, which represents the boundary where the sun's influence ends and the termination shock, which is the boundary of the solar system, i.e., where the solar wind particles slow down so that they travel slower than the speed of sound. Most GCRs have been traveling in our galaxy for tens of millions of years. Although their flux is low, they include energetic

heavy ions that can deposit significant amounts of energy in sensitive volumes and so cause problems to spacecraft electronics and humans in space. The Earth's magnetic field provides partial protection against cosmic rays, a phenomenon referred to as geomagnetic shielding. Nevertheless, GCRs can still penetrate this shield, particularly in polar regions and at higher altitudes where the magnetic field is weaker.

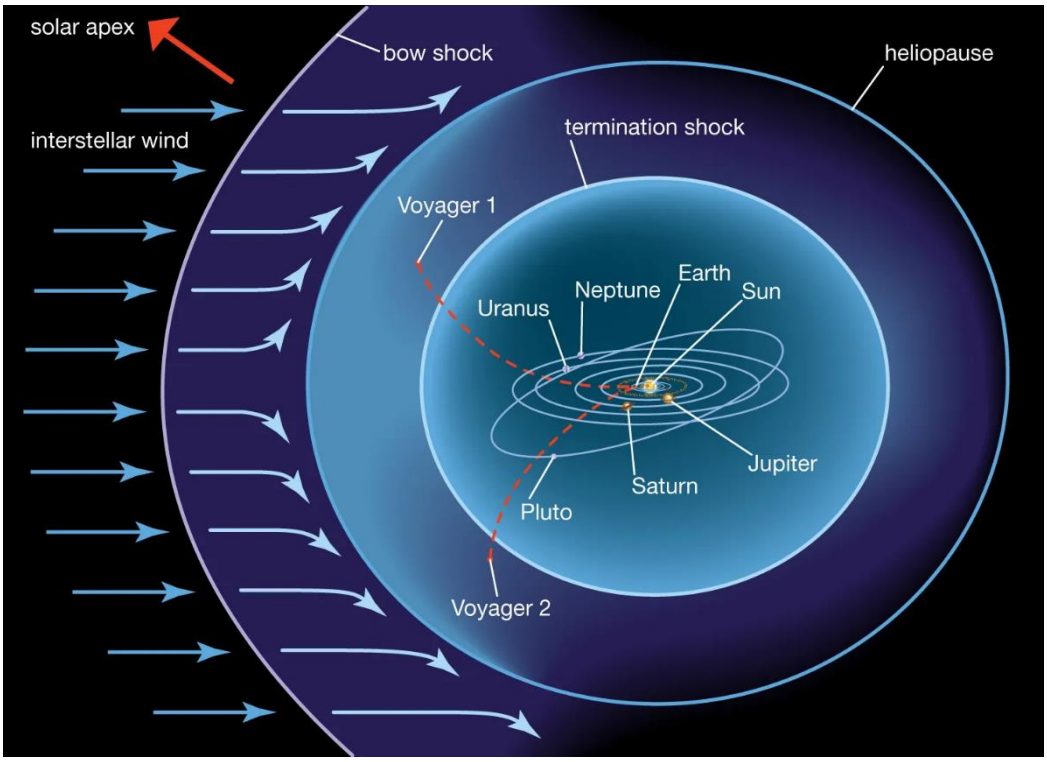


Figure 3.1: Galactic cosmic rays and our solar system [60]

Solar radiation encompasses streams of high-energy particles generated by solar phenomena such as solar wind and solar flares, typically ranging in energy from 10 MeV to 10 GeV. These emissions are predominantly composed of protons, with smaller contributions from electrons and heavy ions. The solar wind originates in the sun’s corona, the outermost layer of the sun, and consists of charged particles accelerated to high velocities that escape the sun’s gravitational pull. This continuous flow interacts with the Earth's magnetic field, inducing geomagnetic disturbances that can lead to phenomena such as magnetic storms, which in turn may disrupt radio communications. In general, the solar wind poses a relatively low risk to spacecraft electronics and human crews due to its lower-energy particle composition, much of which is deflected or trapped by planetary magnetospheres. In contrast, solar flares represent the most intense form of solar activity, characterized by a sudden and localized release of radiation from active regions on the sun’s surface. Although typically short-lived and occurring intermittently

every few days, solar flares can discharge vast amounts of energy, potentially impacting the Earth's upper atmosphere and posing significant hazards to spaceborne electronic systems.

As the solar wind and solar flare flow past Earth, they mostly deflect around Earth's magnetosphere, which is the protective magnetic barrier surrounding our planet. Bursts of solar energy that travel toward Earth smash into the magnetosphere, sending particle radiation along our planet's magnetic field lines.

Radiation belts may develop around any planetary body possessing a sufficiently strong magnetic field, known as a magnetosphere, capable of deflecting and confining charged particles before they penetrate the planet's atmosphere. These belts are composed primarily of particles originating from the solar wind, along with a fraction of lower-energy GCRs. In the case of Earth, its geomagnetic field captures and retains energetic protons and electrons, forming toroidal (doughnut-shaped) regions of high particle density around the planet (Figure 3.2). Earth has two such belts named Van Allen, who is credited with their discovery; an inner belt and an outer belt [131].

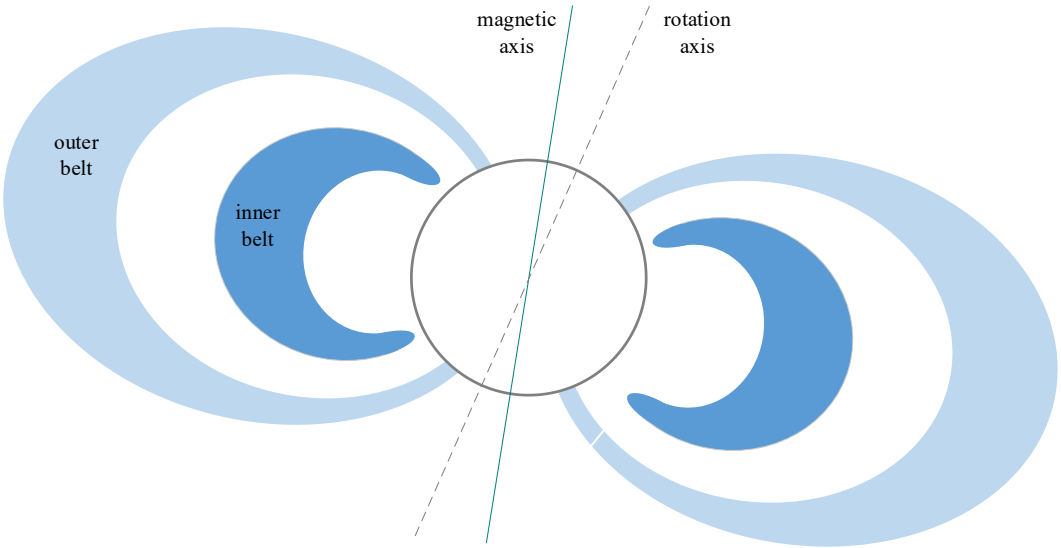


Figure 3.2: Van Allen radiation belts

The radiation belts are most pronounced at the equator, where Earth's magnetic field lines are more horizontal and the field strength is sufficient to effectively trap charged particles. As latitude increases, the belts gradually thin out and eventually vanish near the poles, where the magnetic field becomes perpendicular to the Earth's surface. The inner radiation belt typically extends from approximately 1,200 km to 6,000 km above the equator. However, under conditions of heightened solar activity or in specific regions

like the South Atlantic Anomaly, the inner boundary can descend to altitudes as low as 200 km. This results in elevated fluxes of energetic particles, posing significant radiation hazards to satellites operating in low Earth orbit. The inner belt is characterized by high densities of energetic electrons, with energies ranging from 0.5 to 5 MeV, and protons with energies exceeding 10 MeV, all confined by the intense magnetic fields in this region.

The outer radiation belt exhibits greater variability compared to the inner belt, primarily due to its heightened sensitivity to fluctuations in solar activity. It is almost toroidal in shape and spans from approximately 13,000 km to 60,000 km above Earth's surface. This belt comprises a mix of charged particles originating both from the solar wind and the Earth's upper atmosphere. A significant portion of the solar contribution consists of helium ions carried by the continuous outflow of particles from the Sun. Unlike the inner belt, the protons in the outer belt possess comparatively lower energies but are present in much higher fluxes. The most energetic constituents of the outer belt are electrons, with energies reaching several hundred MeV. These high-energy electrons are particularly abundant throughout the belt, although their flux diminishes sharply near the outer boundary, approaching typical interplanetary levels.

Protons and electrons are the most prevalent particles within Earth's trapped radiation belts, making them the dominant components of the space radiation environment. The energy distribution of these particles plays a critical role in determining how effectively spacecraft shielding mitigates their impact. As particles penetrate the spacecraft's outer structure and electronic housings, their energy spectra are altered, influencing both the level and type of radiation-induced effects. In low-earth orbits, the majority of particles that penetrate standard shielding are high-energy protons, thereby reducing the relative contribution of electrons to radiation effects in this region. In contrast, for spacecraft operating at higher altitudes, electrons become the primary source of radiation, presenting a significant hazard due to their high flux and potential to induce charging and other damage mechanisms.

3.1.2 Type of Orbits

The influence of Earth's magnetic field on space radiation shielding varies significantly with orbital altitude and geometry. Figure 3.3 illustrates the classification of different orbits based on altitude. Satellites in geostationary orbit (GEO) operate above the equator, moving from west to east in synchrony with Earth's rotation. This orbital configuration allows GEO satellites to remain fixed over a specific point on the Earth's surface. To achieve this geostationary position, satellites must orbit at an altitude of approximately 35,800 km and maintain a velocity of roughly 3 km/s. This distance places them well beyond the orbits of most other satellites. GEO is particularly advantageous for applications requiring continuous coverage of a specific region, such as telecommunications, broadcasting, and meteorological observation.

Low Earth orbit (LEO), as the name implies, refers to orbital altitudes relatively close to Earth's surface, typically below 2,000 km, with a lower limit of around 160 km. Satellites in LEO complete one orbit in approximately 90 to 120 minutes, making this region energetically favourable for launch due to the lower velocities required compared to higher orbits. Unlike geostationary satellites, which are restricted to equatorial orbits, LEO satellites can operate in various orbital inclinations, offering greater flexibility in coverage and mission design. This versatility has made LEO the most frequently utilized orbital regime. It is particularly well-suited for Earth observation missions, as its proximity enables high-resolution imaging. The international space station (ISS) is also positioned in LEO, benefiting from reduced travel time for crewed missions and minimal communication latency. The ISS completes about 16 orbits per day, allowing for the frequent repetition of experiments. In addition, many communication systems in LEO operate through satellite constellations, consisting of numerous coordinated satellites that work together to provide continuous global coverage.

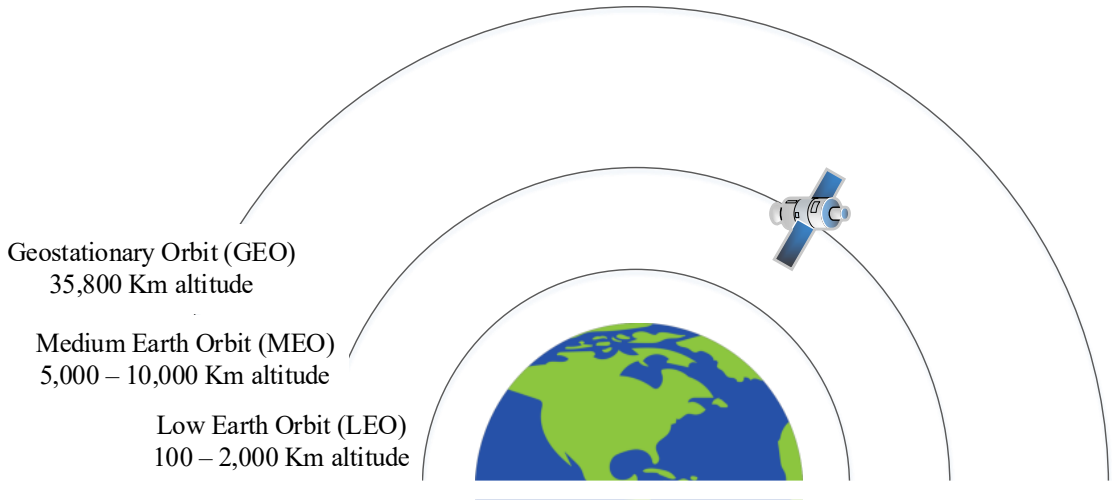


Figure 3.3: Types of Earth orbit by altitude

Medium Earth orbit (MEO) comprises a wide range of orbits anywhere between LEO and GEO (typically 5,000 to 10,000 km). It also does not need to take specific paths around Earth, as LEO. It is commonly used by navigation satellites, like the European Union (EU) Galileo system and the global positioning system (GPS) owned by the United States (US), which use a constellation of multiple satellites to provide coverage across large parts of the world all at once.

The inclination is another element describing the shape and orientation of a celestial orbit (Figure 3.4). It is the angle between the orbital plane and the plane of reference, normally stated in degrees. For a satellite orbiting Earth, the plane of reference is usually the plane containing the Earth's equator. Thus, when a satellite lies close to the equatorial plane

(equatorial orbit), its inclination is equal to 0 degrees. On the other hand, polar orbit passes above or nearly above both Earth's poles on each revolution, i.e., it has an inclination of ± 90 degrees, and thus, satellite track covers all parts of Earth as the planet rotates. If the orbit's inclination in reference to the equatorial plane is neither 0 nor 90 degrees, the orbit is called inclined and covers a range of altitudes in the Northern and Southern Hemispheres.

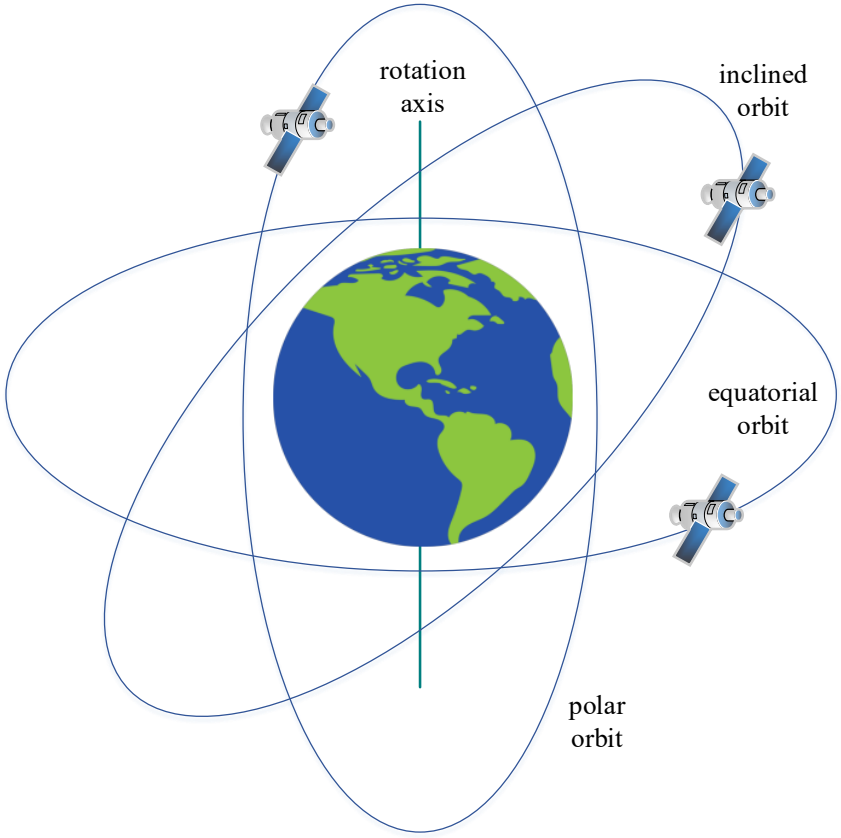


Figure 3.4: Types of Earth orbit by inclination

LEO, especially in equatorial regions, offers the most effective natural protection against space radiation due to the maximal shielding provided by Earth's magnetic field. This shielding significantly reduces particle fluxes, thereby minimizing radiation-induced effects on spacecraft systems. In contrast, orbits at higher altitudes, such as MEO and GEO, as well as orbits with high inclinations or polar trajectories, experience a substantial decrease in magnetic shielding efficiency. As a result, these orbits are subject to increased fluxes of charged particles and a greater likelihood of radiation-related disturbances. Missions operating in polar or highly inclined orbits are particularly vulnerable, as magnetic protection diminishes at latitudes farther from the equator. In the case of interplanetary missions, spacecraft travel beyond the extent of Earth's

magnetosphere, thereby becoming fully exposed to the elevated fluxes of high-energy particles in deep space.

3.1.3 Other Radiation Environment

GCRs undergo significant attenuation as they travel through the interplanetary medium, resulting in only a small fraction reaching Earth. Upon interacting with the upper atmosphere, GCRs initiate cascades of secondary particles, among which neutrons with very significant fluxes. These atmospheric neutrons can penetrate to the Earth's surface and induce soft errors in high-speed electronic systems operating at ground level. These neutron interactions within semiconductor materials can cause upsets analogous to those caused by energetic protons in space, albeit at a substantially lower frequency. In addition to atmospheric neutrons, terrestrial radiation exposure also stems from alpha particles emitted during the decay of trace radioactive contaminants. Such impurities may exist in minute concentrations within integrated circuit materials, including metallization layers, bonding compounds, packaging materials, and even the silicon substrate itself. The terrestrial radiation environment encompasses the region within the Earth's atmosphere, extending from sea level to typical commercial aviation altitudes, reaching up to approximately 22 Km.

Radiation environments are not limited to those found naturally in space; artificial, human-made radiation environments also pose significant challenges, particularly for microelectronic systems. These systems are frequently exposed to radiation in various medical, industrial, and defense contexts, where they must maintain functionality under potentially harsh conditions. In the medical field, radiation exposure primarily arises from diagnostic and therapeutic equipment, such as X-ray imaging systems and proton-beam therapy units. Additionally, electron-beam and gamma-ray irradiation are commonly employed to sterilize surgical instruments and implantable electronic devices within clinical settings. In industrial applications, radiation is extensively used for both material processing, where it induces specific chemical or physical modifications, and for non-destructive testing, such as defect inspection using X-ray, gamma, or electron beams. Within nuclear power facilities, microelectronic components are subjected to significant levels of neutron and gamma radiation, necessitating robust shielding and radiation tolerance. In the defence environment, electronic systems must be designed to withstand brief yet intense exposures to gamma rays and neutrons, as well as the secondary electromagnetic pulses generated by nuclear detonations.

3.2 Radiation Effects in Integrated Circuits

Radiation effects in electronic components can be separated into cumulative effects, which lead to progressive degradation of the component characteristics, and single event

effects, which gather different types of events, destructive or non-destructive, induced by a single particle.

Protons and electrons produce electron-hole pairs as they lose energy through the ionization process. The ionization effect can not only produce long-term ionization damage effects but also induce transient single-event effects. Apart from ionizing dose, particles can lose energy through non-ionizing interactions with materials, particularly through displacement damage, where atoms are displaced from their original sites. This can alter the electrical, mechanical or optical properties of materials [45].

3.2.1 Cumulative Effects

Exposure to both primary and secondary radiation leads to persistent, long-term changes in the characteristics of electronic devices and circuits. Such exposure can result in parametric degradation, and under more severe conditions, may cause complete functional failure. Cumulative effects include both total ionizing dose (TID) and displacement damage (DD), also referred to as total non-ionizing dose (TNID) [47].

TID effects are induced when ionizing radiation, such as charged particles (i.e., electrons, protons, etc.), interacts with a component material, particularly the insulating and semiconductor layers in electronic devices creating electron-hole pairs in the component material. These charges tend to become trapped within dielectric layers, either within the bulk material or near the interface between the dielectric and the semiconductor, where their electrostatic influence on device operation is most pronounced. These trapped charges can lead to a range of adverse effects on device performance, including shifts in threshold voltage, increased leakage currents, and timing deviations. The radiation dose represents the amount of energy deposited per unit mass of a material, primarily through processes of ionization and excitation. TID is commonly quantified in units of rad (radiation absorbed dose). The actual TID accumulated by a device depends on several factors, including the spacecraft's mission profile, orbital environment, and the level of shielding implemented. In practice, the received dose may vary from a few krad to several hundreds of krad, depending on these conditions.

As mentioned above, the key point of the different orbits is that if the vehicle travels within a radiation belt, it will be exposed to a lot of TID effects. If it travels between the belts, i.e., inside the donut hole, it is actually protected. These belts give vehicles an electromagnetic shield from external radiation sources. But also, the inclination and shape of the vehicle's orbit significantly impact the amount of radiation a satellite is exposed to and how they are positioned relative to the belts. Depending on its mission, a satellite may have a different orbit. The radiation dose it receives and, thus, the TID effect is very dependent on orbit.

In Figure 3.5 [116], the TID dependency on the altitude and the inclination is shown. Orbits with altitudes up to 1,000 km and close to GEO have low TID, i.e., less than 1

krad per year. In LEOs near the inner belt, the TID can reach 20 krad/year for equatorial orbit (0 degrees), while it is up to 10 krad/year for polar orbit (90 degrees). For example, LEO constellations such as Jason, which provides global sea surface height observations for climate change monitoring and ocean and seasonal forecasts, and Globalstar, which consists of 25 satellites that support satellite phone and low-speed data communications, are located in orbits with inclinations of about 60 degrees and with TIDs in the range of 1 to 10 krad/year. The MEO satellite navigation systems (EU Galileo, US GPS and Russian GLONASS) have a similar orbital inclination. Although they travel within the outer belt, they have a low TID due to their inclined orbits.

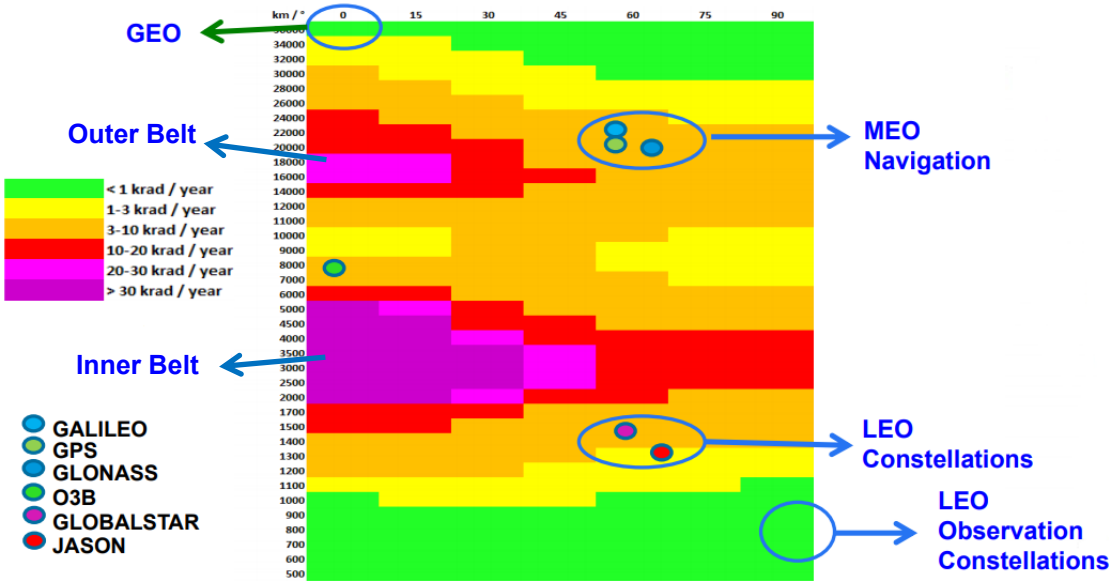


Figure 3.5: Mission TID levels dependent on altitude & inclination [116]

Note that the typical radiation tolerance of a typical COTS device is up to 20 krad, while the tolerance of rad-tolerant and rad-hard devices reaches 200 krad and 1 Mrad, respectively [27]. This means that for a mission lasting, for example, 10 years, a COTS device cannot be affected in orbits indicated with green color in Figure 3.5, while rad-tolerant devices can endure even in the red orbital regions and rad-hard devices can withstand in the purple regions.

DD effects arise from non-ionizing energy transfer processes, where energetic primary and secondary particles interact with the atoms of a material, displacing them from their lattice positions. This interaction results in permanent structural defects within the semiconductor crystal lattice, often leading to the formation of electrically active trap states. These defects can significantly alter the performance of affected devices, particularly in bipolar technologies and optoelectronic components such as light-emitting

diodes and laser diodes. In contrast, digital integrated circuits like ASICs and FPGAs are generally less susceptible to displacement damage effects.

3.2.2 Single Event Effects

Single event effects (SEEs) encompass a range of phenomena in which individual energetic particles can cause transient disruptions or permanent damage to microelectronic devices. Unlike cumulative effects such as TID, which result from prolonged exposure to large numbers of particles, SEEs are triggered by a single particle interaction with sensitive regions of a device. Protons and heavier ions, and neutrons can induce such effects. Heavy ions and low-energy protons primarily induce these effects through direct ionization, where energy is deposited directly into sensitive regions of the semiconductor material. In contrast, higher-energy protons and neutrons typically cause disruptions via indirect mechanisms, particularly through nuclear interactions that occur within or near the active semiconductor. These interactions can produce secondary particles and localized charge deposition, which in turn may lead to transient or permanent faults in the circuitry. Figure 3.6 shows how a charged particle strikes the integrated circuit surface and creates a temporary ionization trail [30].

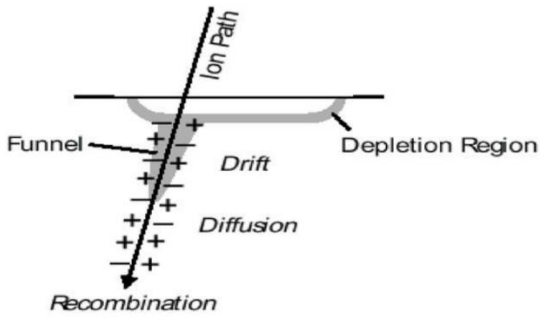


Figure 3.6: Charged particle strike [30]

The ionizing effect of heavy ions is quantified using the linear energy transfer (LET), expressed in units of MeV*cm^2/mg. LET describes the amount of ionizing energy deposited per unit path length of the ion, normalized by the density of the traversed material. In contrast, protons are typically characterized by their kinetic energy, measured in MeV. The susceptibility of a device to SEE is commonly described through a cross section, which is plotted as a function of ion LET or proton energy. The cross section is defined as the ratio of the number of observed single events to the incident particle fluence, i.e., number of particles per cm. Essentially, the cross section represents the probability that an impinging particle provokes a single event.

SEEs can be divided into non-destructive and destructive effects [45].

- Non-destructive effects, also called soft errors, are temporary disruptions in a device's operation that do not result in permanent physical damage. These errors typically manifest as corrupted data or an unintended change in the device's operational state. Recovery is usually possible through techniques such as rewriting data, resetting the system, or performing a power cycle to restore normal functionality. Examples of such effects include single event upset (SEU), multiple-bit upset (MBU), multiple-cell upset (MCU), single event functional interrupt (SEFI), and single event transient (SET).
- Destructive effects occur when high-current conditions are induced, which have the potential to destroy the device, such as single event latch-up (SEL).

An SEU refers to a bit flip, which involves the alteration of the state of a storage element, such as a flip-flop, latch, or SRAM cell. SEUs may either go unnoticed, if the affected element is not utilized or is corrected by an error detection and correction mechanism, or they can lead to various types of circuit-level disruptions. Susceptibility to SEUs varies significantly depending on the device technology, design specifics, and operational conditions. In particular, smaller transistor dimensions and lower supply voltages tend to reduce the critical charge required to induce a state change, thereby increasing the vulnerability to SEEs. In high-density semiconductor devices, it is more likely that a single particle strike will cause a state change in two or more logic cells, resulting in a MCU. The corrupted cells are usually, but not always, physically adjacent. An MBU is a special case of MCU, where the affected cells reside within the same word.

An SET is a temporary voltage spike at a node in an integrated circuit. It results in a large variety of effects at the circuit level, such as spurious voltage transients, false signal or miss counts, temporary disability of the circuit, or at worst SEFI. However, an SET in a combinational circuit might not be captured by the next storage element, e.g., flip-flop, because it can be masked when the pulse resulting from a particle strike reaches the flip-flop, but not at the clock transition where it captures its input value.

An SEFI is a soft error that causes the component to reset, lock up, or otherwise malfunction. Two main types of SEFI are distinguished depending on the actions required to restore operability, reset by software or by power cycling [47].

An SEL is a potentially catastrophic mechanism in which a low impedance path develops suddenly between power and ground and remains after the triggering event dissipates. When it occurs, a high current flows, and if the power supply is maintained, the device can be destroyed by thermal effect.

3.3 Mitigation Techniques

In certain domains, such as industrial and medical applications, radiation effects on microelectronics are often mitigated primarily through the use of shielding. However, in

many space-based and terrestrial applications, shielding against naturally occurring high-energy particle fluxes is not always feasible, as the mass and volume required to provide adequate protection can be prohibitive. As a result, an alternative approach involves reducing the inherent susceptibility of microelectronic components to radiation, thereby ensuring reliable functionality in harsh environments.

This process, known as radiation hardening, encompasses a broad spectrum of design and fabrication techniques aimed at enhancing a device's resilience to radiation-induced disturbances. Importantly, radiation hardening does not imply complete immunity to radiation effects; rather, it refers to improving a component's robustness such that its performance remains within acceptable limits throughout its operational life. Thus, radiation hardening is centered on achieving sufficient tolerance to meet mission reliability requirements, rather than eliminating all radiation-induced effects.

Microelectronics can be hardened against radiation effects through two primary approaches, which may be applied individually or in combination: radiation hardening by process (RHBP) and radiation hardening by design (RHBD) [47]. RHBP focuses on altering the semiconductor manufacturing process to minimize the impact of physical phenomena that influence radiation susceptibility. In contrast, RHBD involves the application of design strategies at various levels, including the physical layout, circuit architecture, and system design.

RHBP can be accomplished through several techniques, such as modifying the doping profiles of devices and substrates, optimizing deposition processes for insulating materials, and selecting specialized materials that enhance radiation resistance. These solutions often have the advantage of being able to make a product radiation-hardened or at least tolerant without modifying the design, thereby reducing development cost and time. On the other hand, they are not flexible, since once a semiconductor device is fabricated with specific radiation-hardening processes, it may be challenging to modify or update the design to meet evolving requirements without significant changes to the manufacturing process.

RHBD techniques have been developed to address the impact of radiation on electronic circuits at different levels of abstraction ranging from circuit layout to system and software design. At the layout level, techniques focus on modifying the transistor's geometries and adding protective elements in order to reduce mainly TID and latch-up phenomena, while also offering protection against SETs and SEUs. Circuit- and system-based techniques, such as redundancy, error correction codes, memory scrubbing etc., basically deal with SEUs and SETs. In contrast to RHBP approaches, RHBD can entirely eliminate certain radiation-induced vulnerabilities. However, the trade-off is that RHBD methods typically lead to increased overhead in terms of layout area or greater design complexity.

In the last decade, the mitigation is shifting from process to design level, as the adoption of COTS components has increased, due to their performance, availability, cost and lead

time. In this context, design mitigation techniques are employed to ensure system functionality and performance under radiation effects. A description of the state-of-the-art radiation hardening techniques used in COTS APSoC devices is provided below. These techniques apply mostly at circuit and system architecture level. It should be noted that the best solution for protecting digital circuits from radiation effects is often a combination of several mitigation techniques.

3.3.1 Redundancy

The basic idea is to introduce additional components or resources beyond what is strictly necessary for normal operation. These redundancies can take various forms, and their purpose is to provide a backup or alternative in case of a failure. The common forms of redundancy presented hereafter can be applied to a digital hardware design, a memory block or a soft IP block, and even to application software or to system level [88].

3.3.1.1 Spatial Redundancy

Spatial redundancy is based on replicating sensitive functional modules and comparing their outputs to identify any inconsistencies. Any discrepancies between the outputs of the replicated modules are detected by a comparator or voter. Hence, as the decision whether the output is correct relies only on these elements, the comparator and voter are critical parts of this architecture. Spatial redundancy techniques can be categorized into two types: those that offer error detection only, and those that provide both error detection and correction [47].

Error detection without correction is typically implemented using duplex architectures, also referred to as dual modular redundancy (DMR). In this configuration, two identical processing units operate in parallel, and their outputs are continuously compared to identify discrepancies caused by SEEs. If a mismatch is detected, the system flags the error and prevents the propagation of incorrect data. However, DMR is fundamentally a fail-stop approach, capable of identifying faults but not resolving them. When both outputs are identical, the comparator assumes that the units operate correctly. If a divergence occurs, the system detects the error but cannot discern which output is valid. In such cases, recovery strategies may involve either discarding the current result and proceeding with the next operation or reprocessing the data to obtain a correct outcome, depending on the system's fault tolerance requirements and application criticality.

Triple modular redundancy (TMR), on the other hand, is a widely adopted and straightforward technique for both error detection and correction. TMR employs three redundant processing elements executing the same task concurrently. A majority-voting mechanism evaluates their outputs to determine the correct result. In the event of an upset, it is generally assumed that two of the three outputs will remain unaffected,

enabling the voter to identify and propagate the correct value. This technique enhances system reliability by tolerating a single fault without interrupting normal operation.

Spatial redundancy-based solutions can be applied to all functions of combinational and sequential logic of a digital design, embedded memories, as well as at system level, e.g., multiple processors performing the same task in order to compare their outputs, and thus, to detect and correct faults.

At the system level, redundancy can be implemented primarily through two approaches: TMR and lockstep. TMR, when applied at the electronic system level, follows the same spatial redundancy principle described previously; triplicating hardware components so that three identical operations are executed in parallel. The majority-voting mechanism then determines the correct output based on consensus among the three replicas.

Lockstep, in contrast, implements redundancy through parallel software execution using duplicated processors. In this architecture, a primary and a backup processor execute the same software independently but in a synchronized manner, without requiring any changes to the application code. Both processors typically share read access to memory, although variations exist where each processor may access separate memory spaces. Write access is generally restricted to the primary processor to maintain data integrity. The synchronization of these processors occurs through a mechanism known as a checkpoint, where a dedicated hardware checker compares the internal states of both processors at defined intervals. If no discrepancies are found at a checkpoint, the current processor state is stored as fault-free, and both cores continue execution. However, if a mismatch is detected, the checker triggers a rollback to the last known good state and coordinates recovery for both processors. This ensures that the system can continue its operation without significant disruptions.

3.3.1.2 Temporal Redundancy

Temporal redundancy involves sampling the same data at different time intervals, with an asynchronous comparator or voter used to determine the correct value. This can be achieved by instantiating the same function, such as a process or computation, multiple times at distinct time points, introducing a timing skew between the instances to enable fault detection and correction.

Alternatively, when the function to be protected is complex or too resource-intensive to replicate spatially, a more practical approach is to sequentially repeat the same computation on a single processing unit. This form of temporal redundancy does not require duplicating hardware, making it attractive for integration within constrained environments. The strategy involves re-executing critical software routines multiple times and comparing the outputs to detect errors. While this method introduces minimal hardware overhead, it incurs considerable time penalties. Additionally, implementing such techniques typically necessitates software-level modifications, which may not be

compatible with all types of applications, particularly those relying on interrupts or dynamic memory management.

Temporal redundancy can be employed at varying levels of granularity. At the instruction level, individual lines of source code are duplicated and re-executed. Task-level redundancy replicates entire functional blocks or processes, while application-level redundancy is applied to complete programs, often useful in cases where source code is unavailable, such as third-party software libraries or operating systems.

3.3.2 Error Correction Codes

Error correction codes (ECC) are algorithmic techniques designed to identify and, in many cases, correct errors in data by appending additional redundant information to the original data. When both detection and correction are performed, these techniques fall under the broader category of error detection and correction (EDAC). The primary objective of EDAC methods is to safeguard the integrity of data stored in memory cells or transmitted across communication channels. ECC operates by embedding parity or redundant bits within the data, enabling the recovery of the original information even if errors, within the limits defined by the specific code, are introduced during storage or transmission. Upon retrieval, the data is validated against the redundant bits to detect inconsistencies and, when possible, correct them.

ECC schemes are generally classified into two main categories: block codes and convolutional codes [47]. Convolutional codes are primarily used in data transmission systems, including digital video, mobile networks, and satellite communications. In contrast, block codes are typically used to protect data stored in memory. The ECC techniques considered in this thesis are based on block codes, which can be further subdivided into those designed solely for error detection and those capable of both detection and correction. This classification depends largely on the amount of redundant data. Each ECC has its own characteristics in terms of fault detection and fault correction. However, they all impact the system by increasing the area required to store redundant data and introducing additional time overhead for both processing the redundant data and verifying the consistency of the original data.

3.3.2.1 Parity Check

A parity bit is an additional binary digit appended to a set of bits to ensure that the total number of bits with a value of “1” conforms to a specified parity condition, either even or odd. As one of the most basic forms of error detection, parity checking provides a straightforward means to identify certain types of bit errors. There are two types of parity: even parity, in which the parity bit is set such that the total number of 1s (including the parity bit) is even, and odd parity, where the total number of 1s is odd.

This method enables the detection of any odd number of bit errors. However, it cannot identify errors involving an even number of flipped bits, as these do not alter the parity outcome, potentially masking the presence of data corruption. Consequently, while parity checks offer a minimal-overhead approach to error detection, they lack the capability to correct errors and are inherently limited in their detection capacity.

3.3.2.2 Cyclic Redundancy Check

A cyclic redundancy check (CRC) is an error detection cyclic code and non-secure hash function designed to detect accidental changes or errors in raw data. CRC is based on polynomial division in the finite field of binary numbers. It is characterized by the specification of a so-called generator polynomial, which is used as the divisor in a polynomial long division over a finite field, with the input data serving as the dividend, while the remainder becomes the result. CRC-32 is a widely-used algorithm used for detecting data corruption. Multiple variants of the algorithm exist, which have similar mathematical properties. The most common variant of the CRC-32 is based on the following polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$.

3.3.2.3 Hamming Code

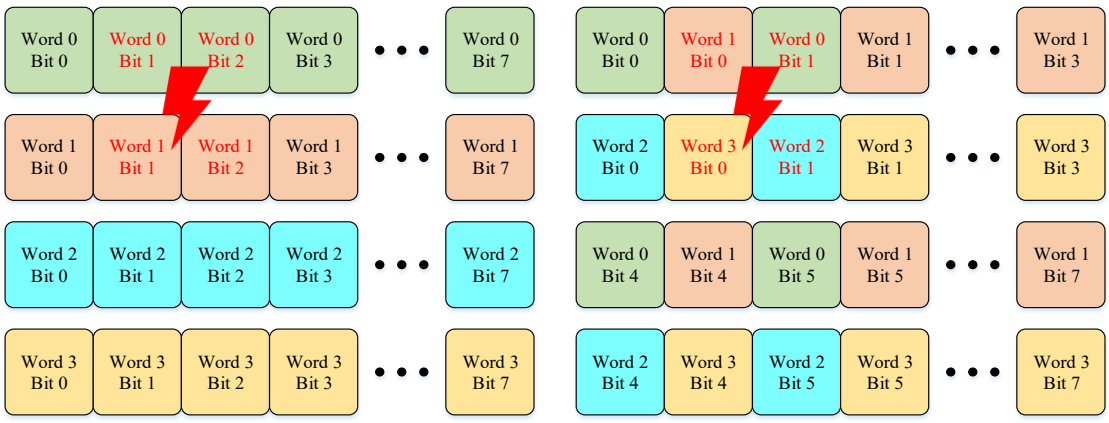
The redundancy in the Hamming Code is introduced through parity bits. Parity bits are added at specific positions in the data block to ensure that the total number of bits with a value of 1, including the parity bits, is either even or odd. The Hamming Code is designed to detect and correct errors by using parity bits. Hamming code employs a block parity technique, where data is divided into blocks, and parity bits are added to each block. This code is capable of correcting single-bit errors and detecting double-bit errors within a data block. The amount of parity data added to the Hamming code is given by the formula $2^p \geq d + p + 1$, where p represents the number of parity bits and d is the number of data bits. For example, if the number of data bits to be stored is equal to 4, the formula would be $2^3 \geq 4 + 3 + 1$, so 3 parity bits are required. The type of error correcting code is given as the name of the algorithm, the total number of bits in the block and then the number of effective data bits. Thus, Hamming encoding with 4 data bits and 3 parity bits for a total of 7 bits is given as Hamming (7,4). The minimum Hamming distance between valid codewords is crucial for error correction. Hamming distance refers to the minimum number of bit changes needed to convert one valid codeword into another. Hamming code is designed to have a Hamming distance of 3, allowing it to detect and correct single-bit errors.

3.3.3 Memory Bit-Interleaving

Memory blocks are particularly susceptible to radiation-induced effects, primarily due to their high density of memory cells. This makes them a critical focus for mitigating SEU phenomena. While spatial redundancy techniques can be effective in certain contexts, their application to large memory arrays is generally impractical due to the significant area overhead they impose. A more area-efficient alternative involves the use of error correction codes often in combination with bit-interleaving.

Standard EDAC schemes are typically capable of detecting two-bit errors and correcting one within a single data word. However, as transistor dimensions continue to shrink, the likelihood of MBUs increases, presenting new challenges for traditional correction mechanisms. Bit-interleaving addresses this issue by reorganizing the physical layout of data within the memory array such that bits that are logically adjacent in a word are physically dispersed across the array. This separation reduces the probability that a single particle strike will result in multiple upsets within the same data word.

As illustrated in Figure 3.7 (a), a conventional memory layout might experience an MBU that affects four adjacent cells across two data words, rendering standard ECC schemes insufficient for error correction, even if the errors are detectable. In contrast, Figure 3.7(b) demonstrates a memory array with bit-interleaving, where the same particle strike leads to isolated upsets across different words (e.g., words 0, 1, 2, and 3), enabling effective correction using existing ECC techniques.



(a) Standard memory topology (b) Example of memory bit-interleaving

Figure 3.7: Memory bit-interleaving

Selecting an appropriate ECC strategy depends largely on understanding the interleaving scheme used within the memory architecture. In some instances, this information is provided by the vendor. However, in other cases, the internal bit-interleaving

configuration remains undisclosed, complicating the implementation of tailored mitigation strategies.

3.3.4 Memory Scrubbing

An alternative method to address data corruption in memory blocks involves overwriting faulty memory bits with golden values, which are stored in a secure location, typically in rad-hard memory chip. This corrective process, known as data scrubbing [88], is carried out at regular intervals suited to the anticipated rate of radiation-induced errors. Its purpose is to prevent the accumulation of corrupted bits in memory regions that hold critical configuration data or other information that is not inherently or frequently refreshed during normal system operation.

Data scrubbing is widely used to refresh the configuration memory bits of SRAM-based FPGAs, without disrupting their operation. Given the vulnerability of FPGA configuration memory to radiation, periodic reloading of the configuration bitstream is essential. This ensures that faulty bits are continuously replaced by their original, error-free values, thereby mitigating fault accumulation and minimizing the duration during which the device operates under an invalid circuit configuration.

For example, in the case of Zynq-7000, as mentioned in 2.3.2, its PL configuration memory is divided into several frames. Each frame represents the smallest unit of resources that can be individually configured. Such a structure allows reconfiguring either the full device (full scrubbing) or only a part of the design (partial scrubbing).

However, it is important to note that scrubbing alone is insufficient for comprehensive protection of SRAM-based FPGAs against radiation-induced effects. While it effectively limits the accumulation of faults in the configuration memory, faults may still occur between scrubbing cycles and lead to application-level errors before the next refresh of the configuration memory. In addition, scrubbing will not correct faults in memories that are altered by the user application, i.e., user registers or embedded RAMs. Consequently, it is important to apply additional mitigation techniques in combination with scrubbing.

3.3.5 Other techniques

3.3.5.1 Watchdog Timer

Electronic systems based on a processor can suffer service or functional interruptions (SEFIs) due to the effect of radiation. In such a case, the system can be equipped to recover a normal operating mode on its own. A watchdog timer monitors continuously some signals, and whenever they are detected not operating as expected, i.e., there are corrupted data, resulting in erroneous information or wrong states in control circuits, it either performs a hard reset of a sub-system or flags the existence of a radiation-induced anomaly through an output signal in order to let the circuitry outside the system act

appropriately. Off-chip watchdog timers, integrated in a rad-hard device, can be used to implement monitoring. If a functional anomaly is detected, a system recovery action is performed, such as a reset or power cycle that can eliminate persistent incorrect states in the processor.

3.3.5.2 Latching Current Limiters

Latching current limiters (LCL) are active devices that protect the power lines of a system against overloading. Their primary function is to provide overload protection without generating dangerous voltage transients. In systems that are particularly vulnerable to single event latch-up (SEL), they play a crucial role in order to detect overcurrent phenomena and rapidly recover them by disconnecting the power supply to prevent permanent device damage. Unlike traditional current limiters, which continuously regulate the current, LCLs provide a mechanism to limit the current temporarily and then latch into a low-current state until reset.

3.4 Validation Methods

For systems designed to operate in harsh or safety-critical environments, it is essential to validate their performance under radiation exposure and assess their compliance with the operational constraints of the final environment. Although in-orbit or real-world testing offers highly accurate insights, such approaches are often time-consuming and resource-intensive. As a result, accelerated radiation testing on the ground is widely used, as it allows the simulation of natural radiation effects on integrated circuits by exposing them to high particle fluxes over short periods. Complementary to these physical tests, fault injection techniques are also employed, where radiation-induced faults are emulated to study system behaviour and evaluate robustness without the need for actual radiation sources.

3.4.1 Fault Injection

Fault injection is a technique to deliberately introduce faults or errors into a system to anticipate the system's behaviour when exposed to radiation during the design phase and validate the circuits and the embedded mitigation techniques. The goal of fault injection is to identify weaknesses, vulnerabilities, or failure points in a system and assess its ability to handle unexpected situations.

Fault injection encompasses various techniques, including code modification to introduce faults, manipulation of input data to induce specific error scenarios, and the utilization of specialized tools and hardware to simulate hardware faults.

Various implementations are feasible for the fault injection system categorized according to the type of the target system [68]. These implementations include:

- Simulation-based fault injection: Encompassing methods designed to introduce faults into a model of the target system, which is then analyzed, using simulation tools. Typically, the target system is represented as an executable model written in hardware description languages such as VHDL, Verilog, or programming languages like C/C++. A testbench, acting as a workload generator, provides the necessary input stimuli to activate the model within a simulation environment. These simulation platforms are often COTS tools frequently reused from the standard design and verification phases. In some cases, the workload generator may be an stand-alone software module that operates on top of the simulation tool. Due to the inherently time-consuming nature of simulation and the increasing complexity of modern digital systems, simulation-based fault injection is best suited for exploring a limited number of strategically chosen fault scenarios, primarily to detect possible design bugs. When very complex workloads and huge amounts of faults have to be injected, the limit of the capability of simulation-based fault injection will be reached. Simulation time will be unfeasible and alternative solutions to speed up injection campaigns will be needed.
- Emulation-based fault injection: Emulating the system using dedicated hardware, such as FPGA-equipped boards. It is an attractive technique, utilizing hardware emulation to enhance simulation performance. It is particularly advantageous in scenarios involving intricate workloads and large-scale fault campaigns, where the time required for each fault-injection experiment can be a limiting factor. In this methodology, the FPGA is configured to instantiate a functional prototype of the system under test, enabling evaluation to be conducted directly on reconfigurable hardware. A host computer orchestrates the fault injection process, executing the software modules responsible for fault management and experiment control. This approach is very effective, primarily due to the acceleration that hardware emulation can offer compared to software-based simulation. In many cases, the execution speed of an emulated hardware model can surpass its simulated counterpart by multiple orders of magnitude. However, the interface between the host computer and the FPGA introduces a critical dependency: input stimuli and observed outputs must traverse this communication channel. Consequently, the bandwidth and latency of this link can become a limiting factor, potentially constraining overall performance. To improve further the performance, the fault injection and data collection mechanisms can be implemented in hardware into the FPGA board, thus assigning only the supervising task to the host computer.
- Software-based fault injection: Employing a physical model of the system comprising processors, memories, and I/O peripherals and fault-injection

software routines embedded within an executable representation of the system under study. This technique is particularly effective when a working prototype of the target system is available, and it is especially applicable to architectures built around general-purpose or embedded processors. While this method enables precise targeting of user-accessible architectural elements, it inherently lacks visibility into internal resources. This limitation becomes more pronounced in advanced processor designs, where many critical internal components are not directly accessible through software. Implementing software-based fault injection requires modifications to the system's software stack, typically by inserting lightweight injection handlers. These handlers are designed to consume minimal memory, thereby imposing little overhead on the system's footprint. Nevertheless, the mechanism used to trigger fault injection, particularly if it relies on specialized processor features such as debug mode, can introduce non-negligible performance penalties. Therefore, careful consideration is needed in balancing injection control flexibility and runtime efficiency.

3.4.2 Accelerated Radiation Tests

Accelerated testing is conducted in specialized facilities such as synchrotrons and linear accelerators, generating significant particle fluxes, including protons, heavy ions, neutrons, gamma rays, and alpha particles. While these facilities cannot precisely replicate space particles in terms of energy, they are calibrated to produce particles with similar linear energy transfer (LET) for heavy ions and energy for protons, mimicking the natural space environment.

The radiation source used for SEE tests shall be a particle accelerator capable of delivering the required flux and fluence of heavy ions or protons of suitable LET and energy [48]. Flux is the rate of particle flow, i.e., the number of particles passing through a unit area perpendicular to the beam in one second, while fluence is the flux integrated over time. The heavy ion accelerator shall be capable of delivering ions with variable flux ranging from a few 10 ions/cm^2 to at least 10^5 ions/cm^2 per sec on the device under test. In the case of high-energy proton tests, SEEs are induced by the ionizing by-products of nuclear interactions between incident high-energy protons and the tested device. The high-energy proton accelerator shall be capable of delivering protons in the energy range from 20 to 200MeV with a variable flux ranging from 10^5 protons/cm^2 to at least 10^8 protons/cm^2 per sec on the device under test. Exposure is usually performed with the device surface positioned vertically to the beam axis. However, some devices have been observed to be more sensitive when positioned with an angle of incidence. It is recommended to check the effect of the angle of incidence on the SEE sensitivity.

For instance, in an accelerated radiation test conducted with a flux of 10^5 ions/cm^2/sec over a duration of one hour (3,600 secs), the total fluence reaches 3.6×10^8 ions/cm^2. This controlled, high-intensity exposure simulates the cumulative radiation effects

experienced over extended periods in natural environments. On Earth's surface, where cosmic ray flux is approximately 10^{-2} ions/cm²/sec [61], this test replicates roughly 1,140 years of exposure. In LEO, where the flux is approximately 1 ion/cm²/sec [26], the same test equates to about 11.4 years of orbital radiation. This accelerated approach allows researchers to rapidly assess the long-term radiation tolerance of materials, electronic components, or biological samples, providing crucial data for the design of space missions and radiation-hardened technology.

Two types of tests can be performed to assess the sensitivity of the device under test (DUT) to upsets, particularly in circuits with memory elements like processors, ASICs, FPGAs, and memory devices, the static tests and the dynamic tests.

In a static test, all memory elements of the DUT are initialized before exposure to particles. The device is powered but is in idle mode, and in some cases without clocking, during exposure. After a period, the particle beam is deactivated, and the DUT's memory cells are compared against the expected values. While this represents a worst-case scenario as all memory elements are tested, real applications typically do not utilize all resources simultaneously, and the content of each memory element may not be critical at any instant. The static test characterizes the device itself, independently of its final application.

A dynamic test aims to evaluate the error rate of a system operating under conditions similar to its final application. This involves running the final application or specific benchmarks under the particle beam until an error is detected in the application's outputs. Dynamic testing is more intricate than static testing but is recommended for detecting various errors that may arise in complex applications. The downside is that any changes in the application can imply the need to perform a new test campaign.

3.4.3 Real-life Radiation Tests

Real-life radiation tests involve operating the target system within a natural radiation environment, which can be the final operational environment. Obtaining results from these experiments can be time-consuming due to the relatively low particle fluences present in natural environments. While one approach to expedite the process is to increase the sensitive volume exposed to the particles, this is frequently impractical given the constraints of embedded missions. Along with the cost of such experiments, this explains why alternative solutions, such as accelerated tests or fault injections, are often preferred.

4 Single Event Effects on the Zynq-7000 PL

In order to study the feasibility of using AMD Xilinx Zynq-7000 in space applications, it is necessary to perform different experiments under various heavy ion and proton ranges for the space environment emulation. Last years, several studies of the radiation effects in Zynq-7000 PL have been presented in the literature, investigating the SEUs in the CRAM and BRAM of the device FPGA fabric by performing experiments under heavy ions and protons. However, these experiments do not cover the necessary ranges of heavy ion LETs and proton energies to guarantee accurate prediction of the SEE rates in space. In [22] and [135], the authors tested the Zynq-7000 CRAM and BRAM for SEUs using heavy ions with LETs from 2 to 22 MeV·cm^2/mg and from 2.6 to 17 MeV·cm^2/mg, respectively. On the other hand, in [40] and [136], proton irradiation experiments were performed to characterize the SEE sensitivity of CRAM memory only with energies from 50 to 250 MeV and from 3 to 10 MeV, respectively.

This chapter presents the impact of radiation-induced SEEs in the Zynq-7000 PL and an in-depth analysis of the SEE susceptibility of all the PL memories, i.e., CRAM, BRAM, FFs, LUTs. Various radiation experiments were performed in different test facilities with ranges from the minimum LET and energy threshold required to occur single events in the target device up to the point that the device cross section saturates. These experiments enabled the analysis of phenomena for different energy levels and LETs, i.e., heavy-ion LETs from 2.29 to 12.45 MeV·cm^2/mg and protons energies from 30 to 200 MeV. The analysis of the radiation experimental results provided profound insight into the SEU vulnerability of the device and enhanced understanding of various SEE phenomena observed in it, such as SEFIs, SETs and multiple-bit upsets that can be key issues for the design of an effective SEE mitigation approach.

The key points of this study include:

- The implementation of a synthetic benchmark design that fully utilizes the PL resources to improve the statistical insignificance of the radiation experiments.
- The calculation of the cross sections for all the different memory types of the PL part including the application layer (i.e., FFs, SRLs, BRAM) and configuration layer (i.e., CRAM).

- The identification of SEFIs and upsets that occurred in SRLs and FFs due to SETs in global signals (e.g., clock tree), and in FFs and BRAMs due to single-event resets effects.
- The estimation of the CRAM essential bit cross section, which provides a more realistic probability metric of SEUs affecting the programmable logic behavior, taking into account only the upsets in the essential configuration bits.
- The analysis of how the SEUs are affected by the memory contents, i.e., the likelihood to occur upset in memory cells preloaded with '0' or with '1'.
- A detailed analysis of the single-event multiple upsets (MBUs and MCUs) observed during radiation experiments in the various memory categories (e.g., CRAM, FFs, SRLs, BRAM), which may be useful for designing effective mitigation techniques.

4.1 Radiation Tests

4.1.1 Test Facilities

The radiation test campaigns required extensive preparation, including hardware/software development, validation, and verification, to ensure reliable operation under irradiation. Experiment planning was particularly challenging, involving long-term coordination with different international facilities, strict beam-time schedules, and compliance with facility-specific safety and operational procedures. The experimental setup had to be deployed and validated within very limited beam-time windows, requiring efficient integration, rapid troubleshooting, and real-time decision making.

The radiation tests were performed a) under heavy ions at the CERN Super Proton Synchrotron North Area (SPS-NA) in Geneva, Switzerland and at the GSI Helmholtz Centre for Heavy Ion Research in Darmstadt, Germany and b) under high energy protons at the Paul Scherrer Institute (PSI) – Proton Irradiation Facility (PIF) in Villigen, Switzerland.

The CERN SPS-NA is a large facility with four beam lines. Ions are completely stripped off their electrons when arriving at the SPS-NA, while they can reach a momentum in the range of 13 – 150 GeV/u. The Ultra-High-Energetic lead ion beams, included in the 2018 CERN experimental program [3], [89], were used to carry out radiation tests on electronics. The first radiation test campaign was performed under heavy ions (Pb) irradiation. The ultra-high energy of the lead ions for this campaign reached 150 GeV/u. The beam arrives at the facility in form of spills of 10 secs, with a periodicity of 41 secs approximately. During the tests, a beam size of 40 mm × 40 mm was set, and the beam flux was varied between 1×10^2 and 2×10^3 ions/cm^2 per spill.

The second test campaign took place at the beam line SIS18 of the GSI in the context of the FAIR Phase-0 research program using iron ions (Fe) with a very high energy of 230 MeV/u. GSI is a unique laboratory worldwide that has the capability to generate intense heavy ion beams at the heavy ion synchrotron facility SIS18, which operates in the range of 0.010 – 2 GeV/u. The beam arrives at the facility in form of spills of 1 sec length, with a periodicity of 3 sec. A beam size of 31 mm × 31 mm was set, and the beam flux was around 10^3 ions/cm^2 per spill.

It must be noted that the very/ultra-high energy of the ion beams in both facilities provides high penetration (GSI: 13.43 mm and CERN: several cm) allowing for testing in air, with packaged parts, while the same LET is maintained crossing the sensitive volume of interest [89]. Especially in CERN, the very high penetration potentially enables a) the irradiation of multiple boards in parallel; e.g., in the current test campaign, six boards from different experiments were tested in parallel and b) tilting the boards up to large angles; e.g., the boards were tested in 0^o and 45^o angles of incidence.

	CERN SPS-NA	**GSI SIS18**	**PSI PIF**
Date of campaign	16-18 Nov 2018	15 April 2019	10-12 Sep 2021
Irradiation type	Heavy-Ion Pb	Heavy-Ion Fe	Proton
Energy	150 GeV/u (ultra-high)	230 MeV/u (very-high)	30, 50, 100, 150 and 200 MeV
Flux	10 to 2×10^2 ions/cm^2/sec	10^3 ions/cm^2/sec	1.8×10^7 to 1.2×10^8 p/cm^2/sec
Size	40 mm × 40 mm	31 mm × 31 mm	40 mm × 40 mm
Period	~ 40 sec (10 sec on + 30 sec off)	~ 3 s (1 sec on + 2 sec off)	

Table 4.1: Radiation test facilities

The third radiation test campaign took place at PSI-PIF, which is specialized for testing the spacecraft components. Specifically, the facility enables the generation of realistic proton spectra encountered at any potential orbit in space and provides mono-energetic beams for radiation hardness tests of materials. The target device was irradiated under five different proton energies: 30, 50, 100, 150 and 200 MeV. The beam spot was rectangular with a size of 40 mm × 40 mm. The beam was located in the center of the target area. The beam flux ranged from 1.8×10^7 to 1.2×10^8 p/cm^2/sec depending on the proton energy.

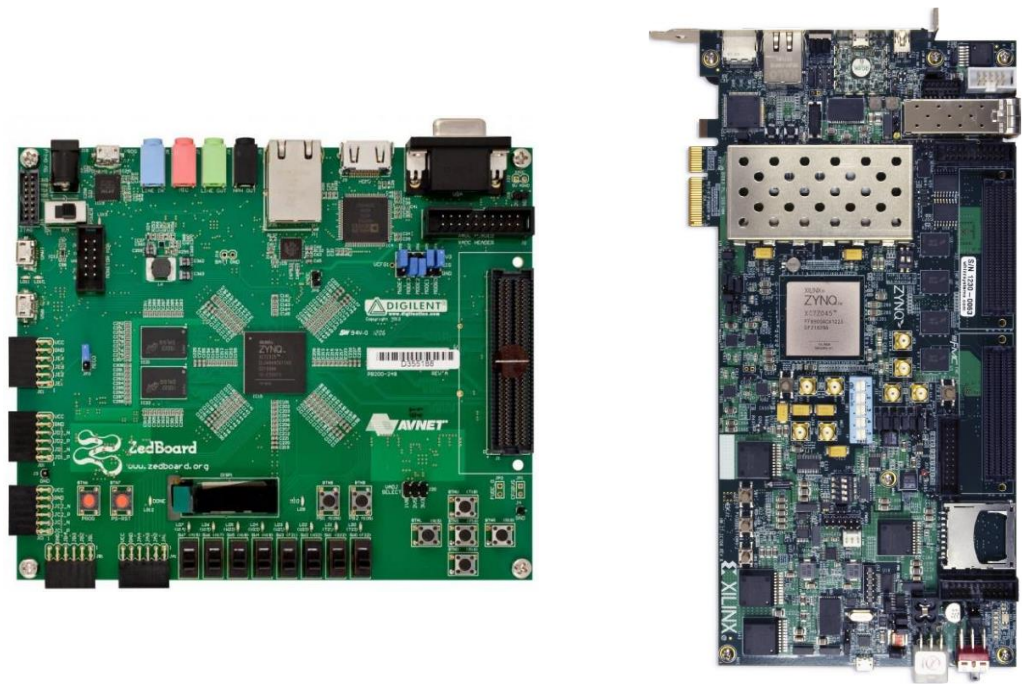
The beam features, flux, size and spill periodicity, for the test facilities are summarized in Table 4.1.

4.1.2 Test Setup

Two different Zynq-7000 parts, whose PL belongs to the 7-Series technologies Artix-7 and Kintex-7, were selected for our experiments: the XC7Z020 (Artix-7) and the XC7Z045 (Kintex-7). Artix-7 uses the same fabric resources as Kintex-7 but is optimized for even lower power consumption and smaller packages, delivering similar advantages. The PL application layer features of both devices [15] are shown in Table 4.2.

PL features	XC7Z020	XC7Z045
Xilinx 7-Series PL Equivalent	Artix-7 FPGA	Kintex-7 FPGA
Logic Cells	85K	350K
CLB Flip-Flops	106,400	437,200
Total CLB LUTs	53,200	218,600
CLB LUTRAMs	17,400	70,400
Block RAM	4.9Mb (140 x 36Kb)	19.2Mb (545 x 36Kb)
DSP Slices	220	900
IO	200	362

Table 4.2: Zynq-7000 PL features



(a) (b)

Figure 4.1: Avnet ZedBoard & AMD Xilinx ZC706

The two Zynq-7000 devices are attached to commercial evaluation boards, the Avnet ZedBoard [25], shown in Figure 4.1(a) and the AMD Xilinx ZC706 [17], shown in Figure 4.1(b). Both boards are low-cost development boards that contain several expansion connectors providing easy access to the processing system and programmable logic I/Os. They have a JTAG interface that allows a host computer to implement various configuration memory functions, such as bitstream programming, readback or partial configuration frame write and read using the AMD Xilinx Vivado software or custom software to access the Zynq device. JTAG is the highest priority configuration interface and will always be able to access the device regardless of the boot mode strapping pins [16].

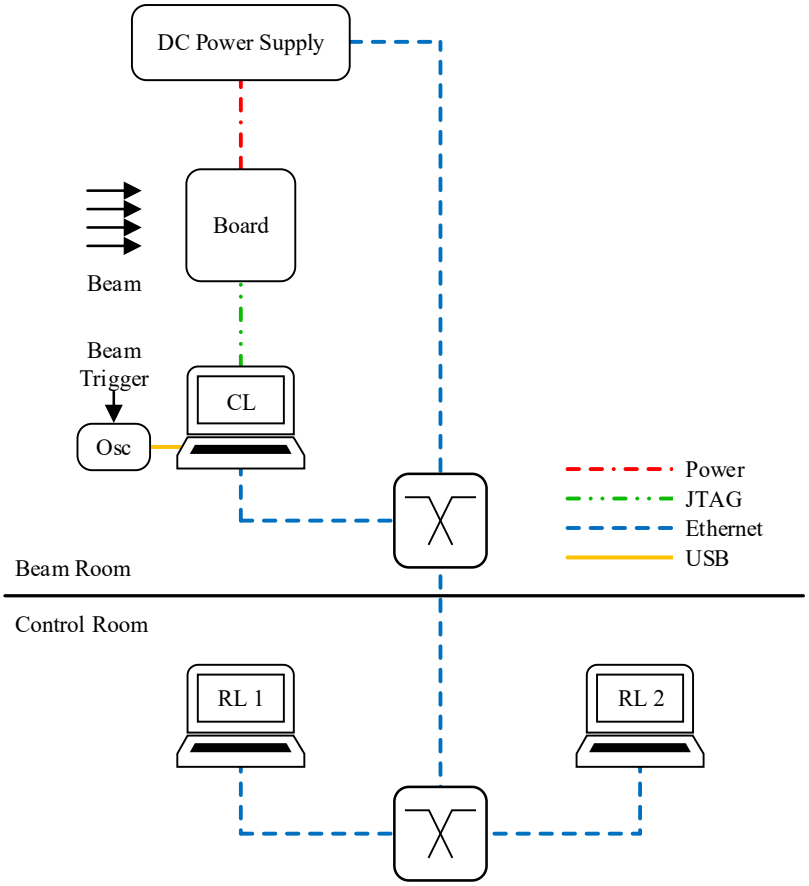


Figure 4.2: Radiation test setup

Figure 4.2 presents the principal layout of the test setup for the three radiation test campaigns performed in CERN, GSI and PSI. The tests were executed in the beam room but remotely monitored and controlled from a separate area, the control room. The test setup includes the target Board, one DC power supply unit (PSU), one control laptop (CL) located in the beam room for control and monitoring purposes, two remote laptops

(RL) located in the control room for remote control and several interfaces. In CERN, the DUT irradiated was the XC7Z020 (ZedBoard), while in GSI and PSI the DUT was the X7Z045 (ZC706).

The main test operations are summarized below:

- The Board is powered by the DC PSU (CERN/GSI: Keysight N6705, PSI: Siglent SPD3303X-E) remotely controlled by RL-2. RL-2 monitors the Ethernet-interfaced PSU to power cycle the board during, if any, high-current events.
- The CL configures and reads back the DUT configuration memory through the PL JTAG port and performs data logging.
- The CL is remotely controlled and monitored by RL-1.
- In the CERN setup, the beam line is monitored by a portable USB Oscilloscope (Digilent Analog Discovery 2) connected to the CL. A beam trigger signal indicates the start and the end of the irradiation periods (1: beam on, 0: beam off). When the Oscilloscope detects a falling edge in the beam trigger signal, it identifies the end of spill. This event detection triggers the host application running on the CL to allow DUT configuration, data acquisition and logging while the beam is off. On the other hand, in the GSI and PSI setups, the tests run continuously regardless of whether the beam is off or on.
- All system clocks are synchronized before the experiments with the facility's clock. All instruments' (PSU, CL, RLs) logs are time-stamped for post processing.

It is worth mentioning that before irradiation begins, critical checks must be performed, such as the alignment of the experimental setup with the beam source, and the clock synchronization of the setup and facility to a common reference time, to enable accurate correlation between setup logs and facility logs. In addition, an initial verification must be performed, to confirm that the observed error rate is consistent with the applied beam flux, thereby validating the correct operation of the device under test, the data acquisition chain, and the overall test setup before a significant radiation dose is accumulated.

Figure 4.3 photo depicts the ZedBoards connected in the CERN beam room. Actually, multiple boards were irradiated in parallel using a metal supporting plate with the UCL frame dimensions. The front boards were for other radiation experiments. Our ZedBoards were the last two boards in line. The first ZedBoard corresponds to the test campaign described in the current chapter, while the other ZedBoard corresponds to the work presented in chapter 6.

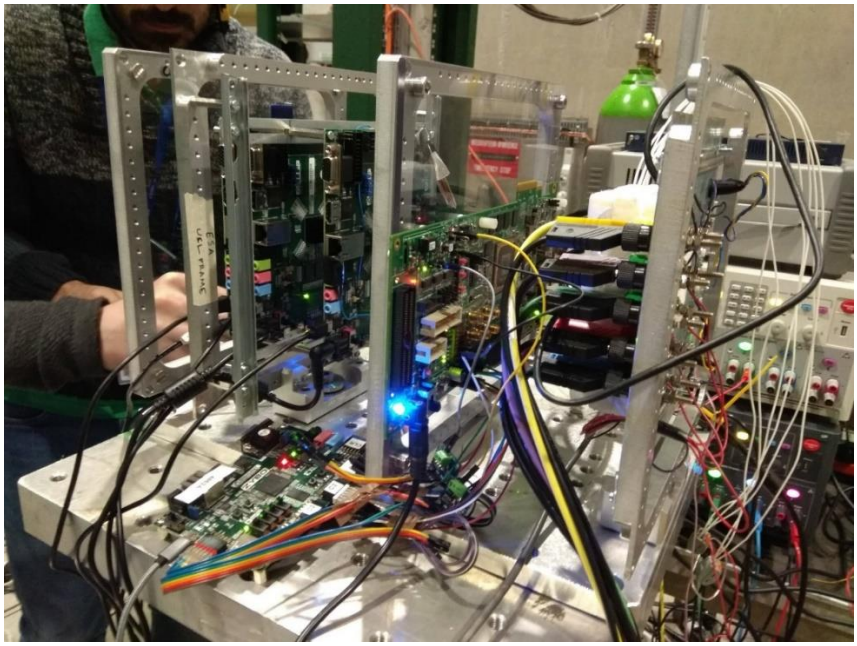


Figure 4.3: Heavy-ion beam experiment at the CERN facility

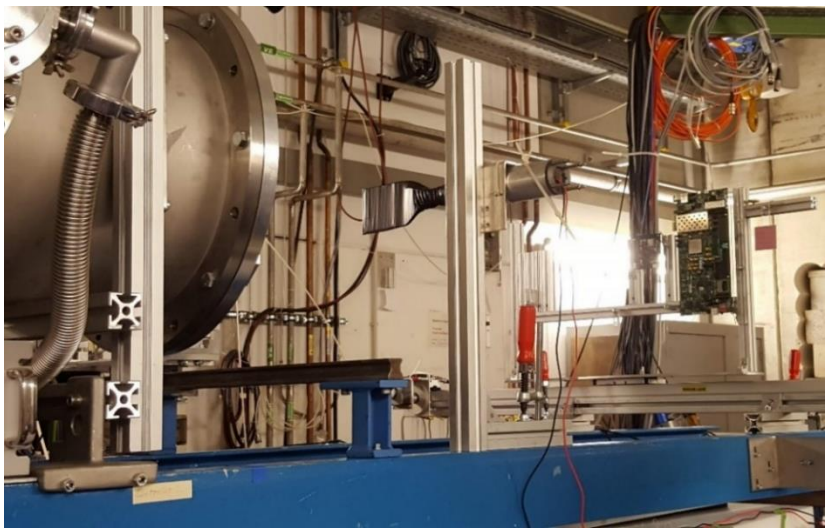


Figure 4.4: Heavy-ion beam experiment at the GSI facility

Figure 4.4 and Figure 4.5 show the ZC706 board irradiated in the beam room of the GSI and PSI facility, respectively.

Figure 4.5: Proton beam experiment at the PSI facility

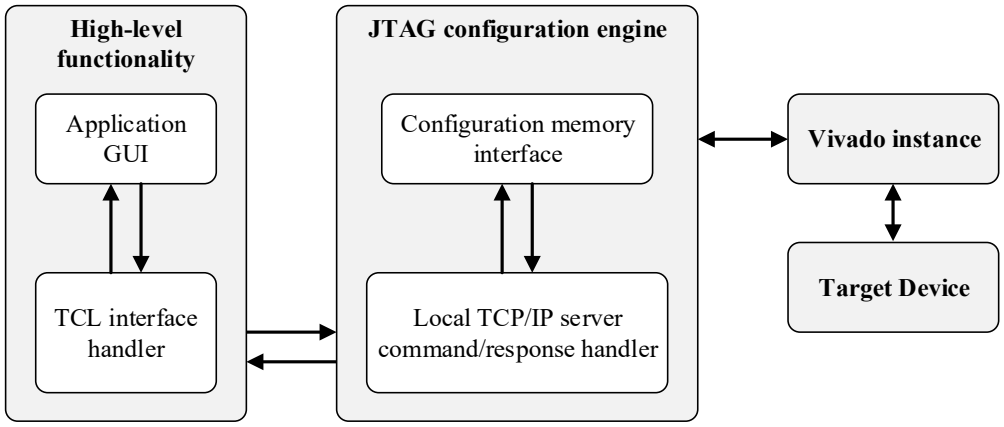


Figure 4.6: FREtZ architecture

All JTAG transactions with the target board, generated by the CL, are performed through the open-source FREtZ software platform [125], [145]. FREtZ, as mentioned in subsection 1.3.1, is a GUI-based software running on a Windows® or Linux host machine developed by the ESLab team for the purposes of fault injection and accelerated radiation testing and enables comprehensive SEE assessment and mitigation strategies. The software architecture is shown in Figure 4.6.

It is composed of two major components:

- JTAG configuration engine (JTAG-CE): provides the implementation of a TCP server and low-level JTAG functions, in the form of TCL functions, for accessing the FPGA configuration memory through the Vivado tool.
- High-level configuration functions (GUI): consists of the user application, which supports widget implementation, and the interface functionality (TCL interface handler) with the JTAG configuration engine.

The user application has been designed using the Qt and PySide2 frameworks. Qt is a well-known and highly-appreciated cross-platform application development framework for creating graphical user interfaces, while PySide2 is a Python binding for Qt which allows developers to use the rich set of functionalities provided by the native implementation of Qt using the Python programming language. Figure 4.7 shows the GUI used for the radiation tests.

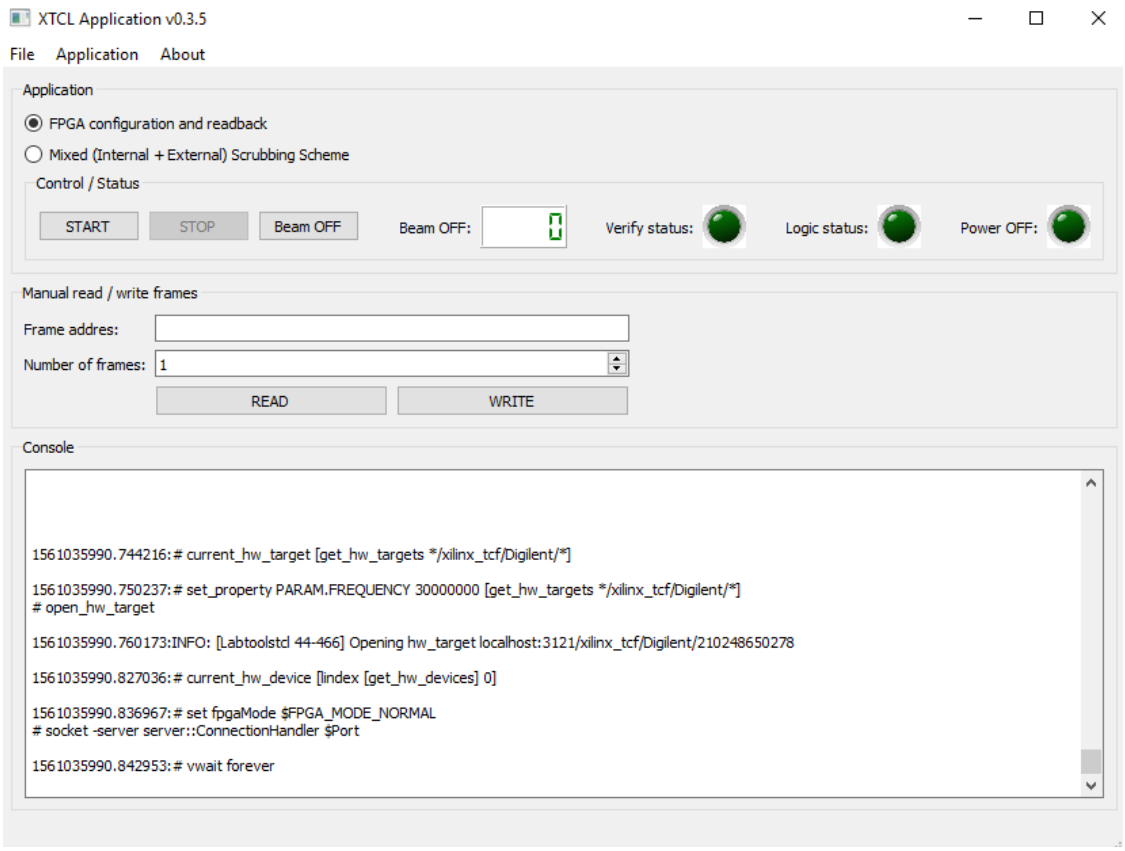


Figure 4.7: Test framework GUI for the radiation tests

The user application uses the APIs exposed by the TCL interface handler to communicate with the JTAG-CE in order to read/write the device configuration memory and configuration registers. Initially, the TCL interface handler starts a Vivado instance in

batch mode and a TCP client. The TCP client is used as inter-process communication (IPC) between the application thread and the JTAG-CE, which runs in a separate thread as a TCP server. The use of TCP client-server scheme significantly improves the execution time overhead introduced by the Vivado instance. Instead of running a Vivado instance upon every script/command execution, the framework creates a single instance during the execution lifetime of the application, eliminating thus the overhead for setting-up the JTAG connection for every script.

The low-level APIs provided by the JTAG-CE run in a separate thread as the TCP server listens in a specific TCP port. The JTAG-CE accepts predefined commands from the target application and executes the associated low-level API while it sends back a response to the application asynchronously. This means the user application should send a dedicated command using the TCP client, wait asynchronously for the response from the JTAG-CE component and process the response as needed. Table 4.3 describes the low-level APIs implemented in the JTAC-CE as well as the execution times of the FPGA configuration memory access functions for the XC7Z020 and XC7Z045 devices with the serial JTAG interface operating at 30 MHz clock rate.

API commands	**Description**	**Execution time (sec)**	
		XC7Z020	XC7Z045
Configure	Configures the FPGA device given a bitstream file	1.45	5.00
Readback	Reads-back the device's configuration memory and saves its content in a target file	4.10	13.40
ReadbackCapture	Reads-back the device's configuration memory in capture mode and saves its content in a target file	4.18	13.50
ReadbackVerify	Verifies the FPGA against the configured bitstream file while using also the mask file	14.11	-
RegisterWrite	Writes a configuration register	0.016	0.016
RegisterRead	Reads a configuration register	0.020	0.020

Table 4.3: Low-level APIs for the configuration memory access

4.1.3 Circuit Under Test

The predominant method for measuring the SEU static cross section of the Zynq PL CRAM involves readback of the CRAM, while the device has been programmed with a thrifty design or no design at all. However, the goal is to use a design to help observe further phenomena, in order to aim for an in-depth analysis of all possible radiation-induced upsets in the device family, as the transients in the clock or reset signals. In addition, no design means that the vast majority of CRAM will be initialized to 0, i.e., bit-flip from 0 to 1 will be more likely than 1 to 0, which is not representative. Thus, a synthetic, parameterized benchmark that fully utilizes the PL resources was designed to be used during radiation tests. The benchmark instantiates all available FFs, LUTs, BRAMs, DSPs and the most IOs, providing a highly utilized and densely routed design by increasing sample size for statistical reasons. Additionally, specific parameters have been forecasted for the benchmark so as to capture different types of SEEs.

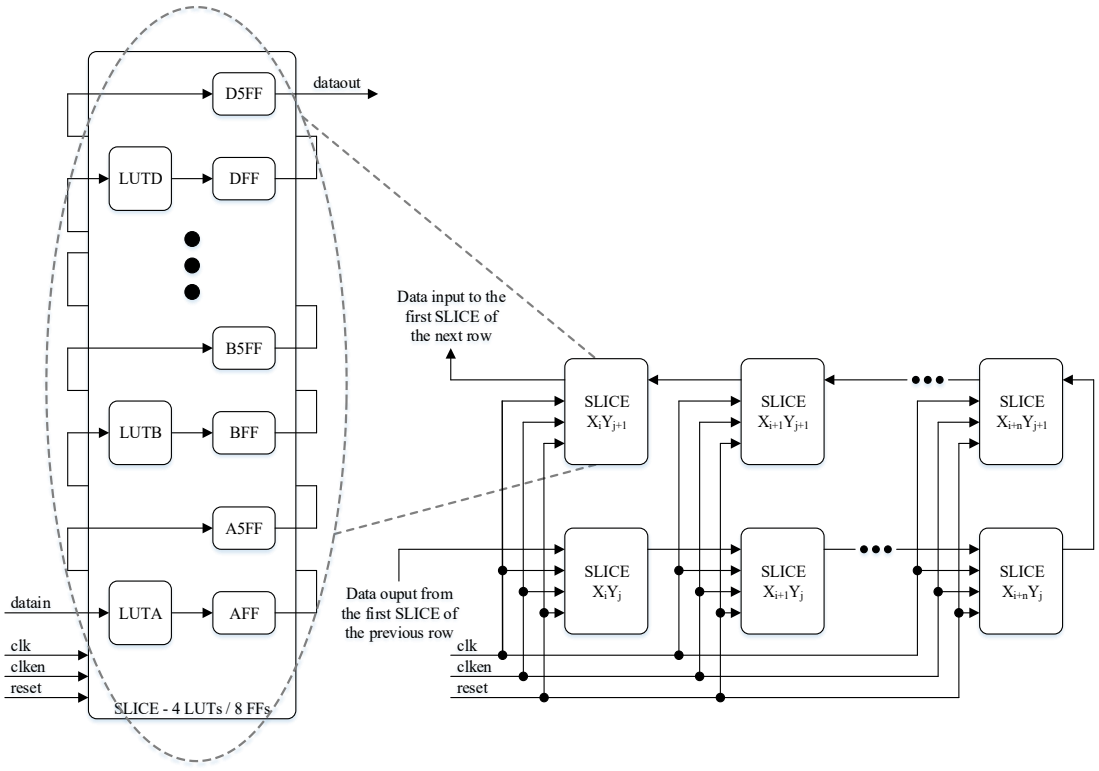


Figure 4.8: Horizontal SLICE chain arrangement

Specifically, all CLB SLICEs are connected to form a long register chain, as shown in Figure 4.8. The four CLB LUTs drive four of the eight CLB FFs of each SLICE and the latter, in turn, drive the other four CLB FFs of the SLICE. In the case of SLICEL, the LUTs are configured as route-through, while in the case of SLICEM, the LUTs are

configured as 32-bit SRLs, creating an 8-stage register or a 136-stage register (i.e., 4 x 32-bit SRLs plus 8 FFs), respectively per SLICE. All SLICEs are interconnected in cascading fashion either horizontally, i.e., across site rows of the device or vertically, i.e., along site columns of the device. The horizontal or vertical arrangement, which can be assigned at synthesis compile time as a parameter of the register-transfer level (RTL) design, creates a flexible and reusable benchmark that can adapt to different specifications, requirements, and scenarios. The horizontal SLICE chain arrangement is depicted in Figure 4.8.

The clock, clock-enable and reset signals of all SLICEs are routed to device inputs, which are physically constrained during the implementation flow by applying a weak logic high (i.e., pull-up) or low (i.e., pull-down) level to avoid floating when not being driven. In the static test case, the register chain is not clocked; the clock input is pulled down while clock-enable can be set either '1' or '0'. Setting the clock-enable to high enables the detection of upsets caused by SETs on the clock tree. Conversely, setting it to low greatly reduces the likelihood of SET-induced upsets, as it is rare for an event to disrupt both the clock and clock-enable signals simultaneously. The circuit can also be used in dynamic tests by driving the register chain with a clock pulse and setting the clock-enable to high. On the other hand, the reset signal is only applied to FFs and not to SRLs and can be configured as synchronous reset or asynchronous reset/preset. The FF reset input is pulled down into the IOBs. Given that the clock is disabled, transients in the reset signal may cause upsets in the FFs when it has been configured as asynchronous. If transient detection is not required, the reset must be configured as synchronous.

In order to be able to recognize the events that occur in the user design upon irradiation, the long shift register is initialized with specific values. The FFs are preloaded with alternate 0 and 1, while the SRLs with continuous 0s and 1s or alternate 0-1 patterns. For example, an event in the reset or preset signals during the static tests causes all the register's states to become low or high, respectively. Thus, it is easy to recognize it, since, for example, the sequence "010101" becomes "000000". Similarly, an event in the clk signal during the static tests causes the data stored in the chain to be shifted from one location to the next, e.g., "010101" becomes "101010".

In another variation of the above design that uses the SLICEs in chain mode, the SLICEM LUTs are configured as DRAMs instead of SRLs. The main difference is that the SRL uses the master-slave scheme presented in Figure 4.9, i.e., it is implemented as the combination of two latches, where the first latch acts as master and the second one acts as slave, occupying a 64-bit LUTM to form a 32-bit SRL. The output of the master latch is connected to the slave latch. The clock signal is connected to the master latch but is applied to the slave latch through an inverter. That means that when the clock signal is high, then the master is enabled and the slave is disabled. And similarly, when the clock is low, then the slave is active and the master is disabled. In the case of a static test, where the register chain is clock-free and the clock is set to low, the master latch is always closed while the slave latch is always open. On the other hand, DRAM uses all the 64

bits of a LUTM as memory cells. Thus, when the clock and the write-enable signals are set constantly to ‘0’, DRAM upsets due to transients in the clock or control signals are unlikely to occur.

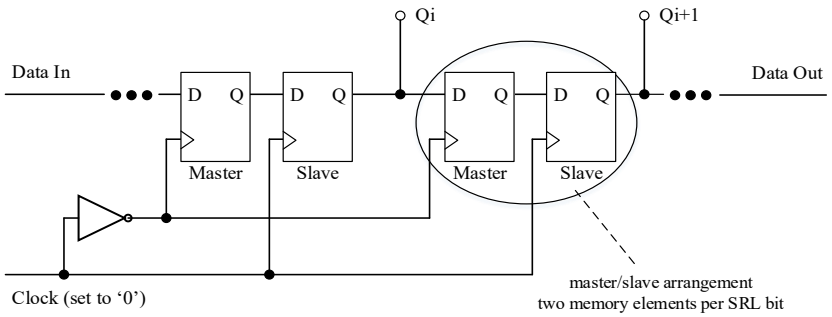


Figure 4.9: LUT SRL master-slave scheme

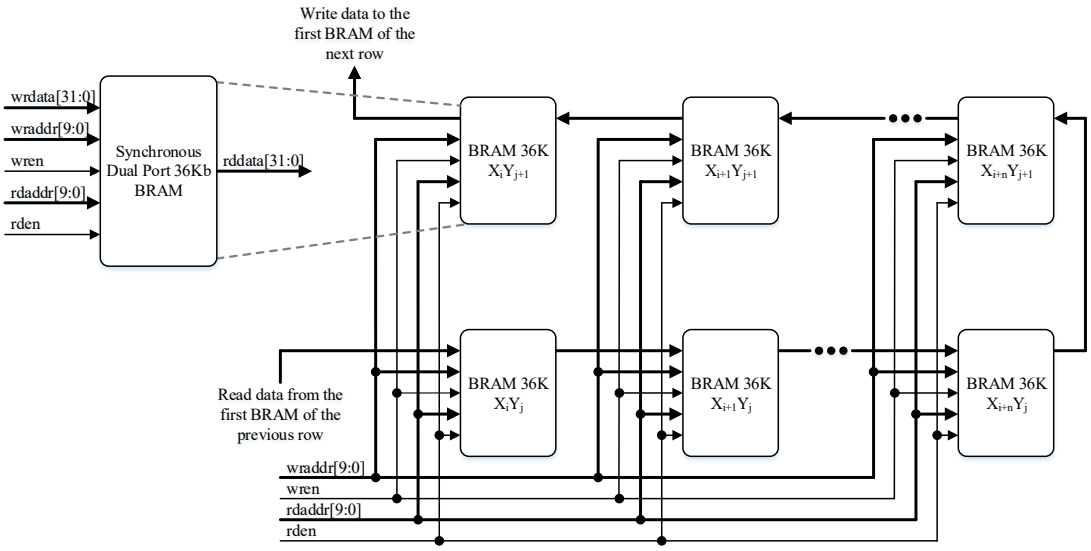


Figure 4.10: Horizontal BRAM chain arrangement

The BRAM units are also arranged in a chain mode, as shown in Figure 4.10. All BRAMs are configured as a synchronous dual port 36Kb. Two common address buses, one for the write path and another one for the read path, are used to access the 1024 32-bit words of each BRAM. The write- and read-enable control signals are also common and routed to all memory units. The 32-bit data bus is designed in such a way that the read data of the preceding memory feeds the write data of the following memory, creating the sequence of memories. The BRAMs are cascaded through the data bus either horizontally or vertically, as in the case of CLB SLICEs.

The clock and write/read-enable signals are routed to device inputs by applying a weak pull-down, so that upsets in the memories due to transients in the clock or control signals are unlikely to happen during the static tests. The address buses are also set to low into IOBs to prevent floating. To observe SEUs in the BRAMs, they are initialized with a predefined pattern during device configuration, i.e., data are set to all 1s or all 0s or checkerboard values and parity bits to 0s. The goal is the 0s and 1s percentages of the device's initial BRAM data values to be balanced and equal to 50%, in order to observe the likelihood of upsets in memory cells preloaded with '0' or with '1'.

The Zynq-7000 DSP48 slices support many independent functions, including multiply, multiply-add, multiply-accumulate (MAC), and three-input add (Figure 4.11). The DSP slices of the benchmark are configured to implement the representative operation of multiplication and are cascaded, as depicted in Figure 4.11. The 40-bit P data output of each DSP slice is connected to the A and B data inputs of the next DSP in the chain forming either a horizontal or vertical chain arrangement, as follows: the P[24:1] output bits drive the A input, while the P[39:25] and P[1:0] output bits drive the B[16:2] and B[1:0] input bits, respectively.

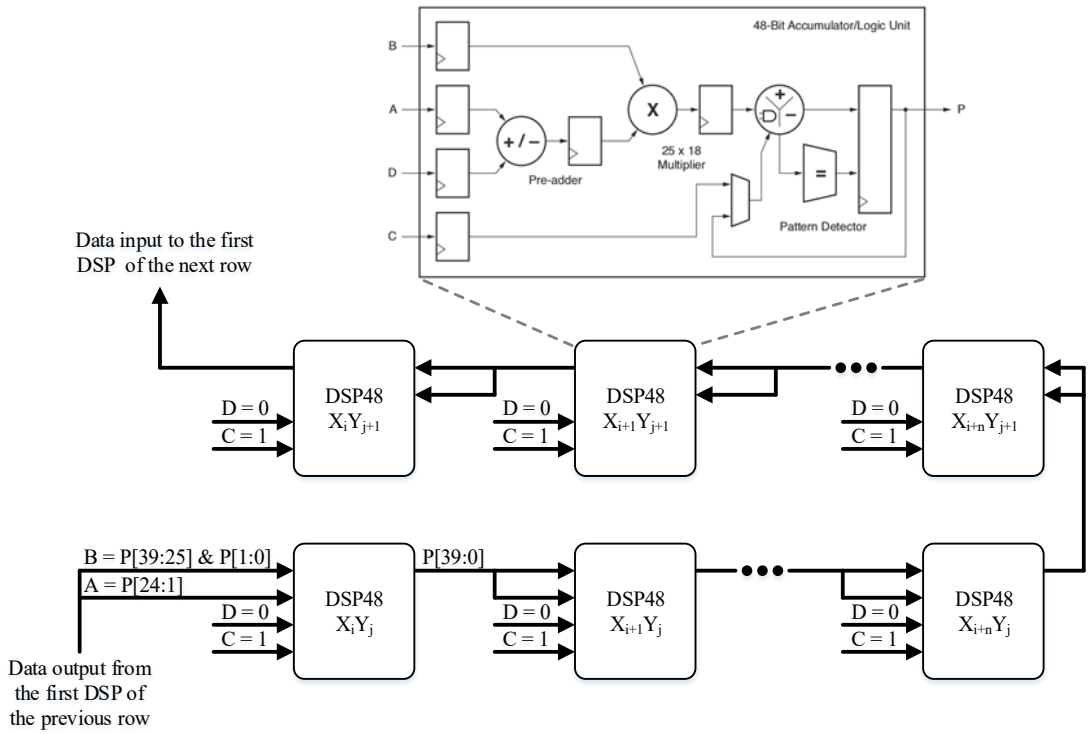


Figure 4.11: Horizontal DSP chain arrangement

The A input of the first DSP in the chain is set to a predefined checkerboard pattern into the IOB, i.e., "0x555555", while the B input is set to a value equal to "0x2", so that a

clock transient will produce doubling the value of A input at the P output (i.e., "0xAAAAAA"). The DSPs are not clocked; the clock input is set to a weak pull-down into the IOB. Additionally, to use the DSP chain in dynamic tests, given the above interconnection between two consecutive DSPs in the chain, all DSPs are clocked performing the multiplication of 0x555555 by 0x2.

Table 4.4 presents details about the configuration bitstream of the circuit under test (CUT) for the XC7Z020 and XC7Z045 devices. Note that the masked bits correspond to dynamically changeable memory locations as FFs or user memory content, such as BRAM, DRAMs, and SRLs. The configuration bits comprise the largest percentage of the bitstream, more than 55%, while the BRAM content bits account for almost one fifth. The FFs and SLICEM LUTs (i.e., SRLs or DRAMs) represent the smallest percentage of the bitstream.

Bit Type	XC7Z020		XC7Z045	
	Bits	%	Bits	%
Configuration bits (non-masked)	18,031,484	55.74	71,017,108	66.65
CLB FF bits (masked)	106,400	0.33	437,200	0.41
CLB SRL/DRAM bits (masked)	1,113,600	3.44	4,505,600	4.23
Other unknown bits (masked)	146,980	0.46	661,116	0.62
BRAM content bits (masked)	5,791,744	17.90	20,090,880	18.85
Unused masked bits (e.g., PS area)	7,158,880	22.13	9,843,904	9.24
Total bitstream bits	32,349,088	100.00	106,555,808	100.00

Table 4.4: Circuit under test – bit type details

Bit Type	XC7Z020		XC7Z045	
	Bits	%	Bits	%
Configuration bits (non-masked) zeros	15,419,274	85.51	60,798,772	85.61
Configuration bits (non-masked) ones	2,612,210	14.49	10,218,336	14.39
CLB FF bits (masked) zeros	53,200	50.00	218,600	50.00
CLB FF bits (masked) ones	53,200	50.00	218,600	50.00
CLB SRL/DRAM bits (masked) zeros	556,800	50.00	2,252,800	50.00
CLB SRL/DRAM bits (masked) ones	556,800	50.00	2,252,800	50.00
BRAM content bits (masked) zeros	3,661,824	63.22	11,489,280	57.19
BRAM content bits (masked) ones	2,129,920	36.78	8,601,600	42.81

Table 4.5: Circuit under test – initial bit values

To analyze the SEU effects regarding their initial values, it is required to know the percentage of the memory cells preloaded with '1' and '0' for the different memory categories (Table 4.5). This relative percentage helps to calculate the likelihood of a bit flip from zero to one (0→1) or from one to zero (1→0). In the case of configuration bits, the vast majority of them are preloaded with the value '0'. In the case of BRAM content bits, around two fifths are initialized to '1', while in the case of FFs and SLICEM LUTs the initialization values are balanced.

Table 4.6 presents the essential bits, which indicate exactly which bits in the bitstream are associated with the user design circuitry. From the values below, it seems that almost half of the device is occupied by the CUT in terms of configuration bits, while all the FFs are essential.

Bit Type	XC7Z020		XC7Z045	
	Bits	%	Bits	%
Configuration essential bits	7.850.897	43.54	31,295,881	44.07
Configuration non-essential bits	10.180.587	56.46	39,721,227	55.93
CLB FF essential bits	106.400	100.00	437,200	100.00
CLB FF non-essential bits	0	0.00	0	0.00

Table 4.6: Circuit under test – essential bits

4.1.4 Test Flow

The radiation test flow of the three experiments is shown in Figure 4.12. The first test flow (left figure) was used in CERN and the second test flow (right figure) was used in both GSI and PSI. The main difference between the two flows is the execution of the readback tasks. The CERN test flow includes both the Readback and Readback Capture, while the GSI/PSI flow only the Readback Capture. The initial values of all the CLB and IOB registers, which are stored in the configuration memory cells, are read during the Readback function. On the other hand, during the Readback Capture, the current state of all the CLB and IOB registers is fetched, storing all register values in the same configuration memory cells on which the corresponding initial values have been programmed. Thus, using both Readback functions, the upsets in both the CLB/IOB registers and the corresponding memory cells of the CRAM can be captured. Note that after a Readback Capture, an FPGA reconfiguration must be performed before the next simple Readback run, otherwise a mismatch in the CRAM bits dedicated to the registers will be observed. In addition, the GLUTMASK configuration bit [5] of the Control Register 0 is set to '1' so that the current state of the LUTRAMs is stored into the

configuration memory cells. Since the CUT is clock-free, any change in a LUTRAM due to irradiation will be reflected in the readback value.

Another difference between the CERN and GSI/PSI beams, which affects the test flow, is the length of the beam idle period; in CERN, it is large enough (e.g., ~30 sec) to allow running FPGA memory readback and reconfiguration steps while the beam is off. No configuration action is performed while the beam is active to avoid the generation of extra errors due to failures in the configuration logic. The host application waits for the beam deactivation to start the readback process and completes all configuration tasks before the next beam activation. Thus, when the host PC is triggered on the beam-off event, it first reads back the FPGA configuration memory through JTAG and records the readback data and then, performs readback-capture. These readback data contain the upsets that occurred in the last spill.

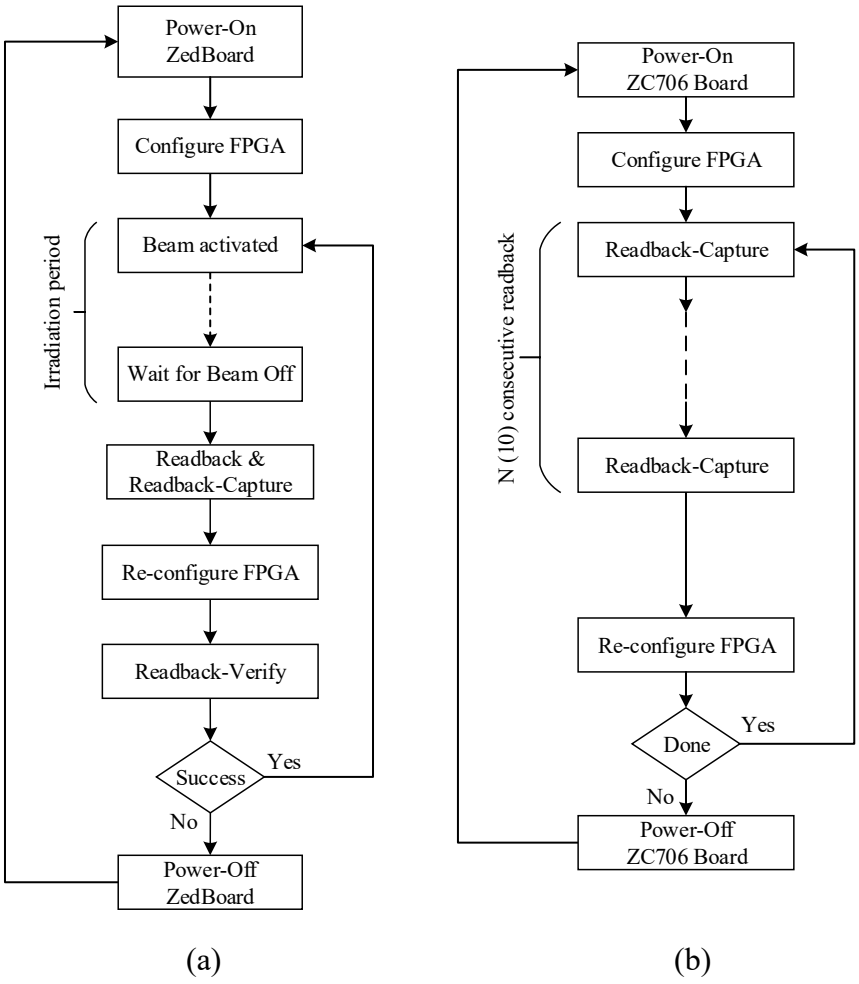


Figure 4.12: CERN test flow and GSI/PSI test flow

On the contrary, in GSI and PSI, all readback/reconfiguration tasks are performed at any time, i.e. regardless if the beam is off or on, since the time distance between spills is too short (1 to 2 sec) to fit in these tasks. Thus, the host PC performs the readback process N consecutive times (i.e., 10 times) and records the readback data before it runs reconfiguration. The readback process includes the readback-capture and the read of all the configuration registers [5], such as the Frame Address Register, Control Register, Status Register, etc. Only readback-capture is performed here in order to save time, because this way, the FPGA does not need to be reconfigured after each readback-capture cycle, which is the case in the CERN test flow as explained above. Readback-capture was chosen instead of simple readback in order to observe the SEEs in the registers and not in the corresponding bits of the CRAM (i.e., initial register values), since the latter occupies only 0.6% of the CRAM and does not largely affect its cross section.

Given that the FPGA configuration lasts 5 sec and the readback capture 13.5 sec, the whole test cycle lasts approximately 140 sec. Assuming a maximum proton SEU CRAM cross section of $7x10^{-16}$ cm^2/bit, as reported by previous experiments [136], and given the maximum beam flux of the facility (1.2×10^8 p/cm^2/sec) and the total number of device configuration bits (71,017,108) as shown in Table 4.1 and Table 4.4 respectively, the device was expected to experience six upsets per second, according to the Eq. 4.1, or otherwise 840 upsets during the test cycle period.

$$\text{upsets per second} = \text{flux} * \text{cross section} * \text{total bits} \qquad \text{[Eq. 4.1]}$$

A larger period would result in more accumulated upsets in the CRAM increasing the likelihood of a test failure, while for a shorter period the reconfiguration would be executed more frequently overwriting upsets in the configuration memory. Thus, the number of ten consecutive readbacks, which determines the period of the test cycle, was chosen as a good compromise between the accumulated and the overwritten upsets.

Since the readback-capture of the XC7Z045 lasts 13.5 sec and the beam period is, for example, about 3 sec (1 sec on, 2 sec off) in GSI, the beam is activated 4 to 5 times during one readback-capture cycle. This means that the recorded data from the repetitive readback cycles contain accumulated upsets, i.e., upsets that occurred in all the previous beam spills after the last reconfiguration. Thus, the post processing of the readback files requires an extra effort to remove the duplicated upsets.

The following steps are performed:

- *Power-on*: The target board is powered-on.
- *Initial configuration*: The target board is configured through the JTAG interface.
- *Readback*: In CERN, the host PC waits for the beam-off event to start the readback process, while in GSI/PSI, the host PC performs N consecutive readbacks without checking whether the beam is on or off.
- *Reconfiguration:* The FPGA is re-configured to remove upsets from the previous

irradiation periods. In CERN, a readback-verify command is performed to check for uncorrected upsets or JTAG malfunction (e.g., SEFI), while in GSI, the Configuration Registers are read and their state is checked. In case of success, the test goes to *Readback* step, otherwise a power cycle is executed and the test flow starts over.

4.1.5 Test Sessions

The minimum proton and heavy ion fluence required for radiation testing is primarily determined by the objectives of the test and the sensitivity of the device under evaluation. For SEE testing, an ion fluence of 10^7 particles/cm^2 or 100 observed errors has proved adequate to ensure statistically meaningful data, especially for devices with moderate to high cross sections [81]. In the case of protons, testing is often stopped at a fluence of 10^{10} protons/cm^2 (or 100 errors) [90]. Ultimately, the minimum fluence must strike a balance between achieving statistical confidence in the observed effects and avoiding excessive total dose that could mask or compound the observed SEE behavior. Consequently, in the case of heavy ions, the fluence per test session reached around 10^5 ions/cm^2, given that the observed errors exceeded 100 by far, while the proton fluence reached 10^{10} protons/cm^2.

Test Session	Angle (θ) of incidence [degrees]	Total fluence [ions/cm^2]	Effective LET [MeVcm^2/mg]
GSI	0	3.58 x 10^5	2.29
CERN 0	0	7.23 x 10^5	8.80
CERN 45	45	0.86 x 10^5	12.45

Table 4.7: Heavy ion test sessions

As mentioned in section 1.1, LET is the amount of energy lost by the incident ion along its path in the absorber medium when colliding with atomic electrons, i.e., the unit of reference for SEE irradiation with heavy ions. On the other hand, energy is the unit of reference for SEE irradiation with high energy protons [48].

The CERN radiation test was carried out at a LET of 8.8 MeVcm^2/mg and for two different angles of incidence (θ), 0^o and 45^o, obtaining two effective LETs of 8.8 MeVcm^2/mg and 12.45 MeVcm^2/mg, respectively. Effective LET is the equivalent LET obtained by tilting the device under test with respect to the beam axis, hence increasing the path length of the ion and the total energy deposited and is calculated as LET/cos(θ) [48]. The total fluence for the two thetas, 0^o and 45^o, were 7.23x10^5 ions/cm^2 and 0.86x10^5 ions/cm^2, respectively. The GSI radiation test was performed at 2.29 MeVcm^2/mg for 0^o angle of incidence and with a total fluence of 3.58x10^5 ions/cm^2. The heavy-ion test sessions are summarized in Table 4.7. In all the above test sessions, only

the CUT with SRLs, described in subsection 4.1.3, was applied; thus, the CRAM, BRAM, FFs and SRLs memory categories were analyzed.

In PSI, the energy and flux were determined separately for each test session, while its duration was configured to reach at least a fluence of 10^{10} p/cm^2. Two hardware benchmarks, PL1 and PL2, were applied for the proton radiation tests. In the first CUT, the SLICEM LUTs were configured as SRLs while in the second CUT as DRAMs. The results from the execution of both benchmarks have been taken into account for the reliability analysis of the CRAM, BRAM and FFs, while the results of the two different CUTs for the analysis of the LUT SRL and LUT DRAM, respectively. Note that two different ZC706 boards were used for the tests to avoid board damage due to TID. During the experiments, the first board received a TID of 117 krad and the second board received 58 krad. The proton test sessions are summarized in Table 4.8. The tilt angle of the device under test with respect to the beam axis is equal to 0 degrees for all the proton tests.

Test Session / CUT	**Energy [MeV]**	**Duration [secs]**	**Flux [p/cm^2/sec]**	**Total fluence [p/cm^2]**
PSI 30 / PL1	30	263	3.803 x 10^7	1.000 x 10^{10}
PSI 30 / PL2	30	237	4.230 x 10^7	1.003 x 10^{10}
PSI 50 / PL1	50	529	3.784 x 10^7	1.963 x 10^{10}
PSI 50 / PL2	50	528	3.793 x 10^7	1.964 x 10^{10}
PSI 100 / PL1	100	546	3.667 x 10^7	1.963 x 10^{10}
PSI 100 / PL2	100	526	3.809 x 10^7	1.965 x 10^{10}
PSI 150 / PL1	150	302	3.321 x 10^7	1.003 x 10^{10}
PSI 150 / PL2	150	234	4.274 x 10^7	1.000 x 10^{10}
PSI 200 / PL1	200	1087	2.422 x 10^7 to 1.947 x 10^8	4.946 x 10^{10}
PSI 200 / PL2	200	1029	4.865 x 10^7	4.929 x 10^{10}

Table 4.8: Proton test sessions

4.2 Reliability Analysis

For the reliability analysis, a post-processing database was created including the upsets recorded during all the experiments. The upsets were extracted by comparing the recorded readback and readback-capture data with the golden readback and readback-capture data, respectively. The golden data is retrieved from the device by performing readback or readback-capture during beam-off. For each bit upset, a database entry was created with fields (Table 4.9), such as the readback file timestamp, the golden bit value, the frame address, the bit position in the frame, whether the bit is masked (based on the

AMD Xilinx .msd file) or essential (based on the AMD Xilinx .ebd file), whether the bit is located into a masked frame (i.e., all bits of the frame are masked) or a non-essential frame (i.e., none of the bits of the frame is essential) and the corresponding circuit logic (based on the AMD Xilinx .ll file) in case the bit is masked.

It must me noted that the .msd, .ebd and .ll files are generated by the AMD Xilinx tools. The .msd file is an ASCII file that contains mask information, i.e., the bits that are associated with the storage elements of the user design, such as FFs, SRLs, LUTRAM and BRAM. The .ebd file is an ASCII file that indicates whether a bit is essential or not. Finally, the .ll file includes information on each of the memory cells in the design that can be captured for readback, i.e. FFs, SRLs, LUTRAM and BRAM. This file contains the absolute bit position in the readback stream, frame address, frame offset, logic resource used (i.e., SLICE_XnYm or RAMB36_XnYm) and specific logic in the design (i.e., the latch or memory bit associated with this memory cell). The database fields and their description are summarized in Table 4.9.

Database fields	Description
timestamp	The timestamp of the readback or readback-capture file
readback capture	False: readback True: readback-capture
golden bit value	Bit value of the golden file, e.g. if this bit is 0 then the bit is flipped from 0 to 1, otherwise the bit is flipped from 1 to 0
frame address	32-bit hex value of the frame address where the erroneous bit is located
block type	The value of the frame address bits [25:23] (see Table 2.2)
top/bottom	The value of the frame address bit [22] (see Table 2.2)
row address	The value of the frame address bits [21:17] (see Table 2.2)
column address	The value of the frame address bits [16:7] (see Table 2.2)
minor address	The value of the frame address bits [6:0] (see Table 2.2)
word of frame	The word position in the frame of the erroneous bit
bit of word	The bit position in the word
bit of frame	The bit position in the frame
masked bit	0: The bit is non-masked 1: The bit is masked
masked frame	False: The bit is located into a non-masked frame (at least one bit of the frame is non-masked) True: The bit is located into a masked frame, i.e., all the bits of the frame are masked. Such frames are the BRAM content frames and the PS area frames

Database fields	Description
essential bit	0: The bit is non-essential 1: The bit is essential
non-essential frame	False: The bit is located into a frame which has at least one essential bit True: The bit is located into a non-essential frame, i.e., all the bits of the frame are not essential
logic block	In case of masked bit, this field identifies the specific logic block position, i.e., SLICE_XnYm or RAMB36_XnYm
specific logic	In case of masked bit, this field identifies the specific logic of the logic block, i.e., the specific FF (Latch=D5FF.Q) and LUTRAM or BRAM bit (RAM=A:1)

Table 4.9: Post-processing database fields

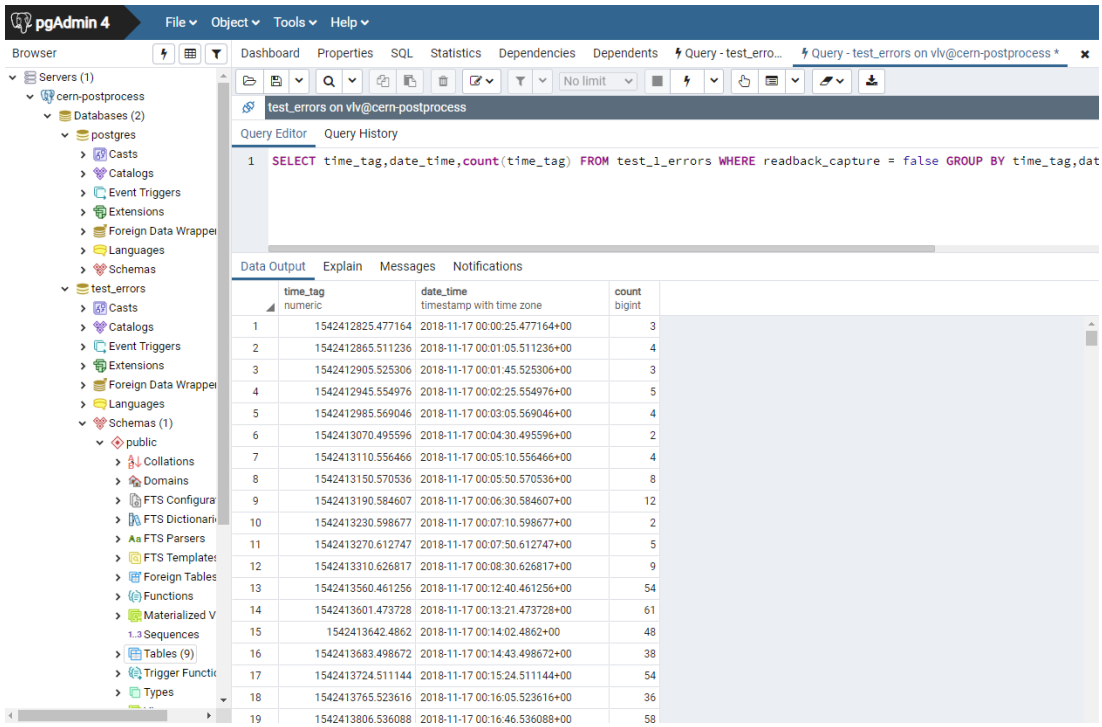


Figure 4.13: pgAdmin database tool

All the reliability analysis results have been extracted from the database using queries. In order to facilitate access and management of the data remotely and from multiple points, it was chosen to implement the database using a PostgreSQL web-based tool such as pgAdmin [115]. The pgAdmin package is a free and open-source GUI administration tool for PostgreSQL [117], which is supported on many computer platforms. PostgreSQL also known as Postgres, is a free and open-source relational database management system

(RDBMS) emphasizing extensibility and structured query language (SQL) compliance. PostgreSQL is supported on all major operating systems, including Linux and Windows, and handles a range of workloads from single machines to data warehouses or web services with many concurrent users. Figure 4.13 depicts a screenshot of a database query for the CERN test.

To analyze the SEUs, the effects in four different memory categories, i.e., CRAM, FFs, BRAM and LUTRAM were studied. In the CERN experiment, the CRAM category includes all the non-masked bits of the configuration bitstream (i.e., the interconnect and block configuration frames for the CLB, DSP, IOB, CLK and configuration parameters of the BRAM) plus the masked bits of the CLB FFs (i.e., the initial FF values) obtained by the analysis of the readback data. In the GSI and PSI experiments, the CRAM category includes only the non-masked bits of the configuration bitstream as obtained by the readback-capture data. Since CRAM bits represent static memory cells, i.e., they do not alter during the circuit execution, any upset that occurs during the DUT operation is supposed to be a SEU.

On the other hand, the BRAM, FF and LUTRAM categories include the masked bits of the configuration bitstream for the current values of the BRAMs, FFs and SRLs or DRAMs, respectively. These bits are assumed as dynamic memory cells and have been obtained by the analysis of the readback-capture data. Given that the DUT clock is paused, any difference in these bits compared to their initial values can be caused by either a SEU in these memory cells or a SET on the clock or the control signals, such as the reset/preset signals of the corresponding DUT chains. To calculate the SEU cross section for these memories, the phenomena where a single event causes several memory cells to upset are first identified; for example, SET-induced events, e.g., due to SET in clock and reset/preset signals, or single-event resets, e.g., events corrupting entire memory clusters. Such phenomena were observed in our experiments and are described in the following subsections and they were excluded from the cross section calculation of the BRAM, FF, SRL and DRAM categories.

4.2.1 Heavy Ions

4.2.1.1 SET-induced Events and Single-event Resets

In the CERN experiments, we observed several events where a SET affects a number of FFs or SRLs. Table 4.10 shows the number of occurrences of various SET-induced events in FFs and SRLs. One case observed several times in CERN was that all FFs of one SLICE, i.e., 8, 16 or 24 FFs are flipped extending in one, two or three SLICEs located in one or two CLBs. Another case observed was that a large number of SRL LUTM bits, i.e., 8, 16, 128, 200 and 256 bits, located in the four LUTMs of the same SLICE are affected. Note that each SLICEM consists of four 64-bit LUTMs and a 32-bit SRL occupies a 64-bit LUTM, since the SRL uses a master-slave arrangement of the 64 bits

for a 32-bit shift register. Although a SET in the clock net could affect all the memory cells of an SRL, the number of SRL bit upsets depends on the preloaded values of the registers; for example, in a register preloaded with all 0s, only the first bit is flipped, while in a register preloaded with a 0-1 pattern, all the 64 bits may present an upset. Another case that was observed three times in CERN was that a large number of FFs (i.e., 400 and 800 bits) and SRL bits (i.e., 200, 3200 and 6400) extending in several SLICEs that belong to adjacent physical half-row columns are affected. This observation is in accordance with the results presented in [93] regarding the transients in FF reset signals. However, in the experiments of the current thesis, we observed that all these events are caused by transients in the clock tree since the data of the chain after the event are the preloaded patterns shifted by one bit position. The analysis of the above cases and the clock tree structure, lead to the conclusion that a SET may hit a clock branch that drives one SLICE, one CLB or two adjacent half-row columns, affecting all the corresponding FFs and SRLs.

Memory category	**Event description**	**LET [MeVcm^2/mg] / Fluence [ions/cm^2]**		
		2.29 / 3.58x10^5	**8.80 / 7.23x10^5**	**12.45 / 0.86x10^5**
FF	all FFs of one SLICE	-	113	18
	all FFs of one ½-column	-	1	-
	all FFs of two adjacent ½-column	-	2	-
	4 of the 8 FFs of one SLICE	-	74	-
	4 of the 8 FFs of one ½-column	1	-	-
LUT SRL	all SRLs of one SLICE	-	26	4
	all SRLs of one ½-column	-	2	-
	all SRLs of two adjacent ½-column	-	1	-
BRAM	1 Kbits upset	-	2	-
	2 Kbits upset	-	3	-
	3 Kbits upset	-	6	2

Table 4.10: Heavy-ion SET-induced events and single-event resets

Another phenomenon that we observed in both the CERN and GSI experiments is that the 4 out of 8 FFs of a SLICE are affected. In CERN, these FFs are preloaded with either 0s or 1s. Since, in these tests, the FFs are configured with synchronous reset and their clock input is disabled, transients in the reset signal can not cause these upsets. In the GSI tests, these FFs are the same 4 FFs of all the 25 SLICEs of a half-row SLICE column, all preloaded with 1s. All the affected FFs are initially configured with asynchronous

preset and change to 0s, so a SET cannot cause this phenomenon on the preset input. Furthermore, neither the output of the 4 SLICEM SRLs nor the other 4 FFs of each SLICE, which are preloaded with a value opposite to the value of the above FFs and feed the above FFs, are flipped. Consequently, this phenomenon is not a SET on the clock signal, but it seems to be a single-event reset which extends down and affects a number of FFs located in the same SLICE or physical column.

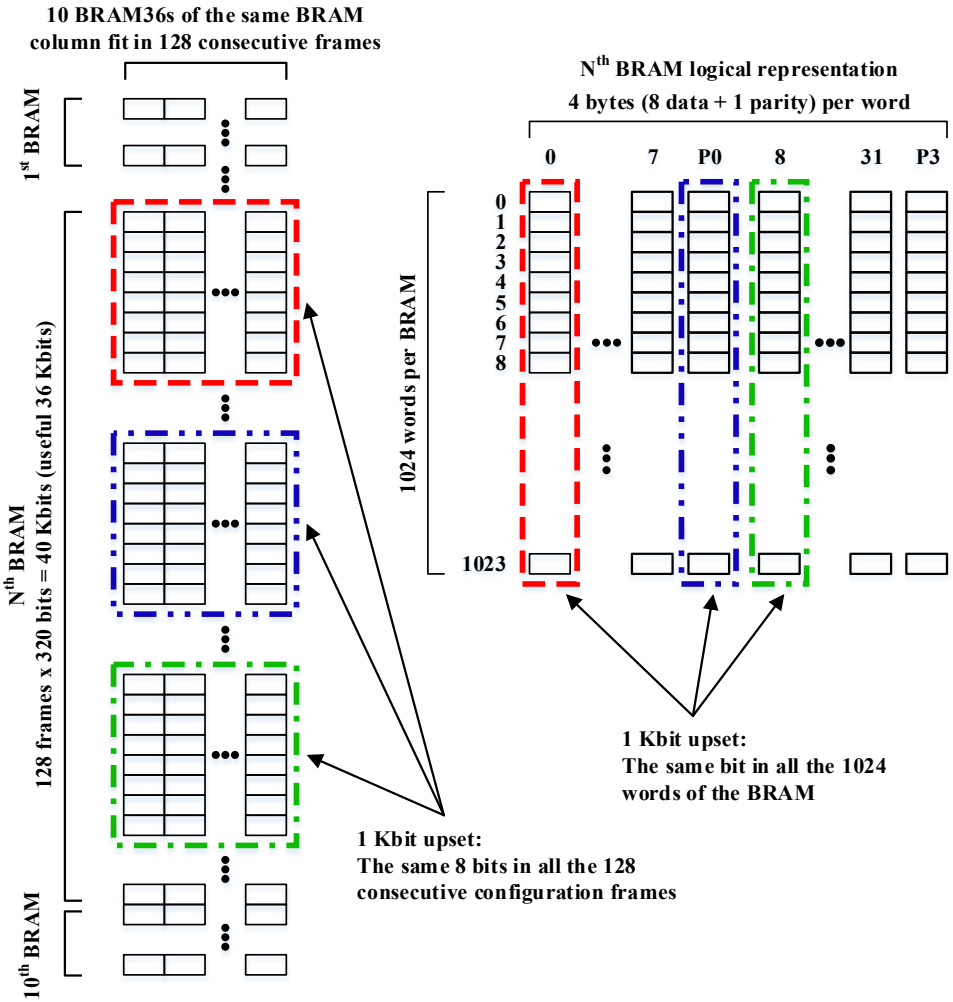


Figure 4.14: Single-event reset in BRAM

We observed another phenomenon where 1024, 2048 or 3072 bits of the same BRAM are affected by a single event. In all these cases, the BRAM cells are preloaded with 1s and change to 0s, while the corresponding parity bits flip from 0 to 1. Figure 4.14 depicts the case of 3072 (3 Kbit) BRAM upsets. The left part of the figure shows the structure of a BRAM column: 128 consecutive configuration frames of BRAM type contain the data

of ten 36Kb BRAMs. Each BRAM consists of 36,864 bits (32,768 data bits + 4,096 parity bits) and occupies the same 320 bits of all the 128 frames. The logical representation of the BRAMs, organized in 36-bit words (4 bytes + 1 parity bit per byte), is shown on the right part of the figure. The 1 Kbit BRAM upset phenomenon (highlighted by the red dashed line in Figure 4.14) affects 8 specific bits of all the 128 frames. These bits correspond to BRAM locations 0, 32, 64, …, 1023 * 32, i.e., the same bit in all the 1024 words. In the cases of 2 Kbit (the red dashed and the blue dash-double dotted boxes in Figure 4.14) and 3 Kbit (all the highlighted boxes in Figure 4.14) BRAM upset, the phenomenon is similar. Two or three non-adjacent blocks of 8 x 128 bits are affected. In all cases, the bits belong to the same BRAM and are either a sequence of BRAM data bits with a distance of 32 bits or a sequence of BRAM parity bits with a distance of 4 bits. This phenomenon cannot be a multiple-bit upset or caused by a SET on the clock signal since the BRAM write enable is disabled. It is similar to the one presented in [134], dubbed as a BRAM bit position reset single event, where a control SET resets exactly 1 Kbits in a particular block and [91] as a reset-like event that affects BRAM clusters of approximately 1 Kbit.

All these effects appear to increase significantly when the beam energy increases from very-high (GSI) to ultra-high (CERN), as shown in Table 4.10. In addition, they are reduced when the DUT is tilted at an angle of 45 degrees. The latter probably occurs because, although the DUT tilt increases the path length of the ion and the total LET deposited, it also causes more energy loss and scattering.

4.2.1.2 SEFIs

In general, SEFIs can occur due to upsets either in the reconfigurable user logic or in the built-in hardwired logic, such as the JTAG test access port (TAP) controller or configuration circuitry. In the former case, SEFIs affect the operation of the circuit, while in the latter, the operation of the configuration functions, e.g., a SEFI in the readback-capture control logic can lead to the observation of an extensive failure in the state of registers. Since the CUT is clock free, SEFIs in the reconfigurable logic are not expected to occur. In both experiments, SEFIs were observed only in the configuration functions, such as readback, readback-capture, reconfiguration, readback-verify, and were probably due to upsets in the built-in hardwired logic. These events fall into both types of SEFIs, i.e., reset by software or device power cycling.

In the CERN campaign, the reset-by-software SEFI occurred once after a readback-verify command, which failed while it was checking for uncorrected upsets in the configuration memory. In contrast, the power cycling SEFI occurred three times. Two of these events also occurred after a readback-verify step and the third during a readback action.

In the GSI campaign, a possible reset-by-software SEFI event was observed. The same bits in a CRAM frame were affected in two consecutive of the ten readback commands. One readback recorded 16 bits in a frame having erroneous values compared with the

golden bitstream. The next readback recorded the same 16 bits with the correct values without performing reconfiguration between these two readbacks, which would have corrected these bits. These 16 bits, which are flipped from 0 to 1, along with 2 other bits, which are '1' and are not flipped, form a sequence of 18 bits having a distance of 8 bits in 5 consecutive words of the frame. This erratic behaviour could occur due to the erroneous operation of some specific flip-flops of the JTAG data/control register of the TAP controller. This type of SEFI was also observed 8 times as 1-bit instead of 16-bit flip, recording two successive readbacks.

4.2.1.3 SEUs and Cross Section

To calculate the SEUs for all the different memory categories and the corresponding cross sections, we removed all the phenomena described above from the total upsets. Table 4.11 and Table 4.12 present the total upsets, the SEUs and SEU cross sections, accordingly. It must be noted that the CRAM and BRAM cross sections in the CERN experiments are approximately 2 to 3 times higher compared to the cross section observed in previous works [22], [135], [136] with standard energy heavy ion irradiation of similar LET values. This difference can be justified due to the higher frequency of occurrence of MCUs caused by the ultra-high energy heavy ions [3], as analyzed in the following subsection, and the contribution of secondary particles, which can be introduced by intercepting elements and contaminate the beam purity, to the final SEU measurement [89].

LET [MeVcm^2/mg]	**Fluence [ions/cm^2]**	**Total upsets & SEUs**	**CRAM**	**CRAM essential**	**BRAM**	**FF**	**LUT SRL**
2.29	3.58x10^5	Total upsets	5921	2935	4087	116	1268
		SEUs	5921	2935	4087	16	1268
8.80	7.23x10^5	Total upsets	27805	14057	39142	3743	16296
		SEUs	27805	14057	12518	542	4091
12.45	0.86x10^5	Total upsets	3351	1822	7278	258	1094
		SEUs	3351	1822	1134	114	507

Table 4.11: Heavy-ion total upsets and SEUs

The CRAM essential bit cross section was also calculated taking into account only the upsets in the essential configuration bits. Observe that the CRAM essential bit cross section is always higher than the CRAM cross section. This can be explained by the fact that the percentage of the essential memory cells preloaded with '1' is greater than that of total memory cells preloaded with '1' and that the likelihood a CRAM cell to flip from 1 to 0 has been measured to be higher than the likelihood to flip from 0 to 1, as presented

below. The rationale behind the calculation of this cross section is that it provides a more realistic probability metric of SEUs affecting the DUT behaviour.

LET [MeVcm²/mg]	CRAM [cm²/bit]	CRAM essential [cm²/bit]	BRAM [cm²/bit]	LUT SRL [cm²/bit]	FF [cm²/bit]
2.29	$2.49\text{x}10^{-10}$	$2.79\text{x}10^{-10}$	$6.05\text{x}10^{-10}$	$8.37\text{x}10^{-10}$	$1.09\text{x}10^{-10}$
8.80	$2.12\text{x}10^{-9}$	$2.44\text{x}10^{-9}$	$2.99\text{x}10^{-9}$	$5.08\text{x}10^{-9}$	$7.04\text{x}10^{-9}$
12.45	$2.16\text{x}10^{-9}$	$2.67\text{x}10^{-9}$	$2.28\text{x}10^{-9}$	$5.31\text{x}10^{-9}$	$12.50\text{x}10^{-9}$

Table 4.12: Heavy-ion SEU cross section

The cross section of the different memories for the three LETs along with their error bars are illustrated in Figure 4.15. For the calculation of the error bars, a Poisson distribution of the SEUs, a confidence level 95% and an uncertainty on the measured fluence 10% were considered [48], [80].

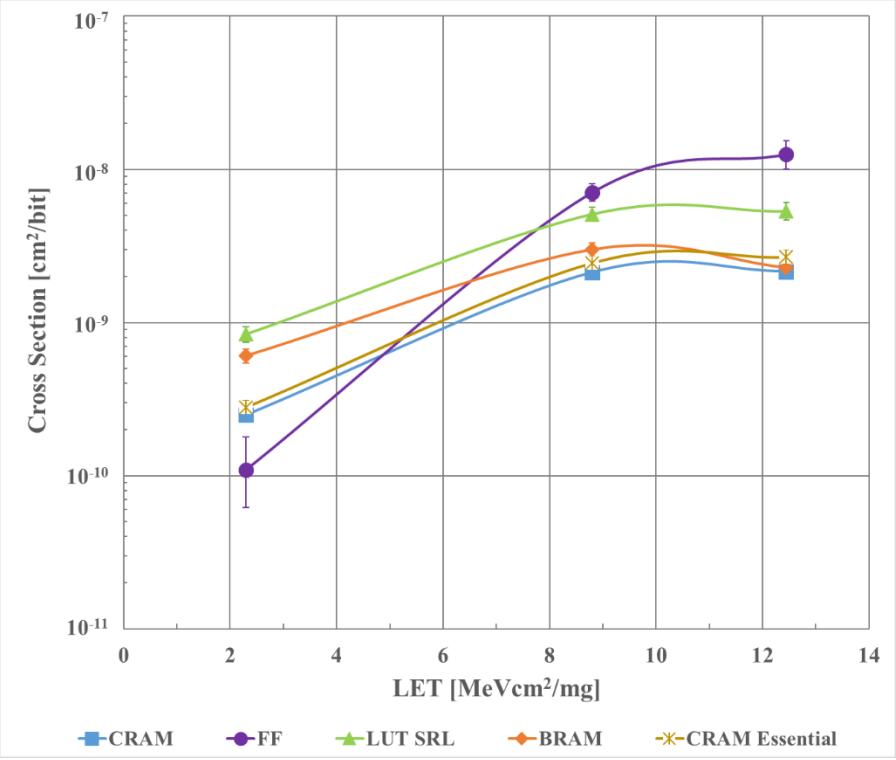


Figure 4.15: Heavy-ion SEU cross section vs LET

Analyzing the SEU effects regarding their initial values, the likelihood to occur upset in memory cells preloaded with '0' relative to those preloaded with '1' was calculated for

the different memory categories. As shown in Table 4.13, the likelihood to occur upset in memory cells preloaded with '1' is generally greater than or approximately equal with those preloaded with '0'. In the cases of CRAM, BRAM and FFs, the likelihood ratio 0→1 : 1→0 slightly decreases as LET increases. Only in the case of FFs at the CERN experiment with a 45-degree tilt, a high decrease is observed, but it may be a statistical error since the number of FFs is significantly smaller compared to the other memory structures and the total fluence of this test is lower. In addition, in the case of SRLs it is independent of the LET and approximately equal to 1.

Memory category	**LET [MeVcm^2/mg]**		
	2.29	**8.80**	**12.45**
CRAM	1 : 1.25	1 : 1.32	1 : 1.65
BRAM	1 : 0.91	1 : 1.04	1 : 1.16
FF	1 : 1.33	1 : 1.80	1 : 4.49
LUT SRL	1 : 0.97	1 : 0.94	1 : 0.95

Table 4.13: Heavy-ion likelihood to upset 0→1 : 1→0

4.2.1.4 MBU/MCU Analysis

In this subsection, the MBUs and the MCUs observed in the various memory categories, are analyzed. This analysis may be beneficial for the design of effective SEU mitigation techniques. For example, the MBUs in the CRAM cannot always be corrected by the embedded ECC of the Zynq-7000 FPGAs and require a more sophisticated scrubbing scheme able to correct multiple errors.

Number of upsets per CRAM frame	**LET [MeVcm^2/mg] / Fluence [ions/cm^2]**		
	2.29 / 3.58x10^5	**8.80 / 7.23x10^5**	**12.45 / 0.86x10^5**
1	5790	26072	2845
2	64 (2)	574 (482)	194 (175)
3	1 (0)	139 (135)	28 (26)
4	-	42 (39)	6 (6)
5	-	-	2 (2)

Table 4.14: Heavy-ion upsets per CRAM frame

Table 4.14 presents the number of occurrences of single and multiple-bit upsets observed within a single CRAM frame in one readback file, while the number in parenthesis shows

the occurrences of adjacent bit upsets. Assuming that when upsets in two or more neighboring bits are observed in a frame, we can safely categorize them as MBUs, the numbers in parentheses indicate the MBUs while the remaining are upsets probably generated by multiple accumulated events. Although most cases are SBUs, there is a significant number of multiple upsets per frame. In the CERN experiment, more than 84% of 2-bit upsets and 92% of three or more upsets per frame are MBUs. On the contrary, the vast majority of GSI multiple upsets are probably generated by multiple events, given that only two 2-bit upsets are MBUs. The large number of accumulated SBUs in the GSI experiment can be explained by the fact that the beam was activated approximately 5 times during one readback-capture period.

Several MCUs expanding in more than one configuration frames were observed. To identify the MCUs and separate them from multiple upsets caused by different particles, the approach proposed in [155] was adopted. This method determines statistically whether a multiple upset is the result of a single event, which means MCU or MBU, or the result of multiple events, which means multiple SBUs. The method was based in a physical adjacency model. Table 4.15 presents the patterns (shapes) of the MCUs and their percentage frequency of occurrence for the three different LETs. The x-axis of the shapes represents consecutive frame address, while the y-axis consecutive bits in a frame.

LET [MeVcm²/mg]	□											
2.29	97.8	1.45	0.29	0.41	-	-	-	-	-	-	0.02	-
8.80	51.2	29.8	14.0	2.28	0.49	0.42	0.30	0.28	0.24	0.13	0.13	0.12
12.45	40.0	32.7	17.2	2.19	0.21	1.12	0.27	0.85	0.75	1.50	0.80	0.27
												
2.29	-	-	-	-	-	-	-	-	-	-	-	-
8.80	0.11	0.10	0.07	0.06	0.05	0.04	0.02	0.02	0.02	0.02	0.01	0.01
12.45	0.21	0.48	-	-	-	-	-	0.32	0.16	-	-	0.16

Table 4.15: Heavy-ion percentage frequency of MCU occurrences in CRAM

From Table 4.15 and the analysis of results, the following are concluded:

- The MCU patterns extend up to two frames and up to four bits per frame. These shapes concur with the patterns observed in the neutron radiation tests of [49].
- The MCUs affect consecutive configuration frames with minor addresses starting from an even value, e.g. 0 & 1, 2 & 3, etc. and not 1 & 2, 3 & 4, etc., which indicates that the Zynq-7000 FPGA has an interleaving depth of two.

- The MCUs affect consecutive bits located in the same 32-bit word of a frame.
- The vast majority of upsets (i.e., more than 90%) are SBUs and 2-bit MBUs/MCUs regardless of the LET.
- The percentage of SBUs decreases as LET increases while the energy remains constant, which indicates that the MBUs and MCUs become more common at higher LETs.
- The percentage of MBUs/MCUs increases with the ultra-high energy heavy ions (CERN experiment) compared to the GSI experiment as expected [3].
- Most MBUs/MCUs occur only in the higher LETs. Specifically, the 2-bit upsets seem to increase sharply when moving from very-high (GSI) to ultra-high energy beam (CERN), which is in accordance with the observation of [39] that the ultra-high energy beam induces more and larger clusters of MCUs compared to the lower energies. Given the fact that the ultra-high energy can better mimic the impact of the GCR environment, it provides a more realistic scenario regarding the rates and patterns of MCUs.
- The upsets with more than 2 bits increase when tilting the board to large angles (CERN 45).

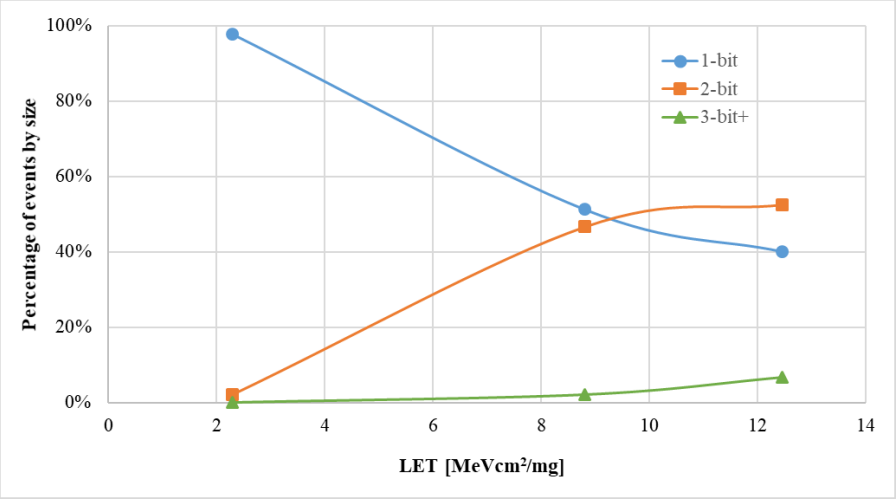


Figure 4.16: Heavy-ion percentage of MCUs by size as a function of LET

The percentage of events by size is plotted in Figure 4.16 as a function of LET. At low LETs, only a small percentage of events are MCUs and the vast majority are SBUs. As the LET increases, the percentage of SBUs decreases, the percentage of 2-bit MCUs significantly increases, while the percentage of MCUs with more than 2 bits slightly increases.

LET [MeVcm^2/mg]	Fluence [ions/cm^2]	Uncorrectable MBUs (by SEM IP)	Cross section [cm^2/bit]
2.29	3.58 x 10^5	0	-
8.80	7.23 x 10^5	127	0.97 x 10^{-11}
12.45	0.86 x 10^5	25	1.61 x 10^{-11}

Table 4.16: Heavy ions - uncorrectable (by SEM IP) MBUs

Note that only the non-masked bits of the configuration memory have been considered in the above analysis, i.e., that are not dynamically altered by the CUT. This means that possible upsets in these bits can be detected and corrected by a scrubbing mechanism. The highlighted patterns in Table 4.15 are the MBU patterns that the AMD Xilinx SEM IP core is not able to correct. It must be noted that the SEM IP in enhanced repair mode [12] supports the correction of single-bit or adjacent double-bit errors per frame, while it cannot correct MBUs with multiplicity higher than two. The number of these uncorrectable MBUs and their cross sections for different LETs are shown in Table 4.16. Notice that there are no uncorrectable MBUs at 2.29 MeVcm^2/mg while the cross section of uncorrectable MBUs increases as the LET increases and reaches 1.61 x 10^{11} cm^2/bit at 12.45 MeVcm^2/mg.

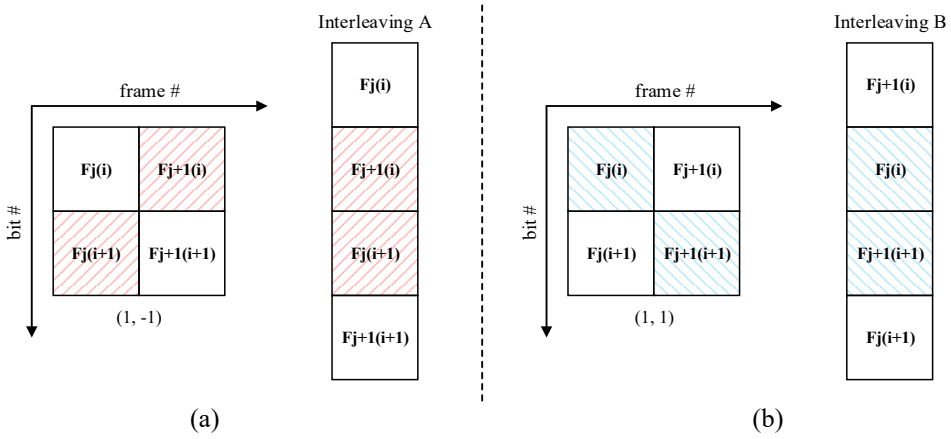


Figure 4.17: CRAM interleaving schemes

We further analyzed the MCUs attempting to justify the patterns observed based on the interleaving schemes that are possibly used in the Zynq-7000 FPGA configuration memory, i.e., configuration bits in sequential frames are physically adjacent to maximize the benefits of the error correction coding. The second and fourth shapes of the first row of Table 4.15 provide an indication of two different bit interleaving schemes. For

example, one could assume that the first MCU shape where the hamming distance of the two upsets is (1, -1), i.e., the right upset is in the i bit of the frame j+1 (Fj+1) and the left upset is in the i+1 bit of the frame j (Fj), has been caused because the bits of the frames Fj and Fj+1 have been interleaved in the following way (scheme A): Fj(i), Fj+1(i), Fj(i+1), Fj+1(i+1), etc., as shown in Figure 4.17 (a). On the other hand, in Figure 4.17 (b), the second MCU shape with hamming distance (1, 1) may provide an indication of a second interleaving (scheme B): Fj+1(i), Fj(i), Fj+1(i+1), Fj(i+1), etc. Following the above reasoning, it could be assumed that based on their orientation, the red and cyan (diagonal stripes) shapes of Table 4.15 belong to the interleaving schemes A and B, respectively. In the three orange shapes (grid), assuming that the consecutive bits of the same frame have been affected by a single event, one could observe that these frames have been arranged without interleaving. Finally, no interleaving scheme can be derived from the four MCU purple shapes (dotted cells). Specifically, the first two purple shapes could belong to either interleaving A or B, while the last two have probably occurred due to multiple events.

Given that the device consists of five configuration column types, i.e., IOB, CLK, CLB, BRAM, DSP, as described in subsection 2.3.2.2, the different interleaving schemes (Interleaving A & B, w/o-Interleaving) per configuration column type and per configuration frame addresses were analyzed. The results are shown in Table 4.17. Interleaving A appears in all column types, Interleaving B in the CLB & DSP columns and w/o-Interleaving in the CLB, BRAM & DSP columns. Analyzing further these results, we observed that Interleaving A appears in most CLB and BRAM frames and all DSP, CLK and IOB frames, Interleaving B appears only in CLB or DSP frames with minor addresses equal to 26, 27, 32 and 33 and w/o-interleaving appears in CLB, BRAM or DSP frames with minor addresses equal to 26 and 27. In the CLB and BRAM frames, Interleaving A does not appear along with Interleaving B or w/o-interleaving. From the above statistical analysis, one can deduce that the CRAM of the Zynq-7000 FPGA supports different bit interleaving schemes for specific configuration frame column types and addresses.

column type	**configuration frame minor addresses**		
	Interleaving A	**Interleaving B**	**w/o-Interleaving**
CLB	0-25, 28-31, 34-35	26-27, 32-33	26-27
BRAM	0-25	-	26-27
DSP	0-27	26-27	26-27
CLK	0-29	-	-
IOB	0-41	-	-

Table 4.17: Interleaving schemes per frame type

The same approach with CRAM was followed to identify the MBUs/MCUs for the FFs and SRLs. We observed only SBUs in the FFs and SBUs and 2-bit upsets in the SRLs; the 2-bit upsets were actually SBUs propagated in the SRL logic due to the master-slave arrangement of its latches. Figure 4.18 presents a 2-bit portion of a 32-bit SRL, which occupies a 64-bit LUTM in master-slave scheme. Given that the clock is set constantly to '0', an SEU on a slave latch (point 1 in the figure) is also propagated to the next always-open master latch (point 2 in the figure), causing a 2-bit upset.

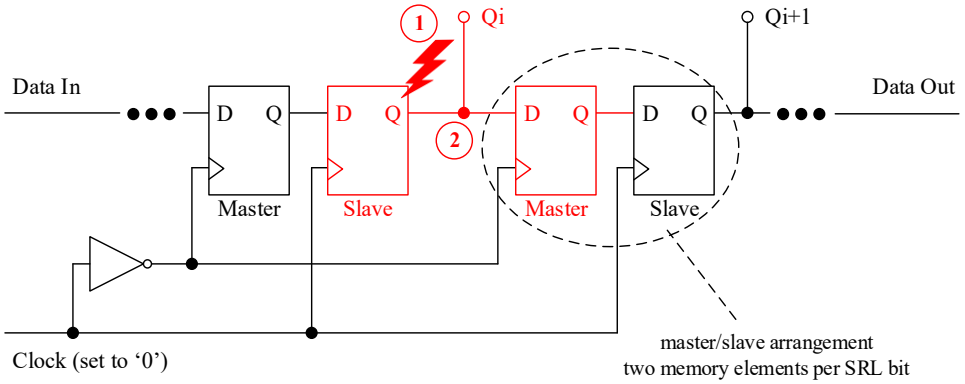


Figure 4.18: Two-bit upsets in LUT SRLs

The MCU patterns for the BRAM for the three different tests are shown in Table 4.18. We observed only SBUs in the GSI test with lower LET and SBUs and a considerable number of 2-bit MCUs in the CERN tests with ultra-high energy beam and higher LETs. Note that the MCUs occurred due to the phenomenon of 1K, 2K or 3K bits upset, previously described in subsection 4.2.1.1, are not considered here. The 2-bit MCU pattern indicates that two bits of a BRAM with hamming distance 256 have been affected, according to the BRAM layout defined in the logic location file generated by AMD Xilinx tools. For example, as shown in Figure 4.19, bits 0 and 256 of a BRAM correspond to the first bit of two consecutive frames. Additionally, the frequency of occurrence of MCUs increases with the LET, but in contrast to CRAM their multiplicity does not exceed 2. This can be explained by the fact that the FPGA architectures are composed of heterogeneous SRAM blocks (i.e., CRAM, BRAM) with different schemes and layouts [140].

	LET [MeVcm^2/mg]		
	2.29	**8.80**	**12.45**
	100.0	83.0	78.3
	-	16.0	21.5

Table 4.18: Heavy-ion percentage frequency of MCU occurrences in BRAM

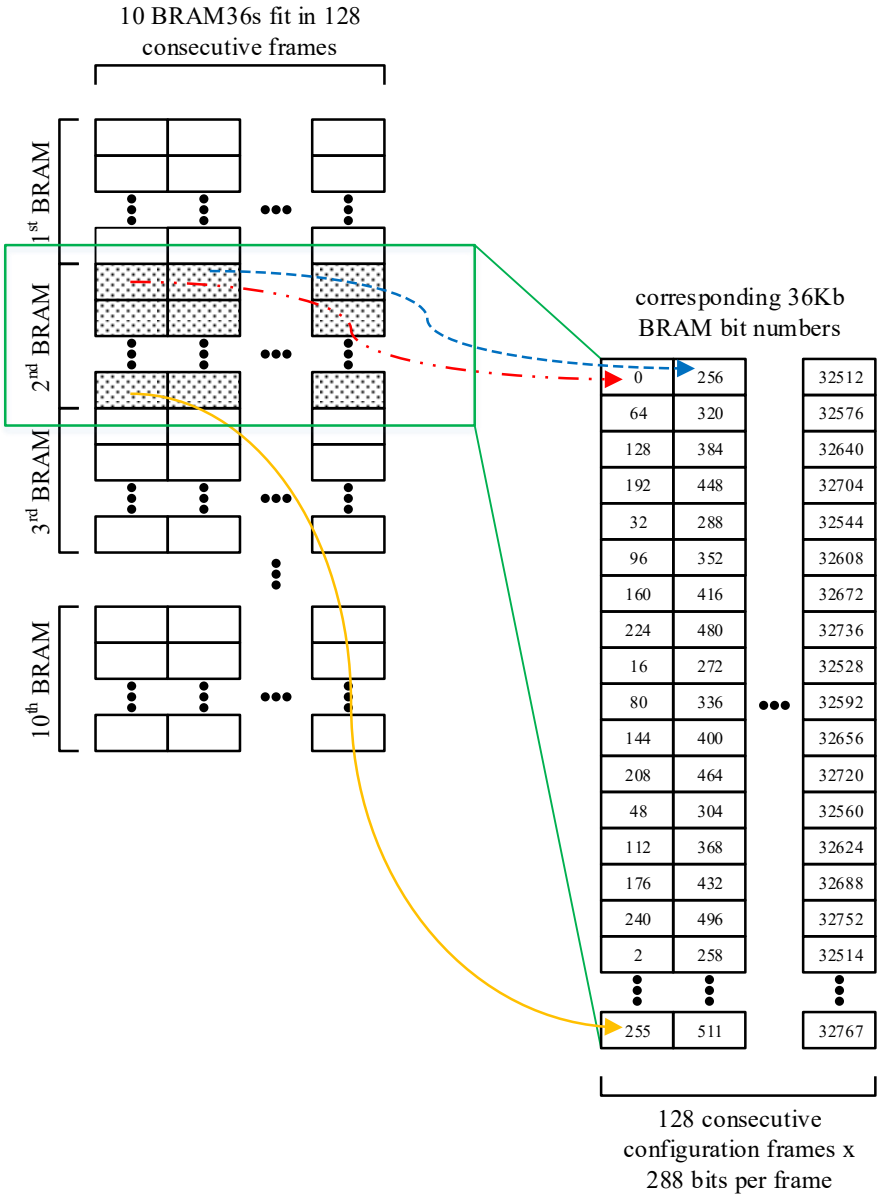


Figure 4.19: BRAM layout

It is important to emphasize that the manufacturers (e.g., AMD) do not disclose important construction details, such as the memory interleaving scheme. Therefore, a reverse engineering analysis was performed to investigate the device's internal architecture and identify its design features

4.2.1.5 SELs

We did not observe latch-ups during the heavy ions radiation tests, which is in accordance with the previous works showing that SELs only appear at LETs higher than 16 MeVcm^2/mg [133].

4.2.2 Protons

4.2.2.1 SET-induced Events and Single-event Resets

Analysis of the proton results revealed similar phenomena with the heavy ion experiments, as presented in Table 4.19. In an event observed in all energies, the 4 out of 8 FFs of a SLICE are affected. All the affected FFs are initially configured with asynchronous reset or preset and their reset, preset and clock inputs are disabled. Since these FFs are preloaded with either 0s or 1s, this phenomenon can be due to a SET on the reset, preset or clock signal. Another phenomenon we observed is that all the 8 FFs of a SLICE are affected. In this case, the SLICE's 8-stage register is preloaded with alternate 0-1 pattern, so a SET may hit the clock branch that drives the SLICE.

Regarding SRLs, we did not observe any reset phenomena similar to the GSI heavy-ion experiment. In both the GSI and PSI experiments, the SRL clock and clock-enable signals were set to 0, thus, it is rare for an event to simultaneously perturb the clock and the clock-enable signals. In contrast, given that the CERN CUT was implemented by setting the clock-enable to high, several SRL phenomena were observed, as described in 4.2.1.1.

The configuration of LUTs as 64-bit DRAM was first tested in the proton experiments. We observed some events where one DRAM LUT of a SLICE is affected, i.e., 16, 26, 29 or 30 bits, initialized to either 0s or 1s, are flipped. Given that the DRAM's clock and clock-enable signals were set to low and its address input was also set to 0, this event cannot be a SET on the clock or any other control signal, but it seems to be a single-event reset.

The 1K, 2K & 3K single-event resets, described in subsection 4.2.1.1 for heavy ions, were also observed in proton experiments. However, the BRAM single-event resets in protons are more extensive since we noticed several cases where multiple of 512 bits of the same BRAM (i.e., 0.5K, 1K, 1.5K, 2K, … 9K) are affected by a single event.

All the above effects appear to increase significantly with the proton energy, as shown in Table 4.19 and according to the fluence per test session presented in Table 4.8.

Memory category	Event description	Energy [MeV] / Fluence [p/cm^2]				
		30 / 2.00x10^{10}	**50 / 3.93x10^{10}**	**100 / 3.93x10^{10}**	**150 / 2.00x10^{10}**	**200 / 9.87x10^{10}**
FF	all FFs of a SLICE	-	11	10	9	41
	4 of the 8 FFs of a SLICE	1	27	22	14	67
DRAM	1 DRAM LUT of a SLICE	-	6	2	2	11
BRAM	0.5 Kbits upset	-	-	-	-	1
	1 Kbits upset	-	1	1	-	-
	1.5 Kbits upset	-	1	1	-	2
	2 Kbits upset	-	-	1	1	-
	2.5 Kbits upset	-	1	-	2	2
	3 Kbits upset	-	-	1	-	2
	3.5 Kbits upset	-	-	1	-	1
	4 Kbits upset	2	-	-	1	-
	4.5 Kbits upset	-	-	-	-	1
	5 Kbits upset	-	-	1	1	-
	5.5 Kbits upset	-	-	1	-	2
	6.5 Kbits upset	-	-	1	-	-
	7 Kbits upset	-	1	-	-	2
	7.5 Kbits upset	-	-	1	-	2
	8 Kbits upset	-	-	1	-	-
	8.5 Kbits upset	-	-	1	-	1
	9 Kbits upset	-	-	-	-	1

Table 4.19: Proton SET-induced events and single-event resets

4.2.2.2 SEUs and Cross Section

To calculate the SEUs for all the different memory categories and the corresponding cross sections, we followed the same approach with the heavy ions, i.e., all the effects of the previous subsection were removed from the total upsets. Table 4.20 presents the total upsets and SEUs observed in the different memory types for the fluence and the five proton energies of the tests. Remember that two versions of the CUT were used depending on the configuration mode of the SLICEM LUTs, i.e., SRLs and DRAMs. The first benchmark was used to characterize the behavior of the SRL LUTs and the second the behavior of the DRAM LUTs, while both benchmarks were taken into consideration for calculating the cross sections for CRAM, BRAM and FFs.

Table 4.21 and Figure 4.20 show the SEU cross sections per bit for all the different memory types and proton energies. To calculate the error bars, we made the same assumptions as in the heavy-ion cross sections, i.e., a Poisson distribution of the SEUs, a confidence level of 95% and an uncertainty on the measured fluence of 10% [48], [80].

Comparing the results with the literature, we observed that our tests show an order of magnitude higher cross section for CRAM, e.g., in [136], CRAM SEU cross section saturates at 6×10^{-16} instead of 6×10^{-15} cm^2/bit. It is worth noting that it is the first time that proton cross sections for the FFs and SLICEM LUTs of the Zynq-7000 FPGAs are presented in the literature; the results were published in [153].

Energy [MeV]	Fluence [p/cm^2]	Upsets	CRAM	CRAM essential	BRAM	FF	LUT SRL*	LUT DRAM*
30	2.00×10^{10}	Total upsets	2420	1295	9315	31	209	227
		SEUs	2420	1295	1124	27	209	227
50	3.93×10^{10}	Total upsets	11967	6484	16914	325	1011	1094
		SEUs	11967	6484	4626	129	1011	929
100	3.93×10^{10}	Total upsets	14080	6842	58377	349	1063	1027
		SEUs	14080	6842	5128	181	1063	981
150	2.00×10^{10}	Total upsets	8110	4567	19795	219	575	602
		SEUs	8110	4567	3411	91	575	542
200	9.87×10^{10}	Total upsets	40716	23421	98146	1126	2909	3136
		SEUs	40716	23421	16215	530	2909	2838

*The fluence for LUT SRL and LUT DRAM is half the value shown in the corresponding column.

Table 4.20: Proton total upsets & SEUs

In addition, comparing the proton with the heavy ion cross sections (Figure 4.15) we observed that the CRAM cross section is smaller than the other cross sections in both protons and heavy ions. Also, the BRAM and CRAM essential cross sections are higher than the CRAM cross section, while the FFs and LUTs (SRL & DRAM) have the highest cross sections, i.e., FFs, SRL LUTs and DRAM LUTs are more vulnerable in protons than CRAM and BRAM, especially in higher energies. All the cross sections increase with the proton energy and saturate at 150 MeV and above energies.

Energy [MeV]	CRAM [cm²/bit]	CRAM essential [cm²/bit]	BRAM [cm²/bit]	FF [cm²/bit]	LUT SRL [cm²/bit]	LUT DRAM [cm²/bit]
30	$1.70\text{x}10^{-15}$	$2.07\text{x}10^{-15}$	$2.49\text{x}10^{-15}$	$3.08\text{x}10^{-15}$	$4.64\text{x}10^{-15}$	$5.02\text{x}10^{-15}$
50	$4.28\text{x}10^{-15}$	$5.28\text{x}10^{-15}$	$5.22\text{x}10^{-15}$	$7.51\text{x}10^{-15}$	$11.43\text{x}10^{-15}$	$10.50\text{x}10^{-15}$
100	$5.05\text{x}10^{-15}$	$5.56\text{x}10^{-15}$	$5.79\text{x}10^{-15}$	$10.54\text{x}10^{-15}$	$12.02\text{x}10^{-15}$	$11.08\text{x}10^{-15}$
150	$5.70\text{x}10^{-15}$	$7.29\text{x}10^{-15}$	$7.55\text{x}10^{-15}$	$10.39\text{x}10^{-15}$	$12.72\text{x}10^{-15}$	$12.03\text{x}10^{-15}$
200	$5.81\text{x}10^{-15}$	$7.58\text{x}10^{-15}$	$7.28\text{x}10^{-15}$	$12.28\text{x}10^{-15}$	$13.05\text{x}10^{-15}$	$12.78\text{x}10^{-15}$

Table 4.21: Proton cross section

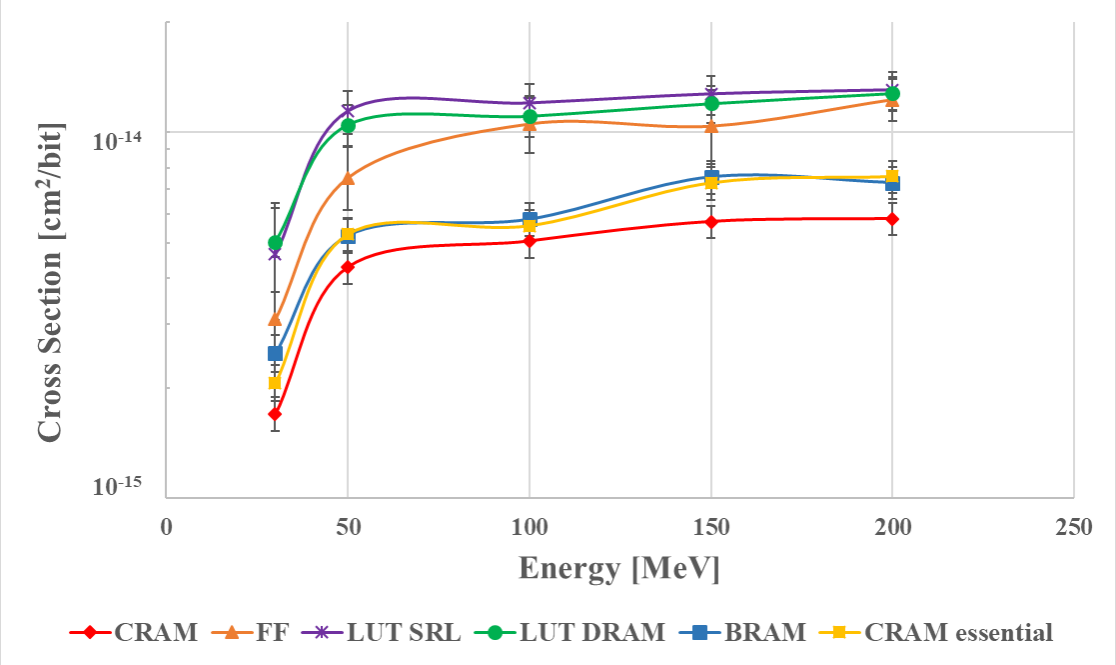


Figure 4.20: Proton SEU cross section vs Energy

Analyzing the SEUs along with their initial values, we calculated the likelihood of upset occurrence in memory cells initialized with '0' or '1'; the results for all the memory types are presented in Table 4.22. The likelihood of an upset in a CRAM cell is always greater when it is preloaded with '1'. Moreover, the ratio of the likelihood of the upsets 0→1 : 1→0 decreases as energy increases, and it nearly subdoubles at 200 MeV versus 30 MeV.

On the other hand, the corresponding FF and BRAM likelihood ratio increases with the proton energy, in contrast to what happens in heavy ions with respect to LET. In the case of BRAM, the likelihood ratio increases slightly, while in the case of FFs, it doubles to 1 over 100 MeV compared to the lowest energy. The LUT SRLs likelihood ratio seems to be independent of the proton energy and approximately equal to 1, as in the heavy ions

experiments where it was also equal to 1 in all LETs. As for the LUT DRAMs, the ratio 0→1 : 1→0 is also independent of the proton energy but approximately equal to 0.7.

Memory category	Energy [MeV]				
	30	50	100	150	200
CRAM	1 : 1.16	1 : 1.27	1 : 1.70	1 : 1.92	1 : 1.97
BRAM	1 : 0.98	1 : 0.93	1 : 0.88	1 : 0.81	1 : 0.77
FF	1 : 2.10	1 : 1.23	1 : 0.92	1 : 1.11	1 : 1.07
LUT SRL	1 : 0.97	1 : 1.00	1 : 1.00	1 : 0.97	1 : 1.01
LUT DRAM	1 : 0.57	1 : 0.73	1 : 0.66	1 : 0.72	1 : 0.68

Table 4.22: Proton likelihood to upset 0→1 : 1→0

4.2.2.3 MBU/MCU Analysis

Following subsection 4.2.1.4, which presents the results of the MBU/MCU analysis for the heavy ion experiments, the corresponding analysis for the proton experiments is presented here. Table 4.23 shows the number of occurrences of single and multiple-bit upsets observed within a single CRAM frame.

As in heavy ions (Table 4.14), the number in parenthesis indicates the occurrences of MBUs, based on the assumption that when two or more adjacent bit flips occur in a frame, this is most likely due to a single event, while the remaining upsets probably were generated by multiple accumulated events. The vast majority of events are SBUs and most of the multi-bit upsets captured in one CRAM frame are MBUs. Several 6-bit (or more, up to 33-bit) upsets per frame were observed, most of which are not MBUs, e.g., less than one third of 6-bit (or more) upsets per frame at 200 MeV are MBUs. This observation applies mainly at high fluences (e.g., more than 2.00×10^{10} p/cm^2), which may explain the scattered occurrence of accumulated upsets.

Upsets per CRAM frame	Energy [MeV] / Fluence [p/cm^2]				
	30 / 2.00×10^{10}	50 / 3.93×10^{10}	100 / 3.93×10^{10}	150 / 2.00×10^{10}	200 / 9.87×10^{10}
1	2223	10440	11789	6593	31892
2	73 (71)	514 (466)	751 (699)	521 (493)	2758 (2503)
3	5 (5)	88 (88)	159 (138)	98 (82)	602 (536)
4	3 (3)	18 (15)	26 (25)	28 (28)	182 (149)
5	-	2 (2)	7 (3)	6 (5)	59 (37)
6	2(0)	4 (4)	5 (3)	-	17 (8)

Upsets per CRAM frame	Energy [MeV] / Fluence [p/cm^2]				
	30 / 2.00x10^{10}	**50 / 3.93x10^{10}**	**100 / 3.93x10^{10}**	**150 / 2.00x10^{10}**	**200 / 9.87x10^{10}**
7	-	4 (4)	-	1 (1)	8 (4)
8	-	-	4 (1)	4 (0)	3 (1)
9	-	1 (0)	-	-	-
10	-	1 (0)	1 (0)	-	3 (1)
11	-	1 (0)	-	-	1 (0)
12	1(0)	-	1 (0)	-	2 (0)
13	-	1 (0)	-	-	-
14	-	-	1 (1)	-	2 (0)
15	-	-	1 (0)	-	-
16	-	-	1 (0)	-	1 (0)
18	-	1 (0)	-	-	3 (0)
20	-	2 (0)	1 (0)	-	1 (0)
22	-	-	-	-	1 (0)
24	-	-	1 (0)	-	-
26	-	-	-	-	1 (0)
33	-	-	-	-	2 (0)

Table 4.23: Proton upsets per CRAM frame

Table 4.24 shows the MCU shapes and their percentage of occurrence for the five different proton energies. We followed the same approach [155] with the heavy ion results to identify the proton MCUs for the CRAM. The x-axis of the shapes represents consecutive frame address, while the y-axis consecutive bits in a frame.

The analysis of proton MCUs and the comparison with heavy ion MCUs conclude:

- The MCU patterns extend up to four frames and up to five bits per frame instead of two frames and up to four bits per frame observed in the heavy ion experiments.
- MCUs spanning more than two frames appear very few times (i.e., less than 2.5%).
- Most upsets (i.e., more than 90%) are SBUs and 2-bit MBUs/MCUs independent of the proton energy; the same is valid for the heavy ions and does not depend on the LET.
- The percentage of SBUs decreases as the proton energy increases and saturates at the energy of 100 MeV and above.
- Several MCUs occur only at energies higher than 50 MeV, especially the MCUs with four or more bit flips.

																
30 MeV	76.3	15.5	3.67	1.49	-	0.80	0.43	0.48	0.48	0.16	-	-	0.21	0.05	-	0.05
50 MeV	69.4	17.8	5.80	1.91	0.33	0.96	0.58	0.51	0.54	0.25	0.28	0.21	0.29	0.27	-	0.02
100 MeV	64.7	18.1	7.61	1.59	0.63	0.92	0.67	0.69	0.71	0.53	0.46	0.42	0.31	0.24	0.13	0.17
150 MeV	63.3	16.5	8.46	1.67	1.21	1.14	0.83	0.79	0.70	0.68	0.42	0.61	0.26	0.06	0.15	0.19
200 MeV	63.4	16.5	7.78	1.98	1.14	1.10	0.92	0.83	0.78	0.62	0.59	0.56	0.27	0.26	0.18	0.18
																
30 MeV	-	-	-	0.05	-	-	-	0.05	0.05	-	-	-	0.05	0.05	-	-
50 MeV	0.02	0.05	0.05	0.05	-	0.05	0.04	0.15	0.13	0.02	-	-	0.04	-	-	0.02
100 MeV	0.16	0.20	0.11	0.10	0.08	0.13	0.06	0.17	0.10	0.06	0.11	0.06	0.10	0.10	0.02	0.08
150 MeV	0.23	0.23	0.15	0.17	0.19	0.21	0.13	0.21	0.17	0.08	0.06	0.04	0.04	0.08	0.08	0.04
200 MeV	0.18	0.18	0.17	0.16	0.16	0.16	0.15	0.15	0.13	0.12	0.10	0.09	0.09	0.08	0.08	0.08
																
30 MeV	-	-	0.05	-	-	-	-	-	0.05	-	-	-	-	-	-	-
50 MeV	0.02	-	0.02	0.02	-	0.02	-	-	-	-	0.04	-	0.01	0.02	-	-
100 MeV	0.04	0.05	0.03	0.02	0.06	0.04	0.02	0.03	-	-	0.04	-	0.02	0.05	0.02	-
150 MeV	0.06	0.15	-	0.02	0.15	0.11	0.04	0.04	-	0.08	-	-	0.08	0.11	0.04	0.04
200 MeV	0.07	0.07	0.07	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.04	0.03	0.03	0.03

Table 4.24: Proton percentage frequency of MCU patterns in CRAM

Figure 4.21 presents the percentage of MCUs by size as a function of energy. At energies above 100 MeV, the percentage of occurrence for each bit multiplicity (1-bit or 2-bit or ≥3-bit MCUs) remains constant. At low energies, i.e. 30 to 50 MeV, as the energy increases, the percentage of SBUs decreases; this SBU drop is equally distributed between the 2-bit and ≥3-bit MCUs. Notice that the percentage of 2-bit MCUs does not exceed that of SBUs at any energy, as in heavy ions of higher LETs.

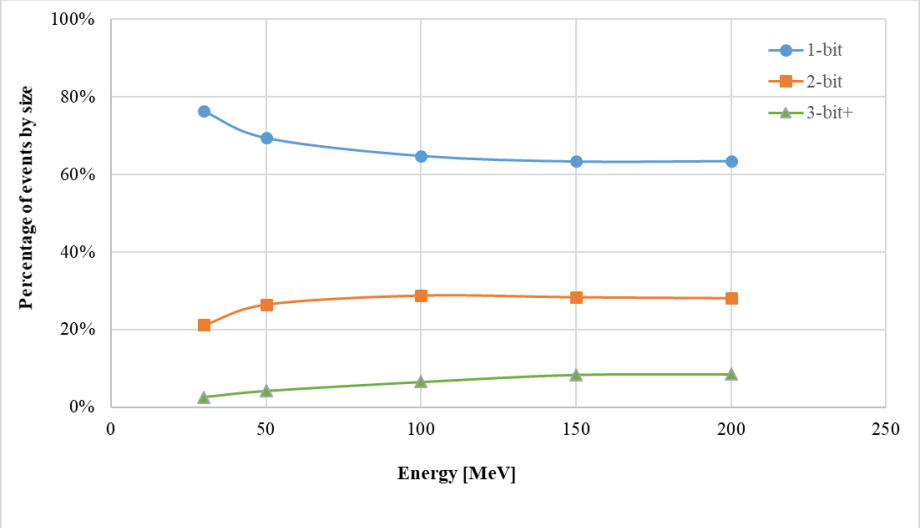


Figure 4.21: Proton percentage of MCUs by size as a function of energy

The above analysis considers only the CRAM bits that the CUT does not dynamically alter and, thus, can be detected and corrected by a scrubbing mechanism. Given that the AMD Xilinx SEM IP in enhanced repair mode [12] supports the correction of single-bit or adjacent double-bit errors per frame, the MBUs with multiplicity higher than two are not correctable. The MBU patterns that the SEM IP core is not able to correct are highlighted in Table 4.24. The number of uncorrectable - by the SEM IP core - MBUs and their cross sections for different proton energies are shown in Table 4.25 and Figure 4.22. The cross section in 200MeV is more than one order of magnitude higher than the cross section in 30MeV.

Energy [MeV]	**Fluence [p/cm²]**	**Uncorrectable MBUs (by SEM IP)**	**Cross section [cm²/bit]**
30	2.00×10^{10}	6	0.42×10^{-17}
50	3.93×10^{10}	63	2.26×10^{-17}
100	3.93×10^{10}	109	3.91×10^{-17}
150	2.00×10^{10}	70	4.92×10^{-17}
200	9.87×10^{10}	433	6.18×10^{-17}

Table 4.25: Protons - uncorrectable (by SEM IP) MBUs

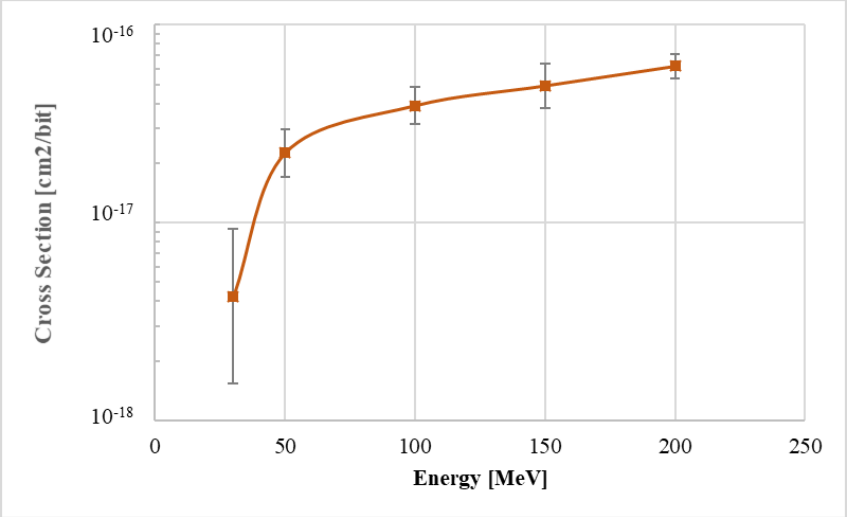


Figure 4.22: Uncorrectable MBUs cross section vs proton energy

Considering the MBU analysis from both the heavy ion (Table 4.16) and proton (Table 4.25) experiments, it is evident that the MBUs in the CRAM cannot always be corrected by the embedded ECC and the SEM IP core of the Zynq-7000 devices and a more sophisticated scrubbing scheme is required, able to correct multiple-bit errors. A novel scrubbing technique, which improves the multiple-bit error correction capabilities of the Zynq-7000 on-chip ECC scheme, is presented in chapter 6 of this thesis.

Regarding the FFs and SRLs, the MCU analysis showed SBUs in these memories as well as 2-bit upsets in the SRLs, as in heavy ions. Remember that 2-bit upsets are actually SBUs propagated to the always-open SRL master latch due to its master-slave arrangement (Figure 4.18).

The MCU analysis for the DRAM, presented for first time in this thesis since the CUT did not support the DRAM testing in the heavy ion experiments, follows, and the results are depicted in Table 4.26. First, we observe that about 80% to 90% of the upsets are SBUs for all the proton energies. The DRAM MCU patterns extend to two frames and up to two bits per frame. The vast majority of MCU patterns indicate that two, three or four adjacent bits of a 64-bit DRAM are affected, according to the DRAM layout (Figure 4.23) as defined in the logic location file generated by the AMD Xilinx tools. For example, a 2-bit upset of bits 1 & 2 of a DRAM corresponds to the second pattern of Table 4.26, a 2-bit upset of bits 2 & 3 to the third pattern, a 4-bit upset of bits 1, 2, 3 & 4 to the sixth pattern and a 2-bit upset of bits 31 & 32 to the eighth pattern. The only MCU shape that does not include neighboring bits is the fourth shape shown in Table 4.26 (e.g., bits 3 & 5 or 37 & 39 of a DRAM).

									
30 MeV	86.6	9.79	1.55	0.52	0.52	0.52	-	-	-
50 MeV	89.4	7.59	1.93	0.24	0.24	0.24	0.12	-	-
100 MeV	85.9	8.03	3.24	0.96	0.36	0.48	0.36	0.12	0.12
150 MeV	87.7	6.60	2.13	1.49	0.64	0.64	0.43	-	-
200 MeV	83.3	7.79	3.94	2.12	0.64	0.59	0.47	0.21	0.17

Table 4.26: Proton percentage frequency of MCU occurrences in DRAM

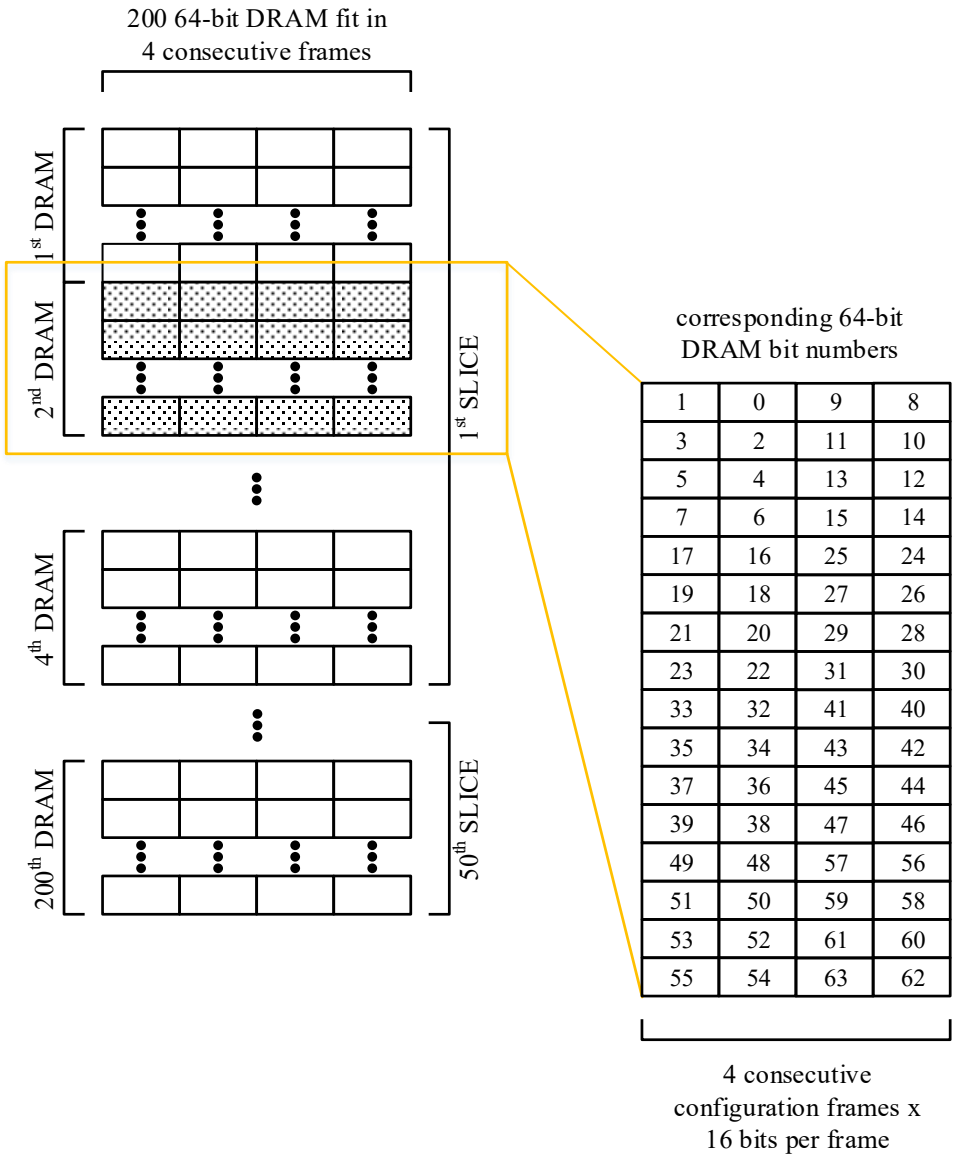


Figure 4.23: LUT DRAM layout

Regarding the BRAM, the MCU analysis revealed the patterns shown in Table 4.27. First, the frequency of the MCU occurrence increases with the proton energy. Compared with the heavy ions (Table 4.18), only the second MCU shape of Table 4.27 appears in both experiments, while all the others are observed only in the proton tests. Finally, studying the MCU shapes and the BRAM layout (see Figure 4.19) does not provide any indication for the existence of an interleaving scheme in the BRAM.

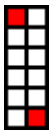
								
30 MeV	95.8	3.87	-	0.31	-	-	-	-
50 MeV	93.3	5.40	0.48	0.18	0.20	-	0.08	0.05
100 MeV	87.9	8.30	1.14	0.37	0.40	0.42	0,10	0.15
150 MeV	87.0	8.34	1.78	0.54	0.29	0.37	0.08	0.12
200 MeV	85.3	8.08	1.88	0.53	0.49	0.47	0.47	0.22
								
30 MeV	-	-	-	-	-	-	-	-
50 MeV	0.03	-	-	-	0.08	-	0.03	-
100 MeV	-	0.12	0.07	0.17	0.10	0.10	0.10	0.05
150 MeV	0.08	-	0.25	0.08	0.08	-	-	-
200 MeV	0.22	0.20	0.17	0.16	0.16	0.15	0.08	0.07

Table 4.27: Proton percentage frequency of MCU occurrences in BRAM

4.2.2.4 SELs

We did not observe latch-ups during the proton experiments, as in the heavy ion tests.

5 Single Event Effects on the Zynq-7000 PS

While many studies have performed accelerated radiation testing on the Zynq-7000 PS, a significant research gap remains; thorough data covering the entire scope of major SRAM arrays in the PS and a complete spectrum of heavy ion LETs and proton energies for space missions is still missing. Thus, without comprehensive data, it is challenging to precisely determine the SEE rates and failure modes of the Zynq-7000 PS, as the current literature does not provide a complete picture. This work fills this gap by presenting results from two radiation campaigns, one with protons and one with high-energy heavy ions.

In [22], the authors tested the Zynq-7000 OCM and L1 cache using a low-energy proton beam of 18 MeV, while in [40], the SEE characterization of L1 cache was performed for the low-energy proton range of 3 to 10 MeV. In [130], a comprehensive analysis of the SEEs observed on the OCM is presented using proton energies from 16 to 200 MeV. Finally, in [136], the authors presented results for the L2 cache using the full range of proton energies, from 50 to 250 MeV. From the above, it follows that OCM and L2 cache results for the full range of proton energies have already been published in the literature. On the other hand, L1 results cover only low proton energies. In this chapter, the full-range proton energies results are presented not only for OCM (static and dynamic testing) and L2 (dynamic testing) but also for L1 (static testing).

Regarding heavy ions, in [135] the authors tested the OCM for SEUs with LETs varying from about 2.6 to 17 MeV·cm^2/mg and in [158] and [159], the authors performed heavy-ion experiments with LET of 13.59 and 25 MeV·cm^2/mg, respectively for both OCM and L1. In this work, the LET range for OCM and L1 is extended up to 32.02 MeV·cm^2/mg.

Finally, no work presents the impact of radiation-induced errors at the application-level by executing software benchmarks on the ARM processor and measuring silent data corruptions (SDC) (i.e., errors in system data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data) and system crashes. This work contributes to the study of the impact of radiation effects at the application level by running various processor benchmarks and for different cache configurations, for the entire range of proton energies, from 30 to 250 MeV, and for heavy ion LETs equal to 21.07 and 32,02 MeV·cm^2/mg.

The key points of this study include:

- The calculation of the SEE cross sections for all the different embedded memories of the PS part (i.e., OCM, L1 and L2 caches) for a large spectrum of heavy ion LETs and proton energies.
- The study of the SEEs at the application level by running various processor benchmarks for different cache configurations (i.e., L1 and L2 enabled, L1 and L2 disabled, only L1 enabled, only L2 enabled).
- The accurate prediction of the PL and PS SEE rates of a Zynq-7000 device by taking into account the results of this study, covering a wide range of heavy ion LETs and the full range of proton energies, for three different orbits: Earth observation orbit, International Space Station (ISS) orbit, and the orbit the Galileo constellation operates.

5.1 Radiation Tests

5.1.1 Test Facilities

The radiation tests were performed a) under high energy protons at the Paul Scherrer Institute (PSI) – Proton Irradiation Facility (PIF) in Villigen, Switzerland and b) under heavy ions at the GSI Helmholtz Centre for Heavy Ion Research in Darmstadt, Germany.

The first radiation test campaign was carried out at PSI-PIF in parallel with the third test campaign for the Zynq-7000 PL described in chapter 4. The target device was irradiated under realistic proton spectra with five different proton energies: 30, 50, 100, 150 and 200 MeV, emulating proton radiation encountered across various potential orbits in space. The beam spot had a rectangular shape, measuring 40 mm × 40 mm. The beam was located in the center of the target area. The beam flux varied between 1.8×10^{7} to 1.2×10^{8} p/cm^2/sec depending on the proton energy.

The second test campaign took place at the beam line SIS18 of the GSI using Uranium (U^{73+}) ions at two different energies, 190 and 330 MeV/u. The irradiations were executed in air, and thus, the energy on the surface of the chip was lowered to approximately 150 and 300 MeV/u, respectively, assuming a 2 mm scintillator and a 30 cm distance between the beam and the target device. The beam was delivered in spills by an intensity-modulated raster scanner. A pencil beam scanned the target area with a 1 mm step size. The scan speed was modulated according to the flux measurement to guarantee a homogeneous fluence distribution over the scan area. The beam flux during one scan cycle ranged between 2,000 and 2,700 ions/cm^2.

The beam features of the two facilities are summarized in Table 5.1.

	PSI PIF	GSI SIS18
Date of campaign	10-12 Sep 2021	08-09 April 2022
Irradiation type	Proton	Heavy-Ion U^{73+}
Energy	30, 50, 100, 150 and 200 MeV	190 and 330 MeV/u (chip surface 150 and 300 MeV/u)
Flux	1.8×10^7 to 1.2×10^8 p/cm^2/sec	4.6×10^4 to 3.6×10^5 ions/cm^2/sec
Size	40 mm × 40 mm	1 mm × 1 mm

Table 5.1: Radiation test facilities

5.1.2 Test Setup

The Zynq-7000 XC7Z045 SoC attached to the AMD Xilinx ZC706 [15] was used for both experiments. The ZC706 is a commercial development board that provides a hardware environment for developing and evaluating designs targeting the Zynq-7000 architecture. It also provides features common to many embedded processing systems, including DDR3 memory. The XC7Z045 PS features are shown in Table 5.2 [15].

PS features	XC7Z045
Processor Core	Dual-core ARM Cortex-A9 MPCore with CoreSight
L1 Cache	32 KB Instruction, 32 KB Data (independent for each core)
L2 Cache	512 KB (shared between the cores)
On-Chip Memory (OCM)	256 KB
External Memory Support	Multiprotocol dynamic memory controller 16-bit or 32-bit interfaces to DDR memories ECC support in 16-bit mode

Table 5.2: Zynq-7000 PS features

The test setup shown in Figure 5.1 was used in both the heavy ions and protons radiation experiments. It includes the target Board, one remotely controlled DC power supply unit (PSU), two control laptops (CL) located in the beam room for control and monitoring purposes, one remote laptop (RL) located in the control room for remote control and several interfaces.

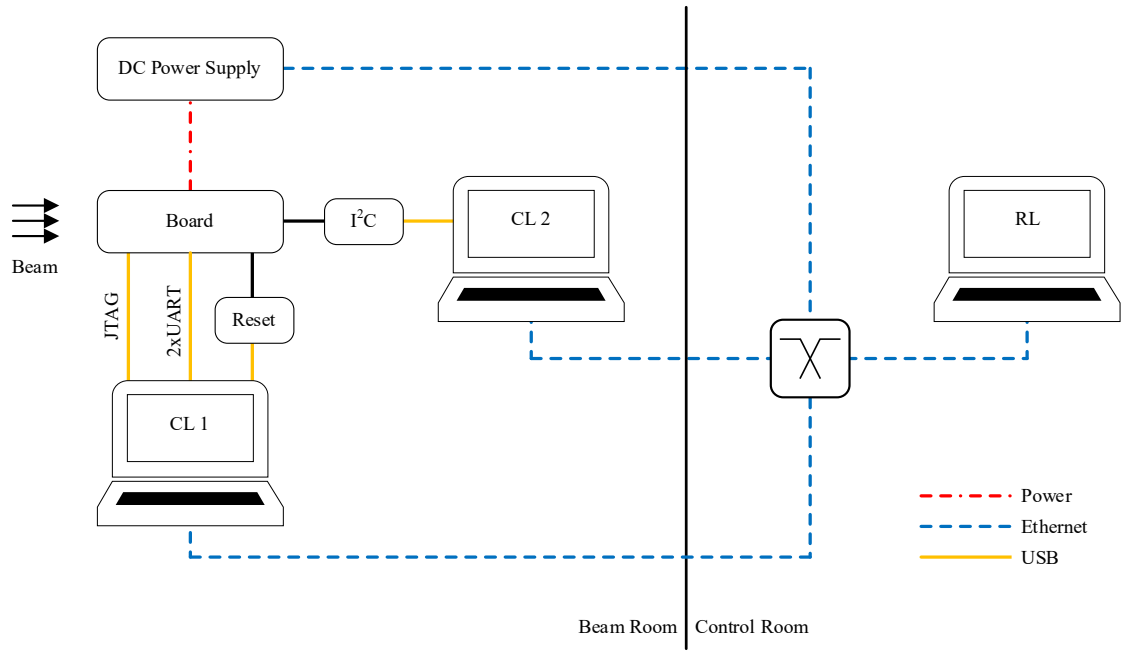


Figure 5.1: Radiation test setup

The primary test operations are summarized below:

- The Board (ZC706) is powered by the DC PSU (Siglent SPD3303X-E) remotely controlled by RL.
- The Board is remotely reset by shorting the Board's Power-on-Reset button (PS_POR_B) through a relay module. The CL-1 can turn the relay module on and off (set/reset) through a USB-to-serial adapter module (Reset). This option is used to reset the target device in the event of a system crash.
- The CL-1 is connected with the PL JTAG port of the target device for PL configuration and logging purposes.
- The CL-1 is connected through an onboard USB-to-UART bridge with the UART interface of the ARM SoC. ARM CPU uses this connection to send results to the CL-1 from the execution of the benchmarks.
- The CL-1 sends test commands and receives test responses to/from the ARM CPU through the PL JTAG and PS UART ports. Whenever the target device becomes unresponsive, it remotely resets the board as above.
- The CL-1 is connected through a USB-to-serial adapter module with the serial interface of the AMD Xilinx SEM IP to receive the log messages generated by the SEM IP. The SEM IP core is integrated into the Zynq-7000 PL to mitigate the SEUs from the CRAM during the execution of the PS tests.

- The CL-2 monitors the voltage and current Zynq-7000 power rails through the Board's I^2C-based power management bus (PMBus) interface to detect any high current events. A USB to GPIO interface adapter evaluation module (Texas Instruments USB-TO-GPIO EVM) is connected to the PMBus connector providing access to the onboard Texas Instruments power controller. The Texas Instruments Fusion Digital Power GUI tool runs in the CL- 2 to monitor the voltage and current values of the chip power rails.
- The CL-1 and CL-2 are remotely controlled and monitored by the RL.

The tests run continuously regardless of whether the beam is off or on. All system clocks are synchronized before the experiments with the facility's clock. All logs are time-stamped for post-processing.

In the first (protons) radiation campaign, test responses were accomplished through the PL JTAG port. In detail, a dedicated test interface logic was added to the PL subsystem to provide external access through the JTAG interface to the PS memories (via JTAG-to-AXI bridges). However, a few failures were observed in this test interface logic, likely due to upsets in the utilized FIFO memories that bridge the JTAG port with the PS AXI interface, which caused the corruption of the test responses. Thus, for the second (heavy ions) radiation campaign, we redesigned the test interfaces to also upload the CPU responses through the PS UART making the system more robust to radiation-induced errors.

Figure 5.2: Heavy-ion beam experiment at the GSI facility

Figure 4.5, which is presented in chapter 4, shows the ZC706 board irradiated in the beam room of the PSI facility, while Figure 5.2 shows the ZC706 board that was irradiated in the beam room of the GSI facility. Note that the proton test campaign was conducted together with the third test campaign for the Zynq-7000 PL.

All JTAG transactions with the target board to access the PS memories, generated by the CL-1, are performed through the same open-source FREtZ software platform used to run the PL tests [125], [145]. This platform has already been described in subsection 4.1.2.

5.1.3 System Under Test

The following components of the Zynq-7000 PS are involved in the radiation tests since either they include the targeted memory elements or they are used to perform the tests:

- Dual core ARM Cortex-A9
- L1 Caches (Instruction and Data), 32 KB each, 4-way set-associative
- Shared L2 Cache, 512 KB, 8-way set-associative
- Dual-ported OCM, 256KB, accessible by PS and PL
- External 32-bit DDR3 memory, 1 GB
- 64-bit ACP interface enabling cache-coherent accesses from PL to PS data in the L1 and L2 caches
- 64-bit/32-bit configurable, buffered AXI interfaces with direct access to DDR memory and OCM, referred to as high-performance (HP) AXI ports

Figure 5.3 shows the interfaces between the target system and the host. The host can access the targeted memories through the PL logic by sending JTAG commands. Two JTAG-to-AXI bridges [10] have been integrated into PL to translate the JTAG commands to AXI transactions and then access the OCM memory through the AXI HP port and the caches through the AXI ACP port. The DDR memory is accessed through the AXI HP port in the same way. These connections are used by the host to initiate the execution of the software tests (i.e., by writing commands in the DDR, which are parsed by the ARM CPU) and capture the test responses stored either in the DDR or in the targeted memories.

There are also two UART connections between the target board and the host: The first UART is used by the PS to transmit test responses, while the second UART is used by the SEM IP to log messages. The SEM IP core was used to prevent upset accumulation in the CRAM during the execution of the PS tests, thereby reducing the likelihood that the tests would fail due to malfunction in the PL circuit. For that reason, the JTAG-to-AXI components and the SEM IP in the PL part have been triplicated.

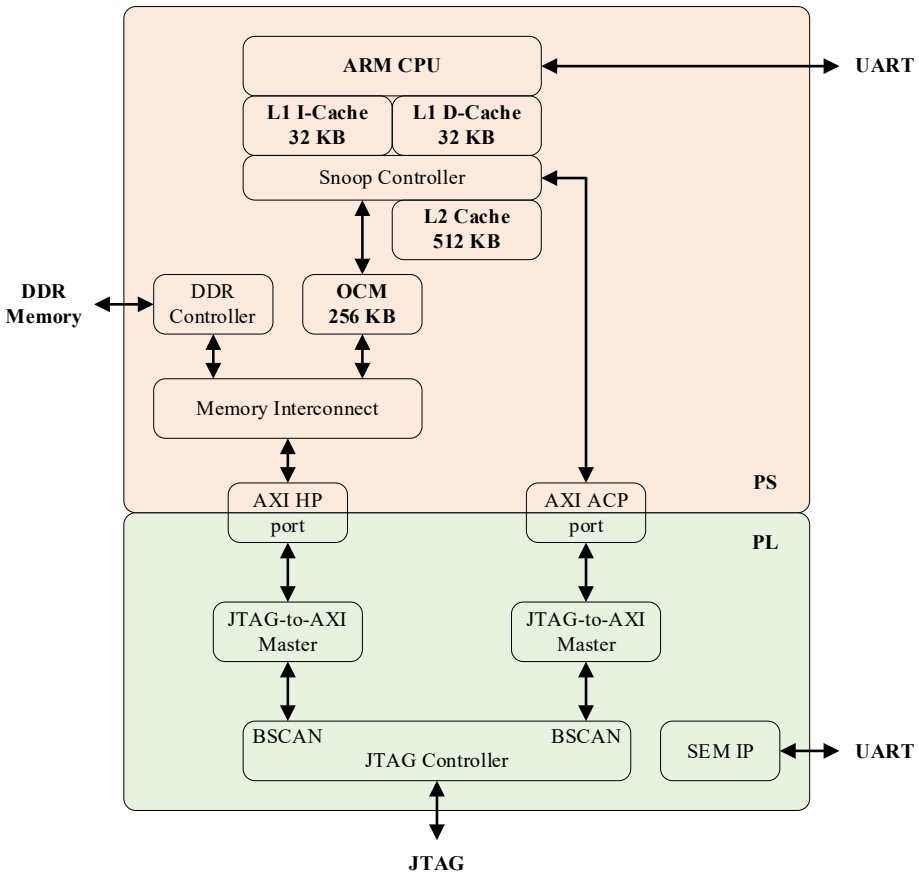


Figure 5.3: Interfaces between the host and the target device

5.1.4 Test Programs

Two types of test programs have been used in the experiments:

- Memory tests: synthetic static and dynamic test programs. Static test programs are mainly executed by the host application, i.e., the host reads and checks the contents of the targeted memories. Dynamic test programs are executed by the ARM CPU, which also checks and logs the results.
- Embedded microprocessor benchmarks: standard embedded microprocessor benchmarks are executed to emulate real-world applications. The benchmarks run on the ARM CPU without having an OS booted (i.e., bare-metal execution).

5.1.4.1 Static Tests

Static memory tests apply a fixed test pattern to the entire memory, and then the memory is scanned several times (N) to capture possible upsets. The memory is written by the

ARM CPU and read/scanned by the host PC. The test pattern is configurable, e.g. all 0s, or all 1s, or checkerboard pattern, or memory address value. Specifically, in this test setup, each memory location is written with its corresponding memory address.

Targeted Memory	Pattern	Repetitions (N)	Cache Configuration	Comments
OCM	Address	100	All caches disabled	-
L1 data cache	Address	200	L2 cache disabled	A 32-KB memory region is written to fill all the L1 data cache lines. During the test, no cache replacements occur, so all memory write/read transactions access the same L1 data lines.

Table 5.3: Static memory test features

Table 5.3 summarizes the static test programs executed in these experiments, including the targeted memory component, the fixed test pattern, the value of N, the cache configuration, and comments for implementing the test. In each test, the unused caches are disabled, e.g., during L1 cache testing, the L2 cache is disabled. Moreover, the cache test programs access a specific memory region to fill all the lines of the targeted cache and avoid cache replacements.

The static memory tests were performed for the OCM and L1 data cache.

5.1.4.2 Dynamic Tests

Dynamic memory tests write and read a known test pattern into all memory addresses of the targeted memory following a specific access sequence. The memory is written, read and compared by the ARM CPU, i.e., the host is not involved. Specifically, the dynamic test programs implement the March-based algorithm. The March-based algorithm [100] consists of a finite sequence of March elements. A March element is a read and write one or zero operation applied to every memory cell in either increasing or decreasing address order of the targeted component. In this test setup, the March-C algorithm is exploited, the most basic form of March-based algorithms, which consists of six March elements: ↕ (w0); ↑ (r0,w1); ↑ (r1,w0); ↓(r0,w1); ↓ (r1,w0); ↕ (r0). According to [142], dynamic tests can stress memory operation and may reveal upset phenomena not observed in the static tests. They represent real processor operation, therefore, the number of upsets may be different, since there is no error accumulation. The March-C elements, which are executed in a specific sequence, are described in Table 5.4.

The dynamic memory tests were performed for the OCM and L2 cache.

March Elements	Description
↕ (w0)	write zero to all memory cells in increasing or decreasing address order
↑ (r0, w1)	read (and compare with zero) and then write one to all memory cells in increasing address order
↑ (r1, w0)	read (and compare with one) and then write zero to all memory cells in increasing address order
↓ (r0, w1)	read (and compare with zero) and then write one to all memory cells in decreasing address order
↓ (r1, w0)	read (and compare with one) and then write zero to all memory cells in decreasing address order
↕ (r0)	read (and compare with zero) all memory cells in increasing or decreasing address order

Table 5.4: Dynamic March-C memory test

5.1.4.3 Benchmark Tests

The following six embedded microprocessor benchmarks were executed: CRC32, FFT, qsort, basicmath, SHA, MatrixMul. All benchmarks, except the MatrixMul, are sourced from the MiBench suite [144]. The MiBench suite is a benchmark suite designed to evaluate the performance of embedded systems. It's widely used in research and industry to assess different architectures, compilers, and optimizations for embedded processors. MiBench programs were modified to be ported to the ARM CPU as bare metal applications.

The memory footprint of the benchmarks is shown in Table 5.5. The data segment (.data) includes global and static variables, while the read-only data (.rodata) includes constant data. The memory stack is not considered for the calculation of the data segments; this is why the data segment in the cases of SHA and MatrixMul benchmarks is zero. Note that the input test sets of the benchmarks have been selected or modified to achieve different data memory segment lengths:

- The data segments of the FFT, basicmath, SHA and MatrixMul fit into the L1 data cache (32 KB).
- The data segment of qsort does not fit into the L1 data cache, but it fits into the L2 cache (512 KB); this means that during the execution of qsort, several replacements in the L1 data cache occur.
- The data segment of CRC32 does not fit into the L2 cache (512 KB); this means that during the execution of CRC32, several replacements in the L2 cache occur.

Benchmarks with various memory footprints were selected, to investigate the impact of the footprint on overall reliability.

Benchmark	.data [KB]	.rodata [KB]
CRC32	1,337.78	0.01
FFT	2.09	0.11
qsort	156.25	312.50
basicmath	6.09	0.00
SHA	0.00	2.32
MatrixMul	0.00	7.32

Table 5.5: Benchmarks' memory footprint

The caches are invalidated before every benchmark iteration. Each benchmark was executed with four different cache settings in order to emulate the impact of caches on the application level failures:

- both caches disabled,
- L1 cache enabled, L2 cache disabled
- L1 cache disabled, L2 cache enabled
- both caches enabled

Remember that the results from the execution of the benchmarks were recorded in two different ways for redundancy reasons: a) ARM CPU checks the output of the benchmarks, e.g., compares it with the golden signature, and sends the results into the UART port and b) ARM CPU stores the output of the benchmarks in a specific DDR memory region (2 KB size) and the host program reads this memory region using JTAG-to-AXI transactions.

5.1.5 Test Flow

The flow of PS tests is shown in Figure 5.4. It performs the following steps:

1. The host application resets the target device activating the Power-on-Reset (PS_POR_B) switch of the ZC706 board through a relay module. The FPGA device is configured and the ARM CPU 0 boots from the QSPI Flash memory. The bootloader copies the test programs (static, dynamic and benchmarks) stored in the QSPI Flash to the DDR memory. If the system has been configured and initialized properly, the ARM CPU asserts a READY flag and waits for the host application to start the tests. Note that all the flags used for the handshake between

the host application and the ARM CPU are bits stored in predefined memory addresses in the DDR. The host application accesses the flags through the JTAG-to-AXI interface.

2. The host application polls the READY flag. READY flag equals zero means that either the system has not been initialized properly, or the test program has been crashed and cannot be recovered through reset. In this case, the host application power cycles the board.

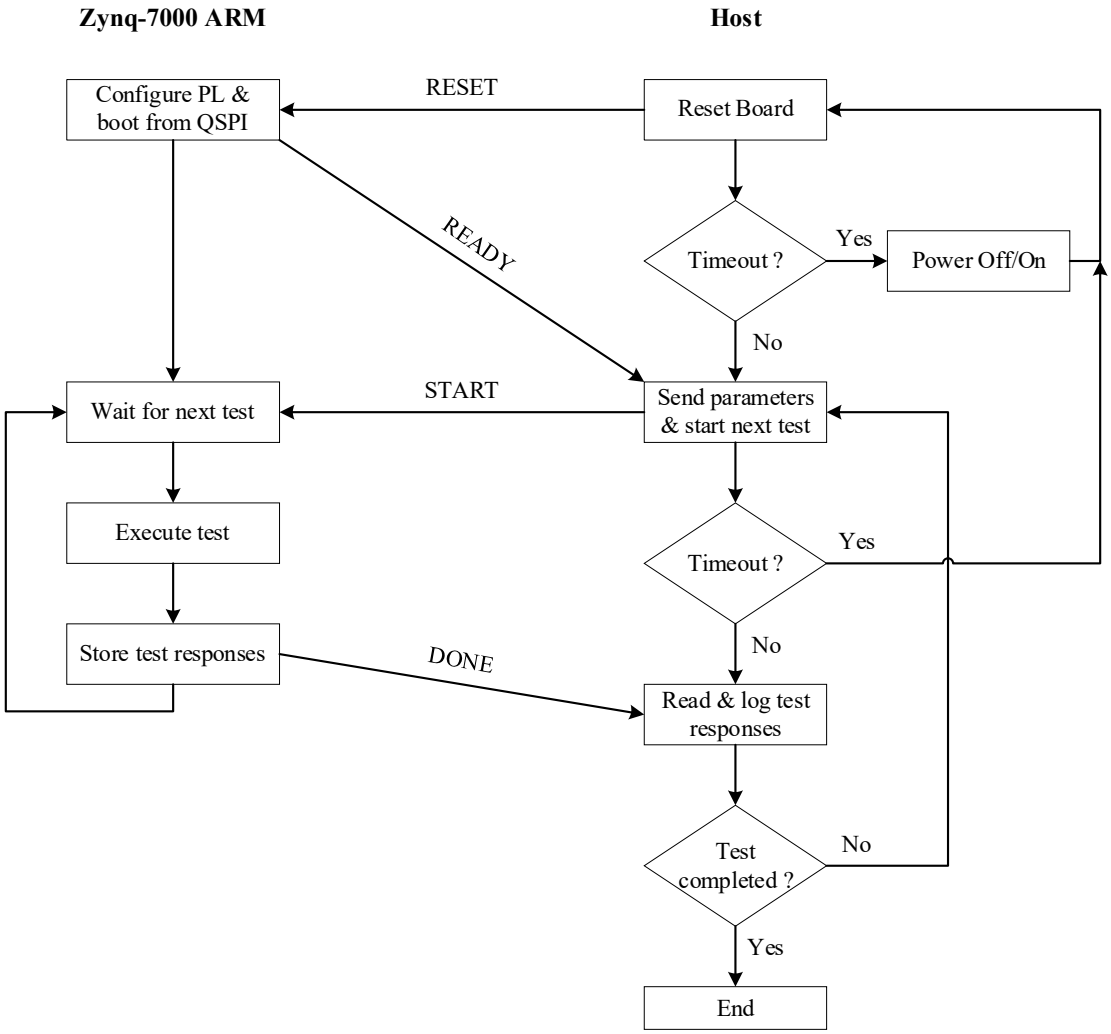


Figure 5.4: PS test flow

3. If the READY flag is set, the host application initiates the test sending the test parameters, including the test sequence (i.e., a list of tests to be executed), the number of iterations (i.e., how many times the ARM CPU will repeat the

sequence of tests), and the cache configurations. The test parameters are written to the DDR through the JTAG-to-AXI interface. Then, it activates a START flag.

4. If the ARM CPU receives the START flag, it executes the test sequence for the specified number of iterations. After each iteration, the ARM CPU checks the output (i.e., compares it with the golden data), and logs the report to the UART port and a specified memory region in the DDR. When it completes the test, it activates a DONE flag.
5. The host application polls the DONE flag; if a timer in the host application expires and the DONE flag is still zero, it means that the system has crashed. In this case, the host application logs the crash event and resets the board. Otherwise, it goes to the next test.
6. The host application repeatedly executes the same test sequence until the target fluence is reached for the current test setup. Then, the test is stopped.

5.1.6 Test Sessions

As mentioned in subsection 4.1.5, a fluence of approximately 10^{10} protons/cm^2 for proton testing or 10^7 ions/cm^2 for heavy ion testing is generally sufficient to yield statistically significant results [81], [90]. Thus, for Zynq PS test sessions, the minimum fluence reached around 10^{10} protons/cm² or 10^7 ions/cm², respectively.

Test Session	Energy [MeV]	Duration [secs]	Flux [p/cm²/sec]	Total fluence [p/cm²]
OCM	30	236	4.244 x 10^7	1.001 x 10^{10}
OCM	50	528	1.895 x 10^7	0.957 x 10^{10}
OCM	100	537	1.863 x 10^7	1.001 x 10^{10}
OCM	150	221	4.522 x 10^7	1.004 x 10^{10}
OCM	200	165	12.14 x 10^7	1.784 x 10^{10}
L1 cache	30	236	4.246 x 10^7	1.002 x 10^{10}
L1 cache	50	533	1.878 x 10^7	1.001 x 10^{10}
L1 cache	100	527	1.899 x 10^7	1.003 x 10^{10}
L1 cache	150	228	4.405 x 10^7	1.004 x 10^{10}
L1 cache	200	165	12.16 x 10^7	1.793 x 10^{10}

Table 5.6: Proton static memory test sessions

In PSI, the energy and flux were determined separately for each test session. The memory test duration was configured to reach at least a fluence of 10^{10} p/cm^2, as summarized in

Table 5.6 and Table 5.7. In the case of benchmarks, a sequence of benchmarks (CRC32, qsort, MatrixMul, FFT, basicmath, SHA) was executed for different cache configurations (i.e., all caches enabled, L1 enabled, L2 enabled and all caches disabled) and this sequence was repeated several times until a fluence of at least 10^{10} p/cm^2 was reached (Table 5.8). The tilt angle of the device under test with respect to the beam axis is equal to 0 degrees for all the proton tests.

Test Session	Energy [MeV]	Duration [secs]	Flux [p/cm^2/sec]	Total fluence [p/cm^2]
OCM	30	210	4.224 x 10^7	0.886 x 10^{10}
OCM	50	322	3.775 x 10^7	1.216 x 10^{10}
OCM	100	327	3.728 x 10^7	1.218 x 10^{10}
OCM	150	149	4.493 x 10^7	0.666 x 10^{10}
OCM	200	831	4.860 x 10^7	4.040 x 10^{10}
L2 cache	30	137	4.233 x 10^7	0.582 x 10^{10}
L2 cache	50	376	2.264 x 10^7	0.851 x 10^{10}
L2 cache	100	356	1.855 x 10^7	0.660 x 10^{10}
L2 cache	150	203	4.496 x 10^7	0.915 x 10^{10}
L2 cache	200	724	4.878 x 10^7	3.530 x 10^{10}

Table 5.7: Proton dynamic memory test sessions

Test Session	Energy [MeV]	Duration [secs]	Flux [p/cm^2/sec]	Total fluence [p/cm^2]
All caches enabled	30	180	4.24 x 10^7	0.761 x 10^{10}
All caches enabled	50	1246	1.88 x 10^7	2.346 x 10^{10}
All caches enabled	100	868	1.86 x 10^7	1.615 x 10^{10}
All caches enabled	150	339	4.50 x 10^7	1.527 x 10^{10}
All caches enabled	200	514	8.12 x 10^7	4.176 x 10^{10}
L1 enabled	30	382	4.24 x 10^7	1.620 x 10^{10}
L1 enabled	50	995	1.88 x 10^7	1.877 x 10^{10}
L1 enabled	100	988	1.86 x 10^7	1.847 x 10^{10}
L1 enabled	150	322	4.50 x 10^7	1.453 x 10^{10}
L1 enabled	200	701	9.54 x 10^7	6.689 x 10^{10}
L2 enabled	30	237	4.24 x 10^7	1.005 x 10^{10}
L2 enabled	50	713	2.14 x 10^7	1.523 x 10^{10}

Test Session	Energy [MeV]	Duration [secs]	Flux [p/cm^2/sec]	Total fluence [p/cm^2]
L2 enabled	100	1150	1.82 x 10^7	2.097 x 10^{10}
L2 enabled	150	354	4.50 x 10^7	1.598 x 10^{10}
L2 enabled	200	301	6.34 x 10^7	1.907 x 10^{10}
All caches disabled	30	565	4.24 x 10^7	2.400 x 10^{10}
All caches disabled	50	1474	1.88 x 10^7	2.780 x 10^{10}
All caches disabled	100	1397	1.86 x 10^7	2.600 x 10^{10}
All caches disabled	150	548	4.50 x 10^7	2.470 x 10^{10}
All caches disabled	200	659	11.91 x 10^7	7.850 x 10^{10}

Table 5.8: Proton benchmarks test sessions

The GSI radiation test was performed for two different energies and LETs, i.e. for energy 150 MeV/u with LET 32.02 MeVcm^2/mg and for energy 300 MeV/u with LET 21.07 MeVcm^2/mg. The heavy-ion test sessions are summarized in Table 5.9 and Table 5.10.

Test Session	Energy [MeV/u]	LET [MeVcm^2/mg]	Duration [secs]	Flux [ions/cm^2/sec]	Fluence [ions/cm^2]
OCM	150	32.02	250	0.51 x 10^5	1.27 x 10^7
L1	150	32.02	330	0.46 x 10^5	1.53 x 10^7
L1	300	21.07	300	3.31 x 10^5	9.92 x 10^7

Table 5.9: Heavy ion static memory test sessions

Benchmark	Energy [MeV/u]	LET [MeVcm^2/mg]	Duration [secs]	Flux [ions/cm^2/sec]	Fluence [ions/cm^2]
All enabled	150	32.02	504	1.23 x 10^5	6.20 x 10^7
All enabled	300	21.07	228	3.14 x 10^5	7.16 x 10^7
L1 enabled	150	32.02	275	1.21 x 10^5	3.33 x 10^7
L1 enabled	300	21.07	178	1.51 x 10^5	2.69 x 10^7
L2 enabled	150	32.02	100	1.21 x 10^5	1.21 x 10^7
L2 enabled	300	21.07	178	3.56 x 10^5	6.34 x 10^7

Table 5.10: Heavy ion benchmarks test sessions

5.2 Reliability Analysis

To calculate the SEE sensitivity of the Zynq-7000 PS, both events in different SRAM memories, i.e., OCM, L1 and L2, and the execution of a suite of processor benchmarks for different cache configurations were studied.

5.2.1 Protons

5.2.1.1 Memory Tests

Table 5.11 presents the results of the static and dynamic memory tests for the OCM, the L1 data cache and the L2 cache for different energies. Table 5.12 and Figure 5.5 show the SEU cross sections calculated from these tests. To calculate the error bars, we made the same assumptions as in the Zynq-7000 PL cross sections, i.e., a Poisson distribution of the SEUs, a confidence level of 95% and an uncertainty on the measured fluence of 10% [48], [80].

Energy [MeV]	OCM static		OCM dynamic		L1 static		L2 dynamic	
	Fluence [p/cm^2]	Upsets	Fluence [p/cm^2]	Upsets	Fluence [p/cm^2]	Upsets	Fluence [p/cm^2]	Upsets
30	1.00x10^{10}	531	0.89x10^{10}	73	1.00x10^{10}	61	0.58x10^{10}	78
50	0.96x10^{10}	120	1.22x10^{10}	38	1.00x10^{10}	34	0.85x10^{10}	116
100	1.00x10^{10}	220	1.22x10^{10}	64	1.00x10^{10}	19	0.66x10^{10}	66
150	1.00x10^{10}	388	0.66x10^{10}	46	1.00x10^{10}	2	0.92x10^{10}	72
200	1.78x10^{10}	371	4.04x10^{10}	241	1.79x10^{10}	42	3.53x10^{10}	337

Table 5.11: Proton memory tests upsets

Energy [MeV]	OCM static [cm^2/bit]	OCM dynamic [cm^2/bit]	L1 static [cm^2/bit]	L2 dynamic [cm^2/bit]
30	2.529x10^{-14}	3.930x10^{-15}	2.322x10^{-14}	3.198x10^{-15}
50	0.598x10^{-14}	1.490x10^{-15}	1.296x10^{-14}	3.250x10^{-15}
100	1.048x10^{-14}	2.506x10^{-15}	0.723x10^{-14}	2.385x10^{-15}
150	1.843x10^{-14}	3.294x10^{-15}	0.076x10^{-14}	1.877x10^{-15}
200	0.992x10^{-14}	2.844x10^{-15}	0.894x10^{-14}	2.276x10^{-15}

Table 5.12: Proton memory tests cross section

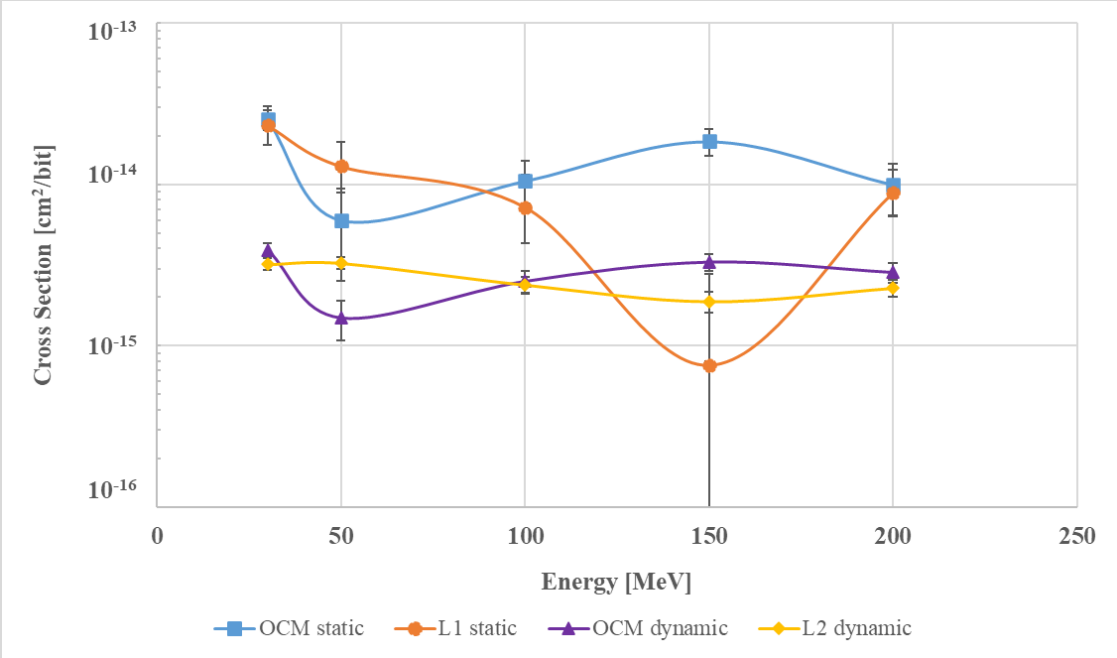


Figure 5.5: Proton memory tests cross section vs Energy

Analyzing and comparing the results with the literature, the following are remarked:

- Higher SEU rates have been observed for all the memories at the lower proton energy (i.e., 30 MeV); this observation is also true for the execution of processor benchmarks (i.e., higher SDC rates in 30 MeV) as described in the following subsection. This could be justified due to charges caused by low-energy direct ionizing and result in SEUs.
- The above observation is also verified by the literature [4], [22] for OCM dynamic, since their cross section is calculated to be 8.05 x 10^{-15} and 7 x 10^{-15} cm^2/bit respectively, for the low proton energy of 18 MeV, which is higher than the cross section at higher energies as presented in this thesis work.
- The OCM dynamic tests show a similar cross section with the work [130], except for the low energies (i.e., lower than 50 MeV). In this thesis work, the OCM dynamic cross section increases as energy decreases, while in [130], the OCM dynamic cross section decreases as energy decreases.
- The L2 dynamic tests show higher cross section compared to previous works (e.g., 250 MeV [136])
- The L1 static cross section at 150 MeV is significantly lower, at least an order of magnitude less, compared to the values observed at other energies. This discrepancy cannot be justified. Note that, the corresponding error bars exhibit

considerable extent, spanning over an order of magnitude, given the limited number of upsets at 150 MeV.

- The static tests show higher cross section in general than the dynamic tests, indicating that the lifetime of the data in the memories impacts their SEU vulnerability (i.e., during the static tests, the data in the OCM and caches remain for a longer time without being updated). Thus, a periodic cache invalidation mechanism may improve the data integrity in caches.
- It is the first time that radiation results have been reported for the L1 data cache across the full proton energy range.
- There is a constant offset between OCM static and OCM dynamic tests, which maybe due to upsets that occur in the FIFOs located in the AXI HP ports. The AXI HP port is used for scanning the OCM only for static tests.
- The proton cross sections of the Zynq-7000 PS memories are of the same order of magnitude as the corresponding cross sections of Zynq-7000 PL memories.

5.2.1.2 CPU Benchmarks

Table 5.13 shows the tests performed for all six processor benchmarks in total (CRC32, FFT, qsort, basicmath, SHA, MatrixMul) combined with four different cache configurations (i.e., all caches disabled, L1 enabled and L2 disabled, L1 disabled and L2 enabled, all caches enabled) at five proton energies (i.e., 30, 50, 100, 150 and 200 MeV).

Energy [MeV]	**All disabled**		**L1 enabled**		**L2 enabled**		**All enabled**	
	Fluence [p/cm^2]	**SDCs**	**Fluence [p/cm^2]**	**SDCs**	**Fluence [p/cm^2]**	**SDCs**	**Fluence [p/cm^2]**	**SDCs**
30	2.40×10^{10}	3	1.62×10^{10}	1	1.01×10^{10}	122	0.76×10^{10}	75
50	2.78×10^{10}	0	1.88×10^{10}	1	1.52×10^{10}	59	2.35×10^{10}	80
100	2.60×10^{10}	0	1.85×10^{10}	1	2.10×10^{10}	64	1.62×10^{10}	47
150	2.47×10^{10}	0	1.45×10^{10}	5	1.60×10^{10}	64	1.53×10^{10}	64
200	7.85×10^{10}	0	6.69×10^{10}	33	1.91×10^{10}	62	4.18×10^{10}	164

Table 5.13: Proton benchmark tests – SDCs per cache configuration

The results are first analyzed per energy and cache configuration. Table 5.14 and Figure 5.6 show the cross section per device for three cache configurations (i.e., L1 enabled only, L2 enabled only, all caches enabled); “all caches disabled” configuration is not depicted because only three SDCs were observed.

Energy [MeV]	L1 enabled [cm²]	L2 enabled [cm²]	All enabled [cm²]
30	0.62×10^{-10}	1.21×10^{-8}	9.85×10^{-9}
50	0.53×10^{-10}	3.87×10^{-9}	3.41×10^{-9}
100	0.54×10^{-10}	3.05×10^{-9}	2.91×10^{-9}
150	3.44×10^{-10}	4.00×10^{-9}	4.19×10^{-9}
200	4.93×10^{-10}	3.25×10^{-9}	3.93×10^{-9}

Table 5.14: Proton benchmarks cross section per device

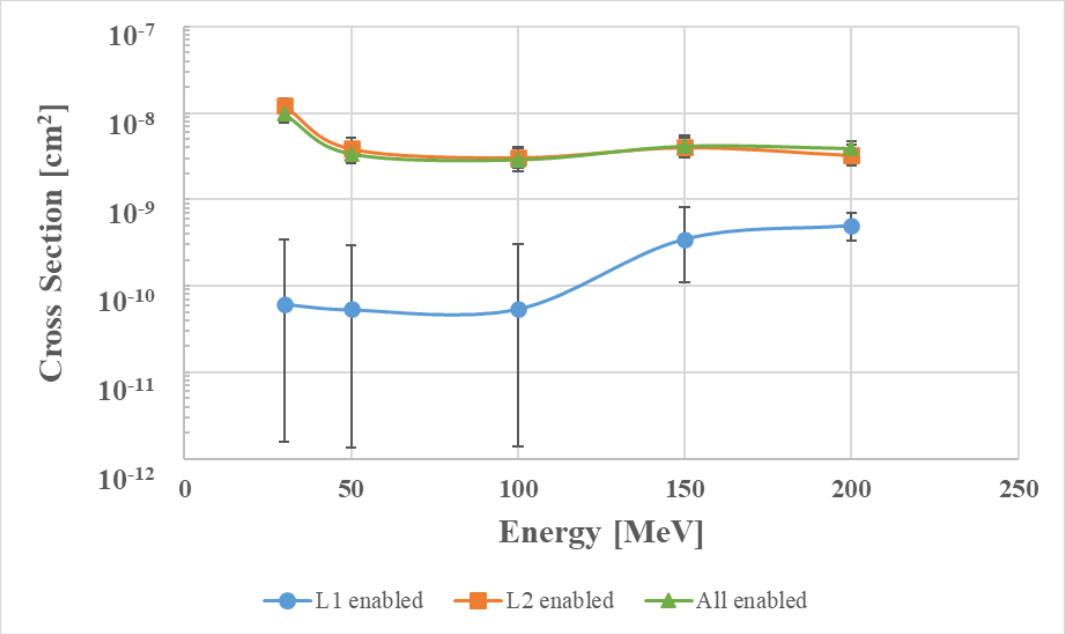


Figure 5.6: Proton benchmarks cross section per device vs Energy

Energy [MeV]	L1 enabled [cm²/bit]	L2 enabled [cm²/bit]
30	0.24×10^{-15}	2.89×10^{-15}
50	0.20×10^{-15}	0.92×10^{-15}
100	0.21×10^{-15}	0.73×10^{-15}
150	1.31×10^{-15}	0.96×10^{-15}
200	1.88×10^{-15}	0.78×10^{-15}

Table 5.15: Proton benchmarks cross section per cache bit

Table 5.15 and Figure 5.7 show the cross section per cache bit for the two cache configurations (L1 enabled only, L2 enabled only). Note that for the calculation of these cross sections, the errors have been divided by the number of L1 data cache bits and L2 cache bits, respectively; the total sizes, and not the active regions, of the caches were considered.

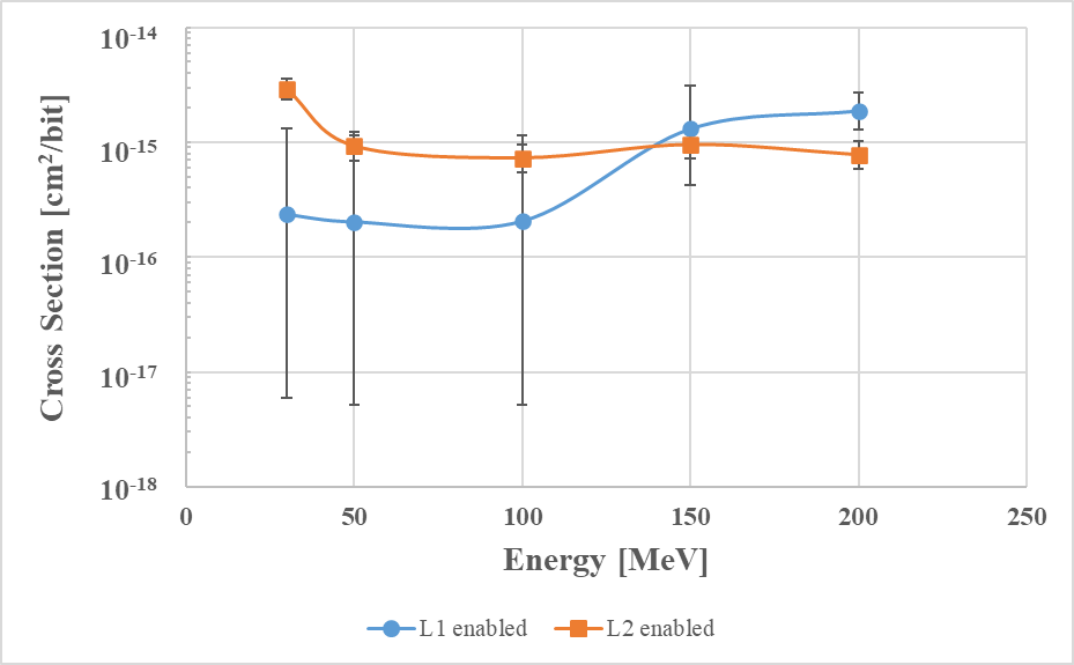


Figure 5.7: Proton benchmarks cross section per cache bit vs Energy

Following are comments on the SDC analysis of the processor benchmarks:

- Very few SDCs or no SDCs were observed for the cache configuration with both caches disabled. Additionally, very few SDCs were observed when only L1 is enabled. Remember that L1 has small capacity of 32 Kbytes.
- Significant number of SDCs and similar cross sections were observed, in the two cases that L2 is enabled ("L2 enabled only" and "all caches enabled"). Remember that L2 has a capacity of 512 Kbytes, 16 times larger than L1 capacity. Thus, it seems that the upsets in the L2 cache dominate the cross-section.
- The cross section per device is lower when only the L1 cache is enabled. In contrast, the cross sections per bit for the L1 and L2 caches are similar. Thus, L1 and L2 caches have the same probability of being affected per bit, but an error is much more likely to occur in L2, not because it is more vulnerable, but because it is larger.

- There is no error when both caches are disabled. This makes sense, since only DDR was used during these tests, and it was not irradiated.

Energy [MeV]	Benchmark	L2 enabled			All enabled		
		Fluence [p/cm^2]	SDCs	Cross section [cm^2]	Fluence [p/cm^2]	SDCs	Cross section [cm^2]
30	**CRC32**	6.372×10^{9}	43	6.75×10^{-9}	3.951×10^{9}	28	7.09×10^{-9}
50		7.766×10^{9}	17	2.19×10^{-9}	1.338×10^{10}	30	2.24×10^{-9}
100		1.372×10^{10}	21	1.53×10^{-9}	8.850×10^{9}	13	1.47×10^{-9}
150		1.032×10^{10}	15	1.45×10^{-9}	8.176×10^{9}	22	2.69×10^{-9}
200		1.334×10^{10}	27	2.02×10^{-9}	2.176×10^{10}	59	2.71×10^{-9}
30	**Qsort**	2.342×10^{9}	77	3.29×10^{-8}	1.667×10^{9}	42	2.52×10^{-8}
50		2.670×10^{9}	26	0.97×10^{-8}	5.327×10^{9}	49	0.92×10^{-8}
100		4.869×10^{9}	42	0.86×10^{-8}	3.575×10^{9}	33	0.92×10^{-8}
150		3.532×10^{9}	49	1.39×10^{-8}	3.560×10^{9}	42	1.18×10^{-8}
200		4.384×10^{9}	35	0.80×10^{-8}	8.813×10^{9}	101	1.15×10^{-8}

Table 5.16: Proton SDCs and cross section per benchamark (CRC32, Qsort)

Energy [MeV]	Benchmark	L2 enabled		All enabled		Benchmark	L2 enabled		All enabled	
		Fluence [p/cm^2]	SDCs	Fluence [p/cm^2]	SDCs		Fluence [p/cm^2]	SDCs	Fluence [p/cm^2]	SDCs
30	**FFT**	5.68×10^{8}	1	6.94×10^{8}	0	**SHA**	4.14×10^{8}	1	5.19×10^{8}	5
50		7.02×10^{8}	1	1.94×10^{9}	0		5.16×10^{8}	0	1.33×10^{9}	1
100		1.26×10^{9}	0	1.38×10^{9}	1		9.19×10^{8}	0	9.98×10^{8}	0
150		9.36×10^{8}	0	1.32×10^{9}	0		6.70×10^{8}	0	9.30×10^{8}	0
200		1.27×10^{9}	0	3.75×10^{9}	1		9.34×10^{8}	0	2.92×10^{9}	0
30	**basicmath**	1.32×10^{8}	0	3.45×10^{8}	0	**MatrixMul**	2.28×10^{8}	0	4.37×10^{8}	0
50		1.86×10^{8}	0	6.84×10^{8}	0		2.76×10^{8}	0	1.02×10^{9}	0
100		2.68×10^{8}	0	5.97×10^{8}	0		4.59×10^{8}	1	8.33×10^{8}	0
150		1.74×10^{8}	0	5.29×10^{8}	0		3.08×10^{8}	0	7.48×10^{8}	0
200		4.07×10^{8}	0	2.04×10^{9}	0		5.45×10^{8}	0	2.50×10^{9}	3

Table 5.17: Proton SDCs per benchmark (FFT, basicmath, SHA, MatrixMul)

The results are then analyzed by benchmark. Table 5.16 and Table 5.17 show the SDCs and the cross section per benchmark for two cache configurations (L2 enabled only and all caches enabled). "L1 enabled only" configuration is not presented because only a few SDCs were observed, and they are sparsely distributed across the benchmarks, resulting in few or no SDCs per benchmark.

The SDC cross sections for the CRC32 and Qsort benchmarks for two cache configurations are shown in Figure 5.8, since the vast majority of SDCs occur in these two benchmarks.

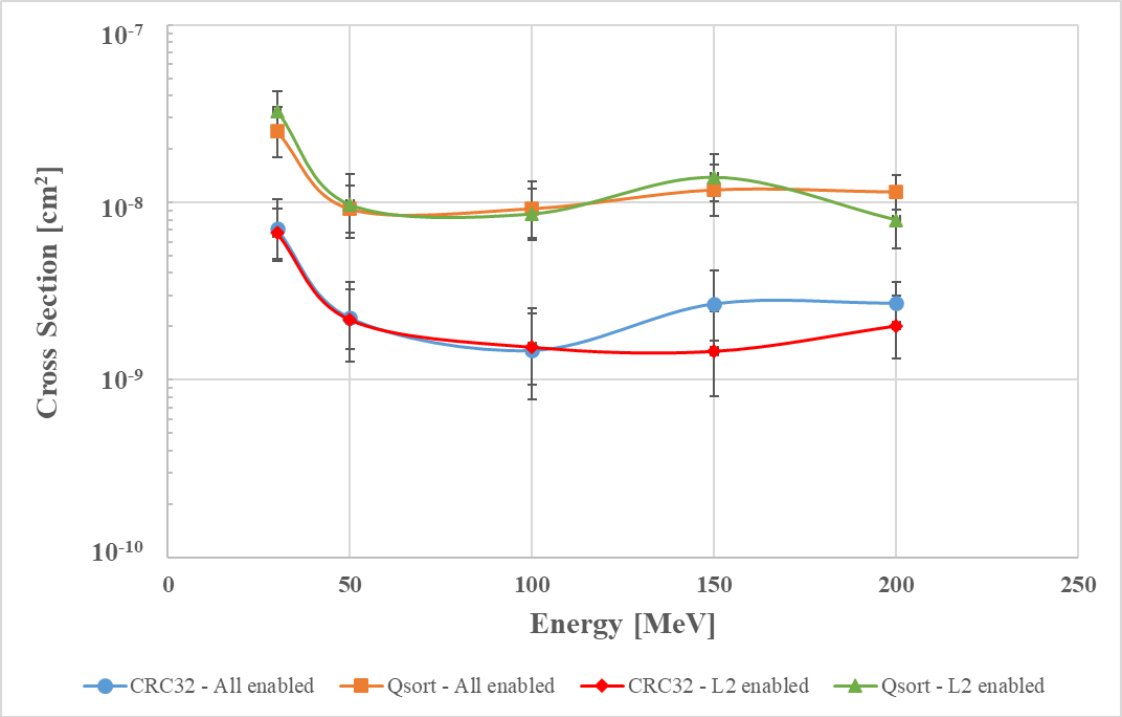


Figure 5.8: Proton SDC cross section for CRC32 & Qsort benchmarks vs Energy

From the analysis, we observe the following:

- Very few or no SDCs have been observed in cache configurations "all caches disabled" and "L1 enabled only". Thus, these events are not included in Table 5.16.
- None SDC in basicmath, and only a few SDCs have been observed in FFT, SHA and MatrixMul benchmarks (i.e., much less than CRC32 and QSort). The low SDC counts can be attributed to a) the benchmarks' low memory footprint (e.g., small data segment size) and thus low occupation in caches, and b) their short execution time, and thus test effective time (i.e., excluding the time to initialize each test), compared to CRC32 and Qsort.

- CRC32 and Qsort present a higher number of SDCs, while CRC32 has a lower cross-section than Qsort. This could be explained by the fact that the data segment of CRC32 (see Table 5.5) is larger than the L2 cache size, which results in frequent L2 cache replacements from DDR, while the data segment of Qsort fits into L2, resulting in less frequent L2 replacements. Thus, CRC32 has a lower SDC cross section than Qsort, despite its larger memory footprint.

5.2.1.3 System Crashes

The system crashes that were observed during the execution of the dynamic memory tests and the CPU benchmarks, which probably caused by SEFIs in the ARM SoC, are analyzed. Table 5.18 and Figure 5.9 show the total system crashes and their cross section per energy.

Energy [MeV]	Fluence [p/cm^2]	System crashes	Cross section [cm^2]
30	0.82×10^{11}	16	1.95×10^{-10}
50	1.12×10^{11}	5	0.45×10^{-10}
100	1.08×10^{11}	10	0.93×10^{-10}
150	0.92×10^{11}	9	0.98×10^{-10}
200	3.01×10^{11}	34	1.13×10^{-10}

Table 5.18: Proton system crashes

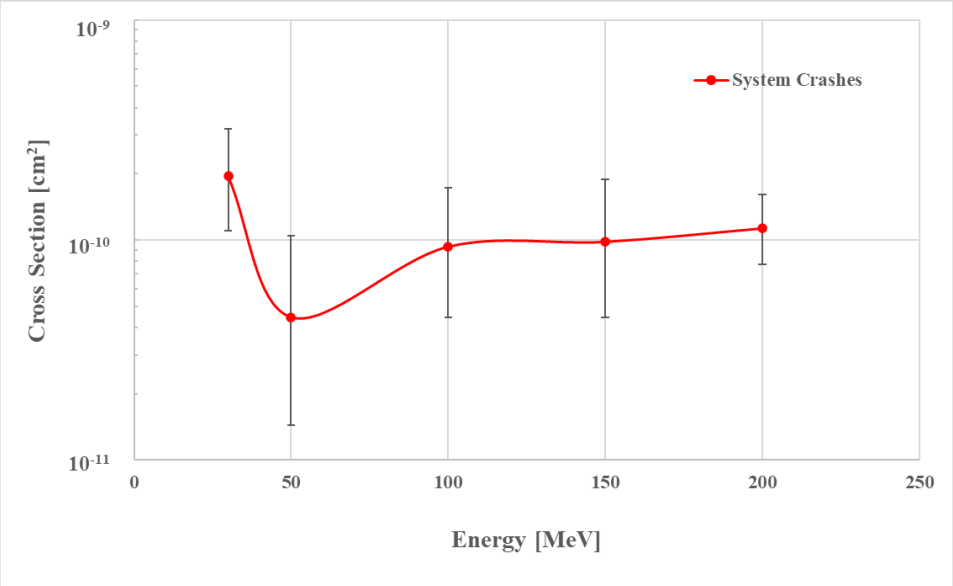


Figure 5.9: System crashes during CPU benchmarks and memory tests

5.2.2 Heavy Ions

5.2.2.1 Memory Tests

Table 5.19 and Table 5.20 present the results and the cross sections calculated for the static memory tests for the OCM and the L1 data cache for different heavy-ion U^{73+} energies and LETs.

Energy [MeV]	LET [MeVcm^2/mg]	Fluence [ions/cm^2]	Upsets	Cross section [cm^2/bit]
150	32.02	1.27 x 10^7	278	1.05 x 10^{-11}

Table 5.19: Heavy-ion OCM upsets and cross section

Energy [MeV]	LET [MeVcm^2/mg]	Fluence [ions/cm^2]	Upsets	Cross section [cm^2/bit]
150	32.02	1.53 x 10^7	567	1.41 x 10^{-10}
300	21.07	9.92 x 10^7	1789	0.69 x 10^{-10}

Table 5.20: Heavy-ion L1 cache upsets and cross section

Analyzing and comparing the results with the previous works, the following are observed:

- The L1 cross section increases slightly as the LET increases from 21.07 to 32.02 MeVcm^2/mg.
- The L1 static tests show similar cross sections compared to previous works (e.g., 3.45 x 10^{-10} cm^2/bit at 13.59 MeV·cm^2/mg [158]).
- The OCM static tests show similar cross section at 32.02 MeV·cm^2/mg compared to [159] which presents an OCM cross section 2.74 x 10^{-11} cm^2/bit at 25 MeV·cm2/mg.
- The L1 data cache cross section is an order of magnitude larger than the OCM cross section.
- It is the first time that radiation results have been reported for the OCM and L1 data cache for higher LETs.
- The heavy-ion cross section is higher than the proton cross section for all types of Zynq-7000 PS memories.

- The heavy-ion cross sections of the Zynq-7000 PS memories are of the same order of magnitude as the corresponding cross sections of Zynq-7000 PL memories.

5.2.2.2 CPU Benchmarks

Table 5.21 shows the results for all six processor benchmarks in total (CRC32, FFT, qsort, basicmath, SHA, MatrixMul), combined with three different cache configurations (i.e., all caches disabled, L1 enabled and L2 disabled, L1 disabled and L2 enabled) at two energies/LETs.

Energy [MeV]	LET [MeVcm^2/mg]	L1 enabled		L2 enabled		All enabled	
		Fluence [ions/cm^2]	SDCs	Fluence [ions/cm^2]	SDCs	Fluence [ions/cm^2]	SDCs
150	32.02	3.16 x 10^7	5	1.21 x 10^7	13	6.14 x 10^7	82
300	21.07	2.69 x 10^7	6	6.34 x 10^7	24	7.16 x 10^7	65

Table 5.21: Heavy-ion benchmark tests – SDCs per cache configuration

The results are analyzed per energy and cache configuration. Table 5.22 shows the cross section per device for three cache configurations (i.e., L1 enabled only, L2 enabled only, all caches enabled) and Table 5.23 shows the corresponding cross section per cache bit.

Energy [MeV]	LET [MeVcm^2/mg]	L1 enabled [cm^2]	L2 enabled [cm^2]	All enabled [cm^2]
150	32.02	1.58 x 10^{-7}	1.07 x 10^{-7}	1.34 x 10^{-6}
300	21.07	2.23 x 10^{-7}	3.79 x 10^{-7}	9.07 x 10^{-7}

Table 5.22: Heavy-ion benchmarks cross section

Energy [MeV]	LET [MeVcm^2/mg]	L1 enabled [cm^2/bit]	L2 enabled [cm^2/bit]
150	32.02	6.03 x 10^{-13}	2.56 x 10^{-13}
300	21.07	8.50 x 10^{-13}	0.90 x 10^{-13}

Table 5.23: Heavy-ion benchmarks cross section per cache bit

From the above analysis, we observed that the cross section per device is greater when all caches are enabled, while the cross sections per bit is greater when L1 cache is only enabled.

5.3 Upset Rate Estimation

The prediction of upset rate is a critical step in the design and reliability assessment of microelectronic systems intended for operation in space environments. Accurately forecasting SEE rates is crucial for the development of radiation-hardened components and for assessing mission risks. The process involves modeling both the external radiation environment and the internal response of electronic devices. Typically, this begins with the use of radiation environment models, such as models for trapped particles, cosmic rays, and solar proton fluxes. These models provide energetic particles flux spectra that vary significantly with orbital altitude, inclination, and solar activity, and they are then used as input for the SEE prediction tools.

The core of SEE rate estimation lies in the convolution of these particle fluxes with the device's sensitivity, characterized by its cross section as a function of LET or energy. The integral SEE rate is calculated by integrating the product of the particle LET/energy spectrum and the device cross section over all relevant LET/energy values.

Several tools and methodologies have been developed to facilitate SEE rate predictions. Among them, the OMERE (Outil de Modélisation de l'Environnement Radiatif Externe) software platform [141] is widely used in European space missions for its capability to simulate the space radiation environment and compute SEE rates based on user-specified device parameters and shielding configurations. It implements standard radiation environment models and provides mission-specific predictions that guide both the design of robust systems and the definition of radiation test campaigns. Developed by TRAD with support from CNES, the French space agency, OMERE has become a widely accepted tool in the European space industry for radiation analysis, particularly for SEE rate prediction and TID assessment.

5.3.1 Space Radiation Environment Modeling

The space radiation environment models include magnetically trapped radiation models, solar particle models, and cosmic rays models [46].

For Earth orbits, the conventional models used to characterize the fluxes of energetic particles in the radiation belts are the AE-8 and AP-8 models, which describe electron and proton fluxes, respectively [126], [148]. These models were originally developed by the Aerospace Corporation for NASA's National Space Science Data Center (NSSDC) at the Goddard Space Flight Center (GSFC), utilizing satellite data collected primarily during the 1960s and early 1970s. As they are derived from long-term averaged datasets,

AE-8 and AP-8 are best suited for estimating cumulative radiation effects on missions exceeding six months in duration. The AP-8 model covers proton energies ranging from 0.1 MeV to 400 MeV, whereas the AE-8 model addresses electrons within the 0.04 MeV to 7 MeV range.

Both models are available in variants corresponding to periods of solar minimum and solar maximum, denoted as AP8-Min/AE8-Min and AP8-Max/AE8-Max, respectively. The appropriate version is typically selected based on the solar activity phase during the mission. Given the difficulty of precisely quantifying the proportion of solar minimum versus maximum exposure for missions not entirely confined to either period, a conservative modeling strategy is adopted. Specifically, the AE8-Max model is applied for electron flux estimation, and the AP8-Min model is used for proton fluxes. This approach ensures that radiation exposure is not underestimated during periods of uncertain or mixed solar activity. In addition, the appropriate geomagnetic field model shall be selected when applying the AE8 and AP8 trapped radiation belt models. According to [46], AE8-Max and AP8-Min shall be used together with Jensen-Cain geomagnetic field model [82]. This ensures consistency and accuracy in radiation environment assessments, particularly for missions in LEO and GEO.

Standard models of solar particle events typically exclude electrons from consideration. However, for certain applications, particularly those involving internal charging, these electrons may play a significant role. In general, the fluxes and fluences of solar energetic particles are assumed to be isotropic. Nonetheless, anisotropies are known to occur, especially in near-Earth space where geomagnetic shielding can affect particle distributions. Anisotropic behavior is also observed during the initial phase of an event, when particles are guided along interplanetary magnetic field lines. Importantly, the orientation of the interplanetary magnetic field may vary and is not necessarily aligned with the Earth–Sun axis. Due to the limited availability of event-specific directional data, an isotropic distribution is generally adopted for modeling purposes.

The cumulative proton fluence from solar particle events over the course of a mission should be estimated using the emission of solar protons ESP model [157]. It is based on historical solar particle event data, typically from multiple solar cycles, e.g., about 40 years of geostationary operational environmental satellites (GOES) data. It statistically characterizes the distribution of event fluences and constructs cumulative probability distributions for different energy channels. It typically covers proton energies from about 1 MeV to several hundred MeV. The ESP model provides fluence spectra for various event sizes, from small to extremely large events. To model worst-case scenarios a confidence level is selected. The confidence level represents the probability that the actual solar proton fluence during a mission will not exceed the model's prediction. For instance, to avoid underestimating long-term damage from solar proton events for cost-sensitive commercial missions, a confidence level of 75 - 90% should be selected.

For characterizing peak fluxes of protons and other ions during solar particle events, the cosmic ray effects on micro-electronics CREME96 model is recommended [143]. It is based on the worst-case historical event data. It covers proton and heavy ions energies from about 1 MeV to several GeV. It models high-flux, short-duration environments needed for SEE analysis, such as the solar storms of August 1972, which were a historically powerful series of solar storms with intense to extreme solar flares, solar particle events, and geomagnetic storm components.

The ISO 15390 model [77] is an internationally recognized standard for estimating the galactic cosmic ray environment in near-Earth space. Developed by the International Organization for Standardization, the model provides parameterized energy spectra for a range of ion species, from protons and helium nuclei to heavy ions such as iron, based on empirical observations. A key feature of the model is its incorporation of solar modulation, whereby GCR fluxes are inversely related to solar activity. ISO 15390 is primarily valid at 1 astronomical unit (AU) outside the Earth's magnetosphere (i.e., equal to the average Earth-Sun distance) and is most applicable to long-duration space missions or interplanetary trajectories where GCR exposure is a concern. It is a widely used and accepted baseline reference for spacecraft design, radiation shielding analysis, and mission planning due to its simplicity, transparency, and international standardization.

Finally, geomagnetic shielding plays a critical role in determining whether charged particles, particularly ions, can penetrate the Earth's magnetic field and reach specific geographic locations. The cut-off energy refers to the minimum energy required for a particle to overcome this shielding effect, and its value varies with geographic position, altitude, and geomagnetic activity. Several established methodologies are available to compute these cut-off energies. One classical approach is Størmer's theory [2], which provides analytical solutions for particle trajectories in an ideal dipole magnetic field. For highly conservative design scenarios, it is also acceptable to assume no geomagnetic shielding at all, thereby using the unshielded interplanetary fluxes directly. The choice of method should align with the mission's orbital characteristics and the level of conservatism required in the radiation environment assessment.

5.3.2 SEE Rate

In order to ensure the reliability of electronics, it is essential to quantitatively estimate the SEE rate, i.e., the expected number of events per unit time for a given radiation environment and electronic component. SEE rate estimation is typically performed through a convolution of the incident LET spectrum with the device's cross-section response to LET. This relationship is commonly represented using a Weibull function, which is a widely used mathematical model in space radiation engineering to describe the sensitivity of an electronic device to SEEs as a function of the LET of incident particles. It is an empirical curve fit based on heavy ion test data, capturing the probability that a particle of a given LET will cause an SEE in a device.

The cumulative Weibull distribution used in SEE modeling is defined as:

$$\sigma(L) = \begin{cases} 0, & if\ L < L_0 \\ \sigma_{sat}(1 - e^{-\left(\frac{L-L_0}{W}\right)^S}), & if L \geq L_0 \end{cases} \quad \text{[Eq. 5.1]}$$

where L is the LET, σ(L) the cross section per device at LET L, L_0 is the LET threshold, i.e., the minimum LET required to trigger an SEE, σ_{sat} is the saturation cross section which indicates the maximum vulnerability of the device, i.e., the maximum value the cross-section approaches at high LET, W is the width parameter that describes how rapidly the cross section increases after L_0 and S the shape factor that determines the curvature of the transition.

When convolved with an LET spectrum ϕ(L), the Weibull-modeled cross section gives the integrated SEE rate:

$$R_{SEE} = \int_{L_0}^{\infty} \sigma(L)\phi(L)dL \quad \text{[Eq. 5.2]}$$

where σ(L) and ϕ(L) are the device cross section and the differential particle flux, respectively, as a function of LET.

The Weibull function, derived from heavy-ion test data, is commonly employed in proton SEE rate estimation through an indirect approach. Since protons typically cause SEEs via nuclear reactions that generate high-LET secondaries, the resulting secondary LET spectrum is convolved with the device's Weibull-modeled cross section to predict SEE rates.

5.3.3 Use Cases of SEE Rate Calculation

The SEE rate is highly dependent on both the space radiation environment and device specific parameters. In practice, shielding, orbit altitude, inclination, and solar activity must also be taken into account, as they affect the spectrum of particles reaching the device. Due to the complexity of these interactions and dependencies, dedicated simulation tools are employed to perform SEE rate estimations in a systematic and accurate manner. One such tool is OMERE [141], which provides an integrated framework for modeling the radiation environment, particle transport through shielding, and SEE rate prediction based on empirical device data. The following subsections detail the use of OMERE v.5.9.1 for SEE rate estimation of a Zynq-7000 device for three different orbits: a) a Low Earth Orbit (LEO) at an altitude of approximately 800 km used for Earth observation, b) a LEO at an altitude of approximately 400 km, where the International Space Station (ISS) operates, and c) a Medium Earth Orbit (MEO) at an altitude of 23,222 km, where the Galileo constellation operates.

5.3.3.1 LEO - Earth Observation Orbit

A LEO with an inclination of 98 degrees and an altitude of approximately 800 km is characteristic of a Sun-synchronous orbit (SSO). This specific configuration is widely utilized for Earth observation and remote sensing missions due to its unique orbital mechanics. An inclination near 98 degrees enables the satellite's orbital plane to precess at the same rate that Earth orbits the Sun. This synchronization ensures that the satellite passes over any given point on Earth's surface at the same local solar time, providing consistent lighting conditions for imaging purposes. At an altitude of 800 km, the satellite completes an orbit approximately every 100 minutes. This allows for multiple passes each day, facilitating frequent data collection and monitoring. The high inclination allows the satellite to pass over both the North and South Poles on each orbit, enabling global coverage over time.

The first step for SEE rate calculation using the OMERE tool involves defining the parameters of the orbit and mission duration, which determine the radiation environment. The orbit parameters include the orbit type, the altitude and inclination, which are used to derive trapped radiation fluxes. On the other hand, the mission duration influences cumulative fluence and long-term effects. In this case, an inclination of 98 degrees and an apogee and perigee of 800 km are chosen while the duration of the mission is set at 3 years.

To characterize the radiation environment, a suite of established models is employed. The trapped radiation environment is modeled using the AE-8 and AP-8 empirical models for electron and proton fluxes, respectively, with the AE8-Max and AP8-Min variants applied to represent bounding environmental conditions. The Jensen–Cain geomagnetic field model is adopted as the reference framework for trapped particle trajectories. For the assessment of solar energetic particle fluxes, the ESP statistical model is utilized, calibrated at an 80% confidence level to capture average conditions during solar events. To account for solar flare conditions, the CREME96 worst case one-day model is incorporated, providing conservative estimates of peak fluxes for both protons and heavy ions. Finally, cosmic ray exposure is evaluated using the ISO 15390 model, which defines the GCR environment under standard space weather conditions. For both solar particle and GCR assessments, magnetospheric cut-off energies are computed using Størmer’s theory, ensuring accurate consideration of geomagnetic shielding effects.

The third step is the LET spectrum calculation. The LET spectrum characterizes the distribution of incident particle flux as a function of LET in silicon, enabling accurate estimation of SEEs. The LET spectrum is computed based on predefined orbital parameters, shielding configurations, and radiation environment models, including trapped protons and electrons, solar particle events, and galactic cosmic rays, by integrating the contributions of various ion species over a range of energies. The resulting

LET distribution serves as a critical input for probabilistic SEE rate prediction through convolution with the device's cross section curve.

In OMERE, the LET spectrum is determined either from the previously specified mission environment configurations or from user-defined flux input. In this case, the spectrum is based on the mission environment settings generated by the tool. In addition, the shielding thickness shall be specified, in g/cm^2 or mm of Aluminium equivalent, around the electronic component. For this calculation a simple spherical shielding of 2.5 mm Aluminium is selected, which can provide moderate protection against radiation and particle impacts, depending on the context.

The calculated LET spectrum as a function of flux is depicted in Figure 5.10.

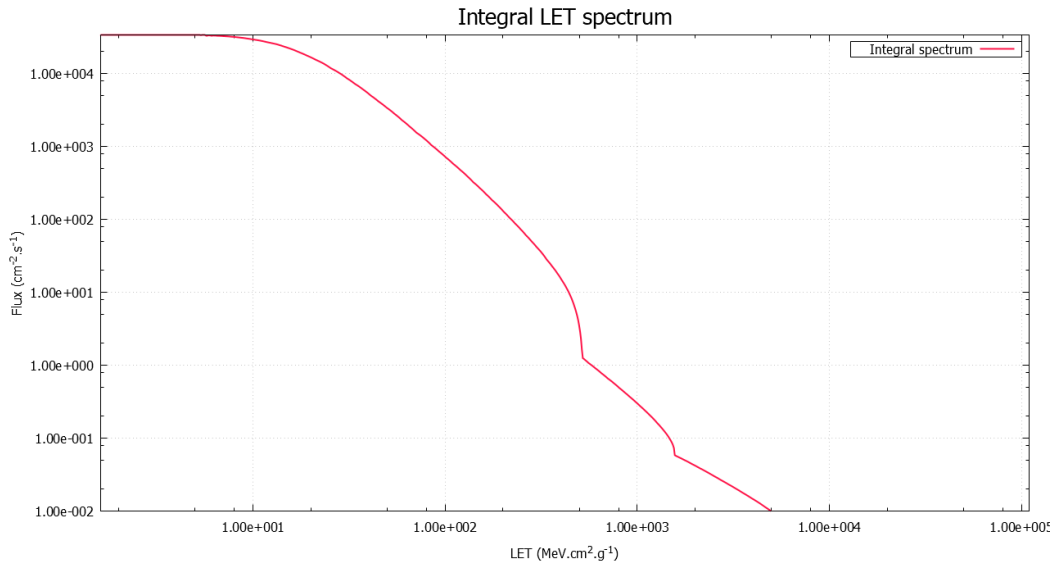


Figure 5.10: LEO (Earth observation orbit) - LET spectrum

The last step after modelling the LET spectrum is to determine the heavy ion and proton cross sections of the device for each LET. For each component of the device (i.e., CRAM, BRAM, etc.), the following must be defined:

- the number of sensitive cells and the cell size,
- the sensitive volume thickness
- the radiation environment models that are taken into account
- the experimental cross section obtained with heavy ion and proton testing using specific Weibull parameters

The OMERE tool calculates the SEE rate by performing a convolution of the LET spectrum with the device's sensitivity based on the predefined orbital parameters and radiation environment models determined in previous steps. The Zynq-7000 XC7Z045 SoC device was selected for the calculations. Table 5.24 below shows the number of sensitive cells and the saturation cross section σ_{sat} per component. For all component calculations, the Weibull fit parameters for heavy ion and proton cross section are set to: LET threshold = 2 MeV*cm^2/mg, W = 1.85 and S = 1.12, for heavy ions, as well as energy threshold = 30 MeV, W = 1.06 and S = 0.167, for protons. In addition, the cell size is set to 1 μm and the single shielding thickness is set to 2.5 mm.

Component	Sensitive cells	σ_{sat} [cm^2/device]	σ_{sat} [cm^2/device]
		Heavy ions	Protons
PL – CRAM SEUs	71,017,108	1.53 x 10^{-1}	4.15 x 10^{-7}
PL – BRAM SEUs	20,090,880	6.00 x 10^{-2}	1.70 x 10^{-7}
PL – FF SEUs	437,200	5.50 x 10^{-3}	5.37 x 10^{-9}
PL – SRL/DRAM SEUs	4,505,600	2.40 x 10^{-2}	5.90 x 10^{-8}
PS – OCM SEUs	2,097,152	2.20 x 10^{-5}	3.90 x 10^{-8}
PS – L1 SEUs	262,144	3.70 x 10^{-5}	2.40 x 10^{-9}
PS – SDCs L1 only	262,144	2.00 x 10^{-7}	4.93 x 10^{-10}
PS – SDCs L2 only	4,194,304	4.00 x 10^{-7}	4.00 x 10^{-9}
PS – SDCs L1&L2	4,456,448	1.34 x 10^{-6}	4.20 x 10^{-9}

Table 5.24: Sensitivity cells and saturation cross section per device

Before SEE rate estimation, the tool can plot the device cross section graphically as a function of LET (for heavy ions) or energy (for protons) using the above Weibull parameters. Figure 5.11 and Figure 5.12 present the graphical representations of the PL CRAM SEU cross section for heavy ions and protons, respectively, as an example.

The SEE rate calculation for each different PL and PS component is shown in Table 5.25. In addition, the participation of heavy ions and protons in the total rate per device per day is shown in detail. Two different types of rates are calculated, the in-flare rate and the out-of-flare rate. The in-flare rate encompasses the effects of all radiation environments, including solar flare events, whereas the out-of-flare rate measures the device’s upset rate during periods when no flaring activity is present. A solar flare is a sudden, intense burst of energy from the Sun. Because solar flares occupy only a small fraction of a mission's lifetime but can dominate the total number of radiation-induced upsets, separating the upset rates into flare and non-flare periods is a standard way to assess radiation performance and design appropriate fault-tolerance strategies.

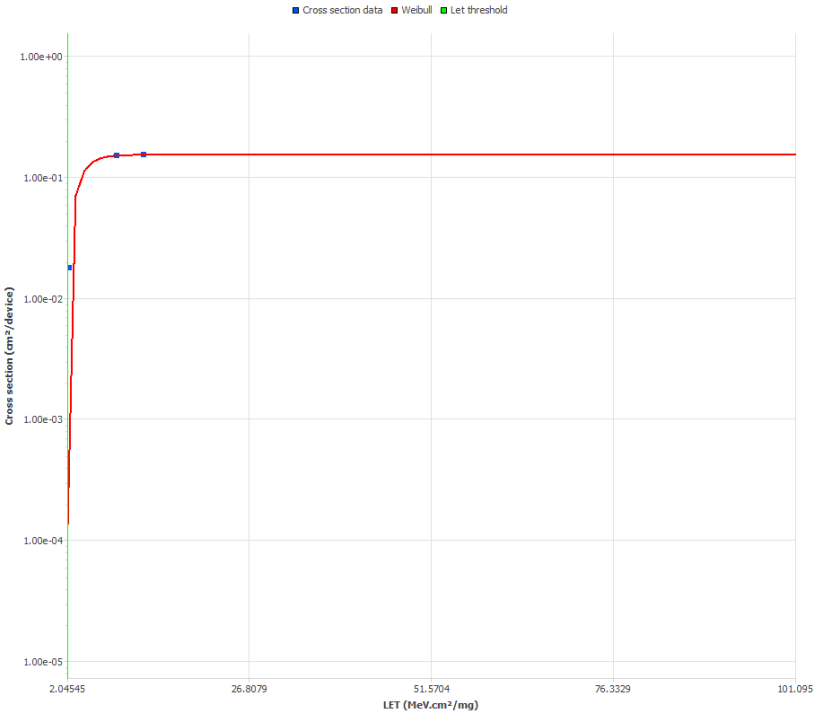


Figure 5.11: CRAM SEUs heavy ion cross section per device

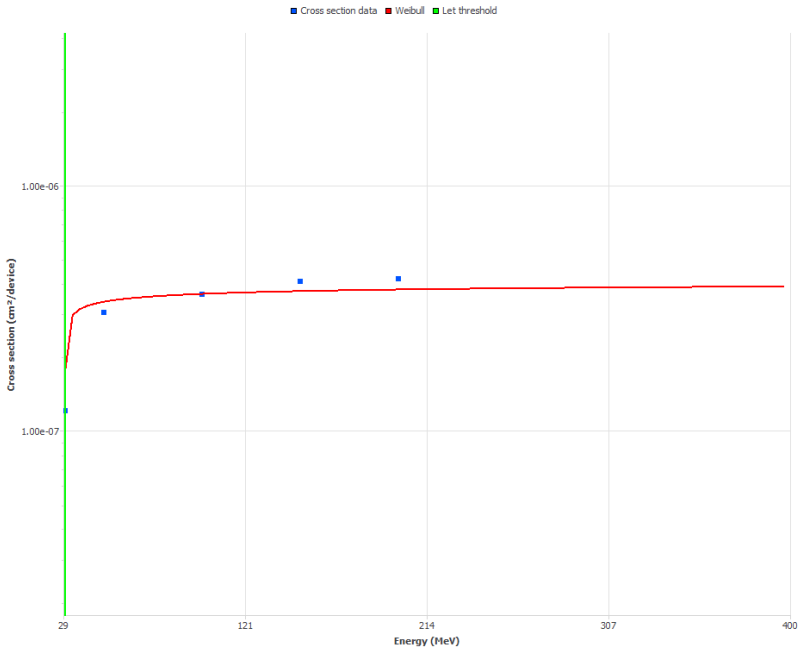


Figure 5.12: CRAM SEUs proton cross section per device

	In-flare rate (upsets per device per day)			Out-of-flare rate (upsets per device per day)		
	Heavy ions	Protons	Total	Heavy ions	Protons	Total
PL – CRAM SEUs	151	304	454	0.22	2.99	3.21
PL – BRAM SEUs	67	124	191	0.10	1.22	1.32
PL – FF SEUs	8.8	3.9	12.7	0.018	0.039	0.057
PL – SRL/DRAM SEUs	31.6	43.2	74.8	0.053	0.425	0.478
PS – OCM SEUs	$2.5x10^{-4}$	28.5	28.5	$3.2x10^{-7}$	0.281	0.281
PS – L1 SEUs	$7.0x10^{-3}$	1.76	1.76	$9.0x10^{-6}$	0.017	0.017
PS – SDCs L1 only	$1.6x10^{-7}$	0.36	0.36	$2.0x10^{-10}$	0.0036	0.0036
PS – SDCs L2 only	$4.0x10^{-8}$	2.93	2.93	$5.1x10^{-11}$	0.0288	0.0288
PS – SDCs L1&L2	$4.3x10^{-7}$	3.07	3.07	$5.4x10^{-10}$	0.0302	0.0302

Table 5.25: LEO (Earth observation orbit) SEE rate estimation

Analyzing the results, the following conclusions emerge:

- The impact of protons is significantly greater than that of heavy ions for the out-of-flare rate. In the PL memories, the error rate is about one order of magnitude higher, while in the PS components, it is several orders of magnitude higher, resulting in total rates that depend only on protons.
- The above also applies to the in-flare rate for both PS components: the errors caused by protons dominate the total error rates. On the contrary, in the PL memories, both protons and heavy ions have a significant contribution to the total error rates. This can be justified by the fact that protons dominate the particle population, often making up over 90% of solar flare particle emissions; however, heavy ions can also be significant, especially for PL components that have more sensitivity cells than PS, depending on the flare and solar conditions, since they can carry high energies due to their greater mass.
- In both PL and PS components, the in-flare upset rates are aproximately two orders of magnitude higher than the out-of-flare rates, for both heavy ions and protons. In other words, the flux of both heavy ions and protons increases by a factor of about 100 compared to non-flare.
- In this orbit, the protection of the PL memories and PS OCM with advanced SEE mitigation techniques must be prioritized, without this meaning that we do not need to deal with the remaining PS components, especially if a highly critical application runs on PS, since even a single error can cause the SW to crash.

- The maximum total upset rate in this orbit is observed in PL CRAM SEUs, 454 upsets per day, which is equal to approximately 1 upset every 3 mins. This is a relative low upset rate. Thus, using typical fault-tolerant techniques, the Zynq-7000 PL can remain operational, even under the solar activity conditions expected in the specific orbit.

5.3.3.2 LEO - ISS Orbit

The ISS operates in a LEO characterized by an altitude of approximately 400 km and an orbital inclination of 51.5 degrees. This orbital regime represents a crucial segment of near-Earth space, enabling both sustained human presence and a wide range of scientific and technological experiments. The ISS maintains a nearly circular orbit with an inclination of 51.5 degrees to balance global coverage, launch accessibility, and fuel efficiency. At 400 km, the orbital period is approximately 94 minutes, allowing the ISS to complete ~15.3 revolutions per day. This rapid orbit ensures frequent revisit times, which is advantageous for time-sensitive experiments and Earth monitoring. This altitude strikes a balance between scientific utility and crew safety, making it optimal for long-duration human habitation and science experiments that require real-time monitoring and regular resupply missions.

Compared to the aforementioned LEO at 800 km, which is closer to the inner radiation belt, LEO at 400 km is mostly below the inner belt, hence it sees less trapped radiation, exposing satellites to less proton radiation, but is still affected by anomalies such as SAA. Radiation spikes due to SAA occur since this orbit passes through the SAA frequently. Finally, an 800 km LEO has a higher particle sensitivity to solar particles due to its less magnetic shielding.

To calculate the LET spectrum and upset rate at this orbit, the same configurations for radiation environment models are used as before. The calculated LET spectrum as a function of flux is depicted in Figure 5.13, while the SEE rate calculation for each different PL and PS component is shown in Table 5.26.

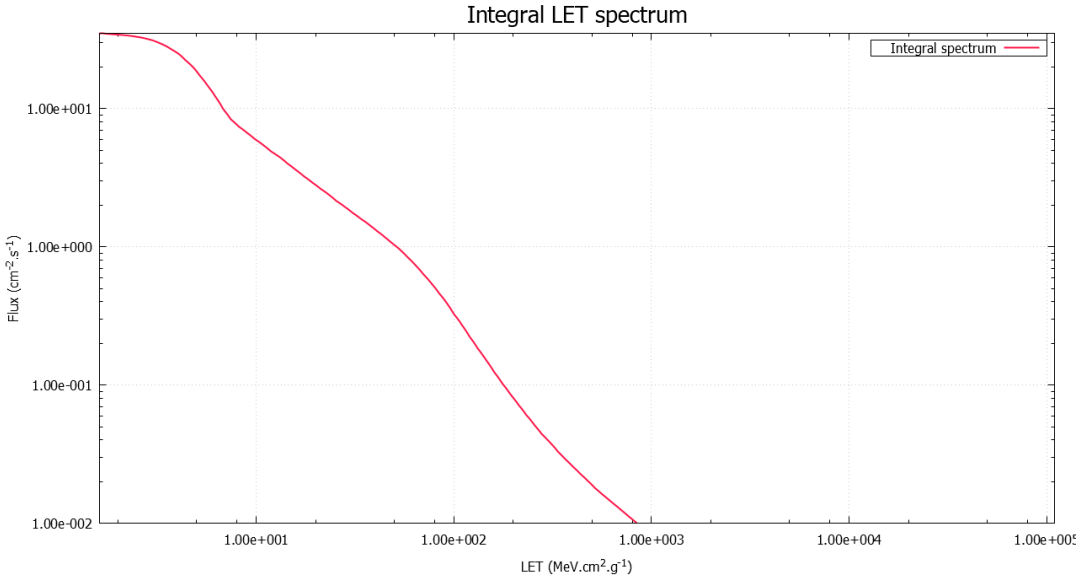


Figure 5.13: LEO (ISS orbit) - LET spectrum

	In-flare rate (upsets per device per day)			Out-of-flare rate (upsets per device per day)		
	Heavy ions	**Protons**	**Total**	**Heavy ions**	**Protons**	**Total**
PL – CRAM SEUs	13.2	1	14.2	0.026	0.356	0.382
PL – BRAM SEUs	5.7	0.41	6.11	0.012	0.146	0.158
PL – FF SEUs	0.687	0.013	0.70	0.0026	0.0046	0.0072
PL – SRL/DRAM SEUs	2.61	0.14	2.75	0.0069	0.0506	0.0575
PS – OCM SEUs	2.5×10^{-5}	0.095	0.095	3.5×10^{-8}	0.033	0.033
PS – L1 SEUs	0.0007	0.0058	0.0065	9.9×10^{-7}	0.002	0.002
PS – SDCs L1 only	1.6×10^{-8}	0.0012	0.0012	2.2×10^{-11}	0.0004	0.0004
PS – SDCs L2 only	4.0×10^{-9}	0.0097	0.0097	5.6×10^{-12}	0.0034	0.0034
PS – SDCs L1&L2	4.3×10^{-8}	0.01	0.01	5.9×10^{-11}	0.0036	0.0036

Table 5.26: LEO (ISS orbit) SEE rate estimation

Analyzing the upset rates, the following are remarked:

- The impact of protons is greater than that of heavy ions for the out-of-flare rate, as at the 800 km LEO.
- Protons contibute mainly to the total in-flare upset rates of the PS components, also as at the 800 km LEO. In contrast, the effect of heavy ions is greater than

protons for the PL components to the in-flare rate, which is not the case for the 800 km LEO.

- In both PL and PS components, the in-flare upset rates are at least two orders of magnitude higher than the out-of-flare rates for heavy ions, but for protons, the in-flare and out-of-flare upset rates are approximaely of the same order of magnitude, since this orbit has low sensitivity to solar proton particles, i.e., at 400 km, protons are largely shielded by atmosphere and magnetic field, while at 800 km, reduced shielding allows protons to dominate over heavy ions.
- The Out-of-flare rates (for heavy ions and protons) and the heavy ion in-flare rates are one order of magnitude lower compared to LEO Earth observation orbit. On the other hand, the proton in-flare rates differ by two orders of magnitude, which also confirms the low sensitivity to solar proton particles of the ISS orbit.
- The maximum total upset rate in this orbit is also observed in the PL CRAM, as Earth Observation Orbit and it is equal to approximately 1 upset every 1.5 hours.
- In general, this orbit does not have a high error rate even in solar flare conditions, which is why it is used for human spaceflight, including ISS. Thus, simple mitigation techniques, as described in section 3.3, may be sufficient for the smooth operation of the device.

5.3.3.3 MEO - Galileo Constellation Orbit

The Galileo global navigation satellite system (GNSS), developed by the European Union in cooperation with ESA, operates in a precisely defined MEO configuration. The system comprises a constellation of 24 satellites positioned at a nominal altitude of 23,222 km and an orbital inclination of 56 degrees. This configuration is a deliberate design choice that balances global coverage, system interoperability, and optimized signal performance for users within and beyond Europe. The orbital period at this altitude is approximately 14 hours, ensuring that satellites revisit the same locations at the same local times every few days, which is critical for GNSS signal predictability and coverage modeling.

Compared to LEO orbits, Galileo's MEO orbit is within the outer radiation belt with heightened sensitivity to fluctuations in solar activity. Thus, the cost of increased radiation exposure is significant requiring robust shielding and hardened electronics.

The LET spectrum as a function of flux and the SEE rate for each different PL and PS component, as calculated by OMERE using the same radiation models as for the LEO orbits, are shown in Figure 5.14 and Table 5.27 respectively.

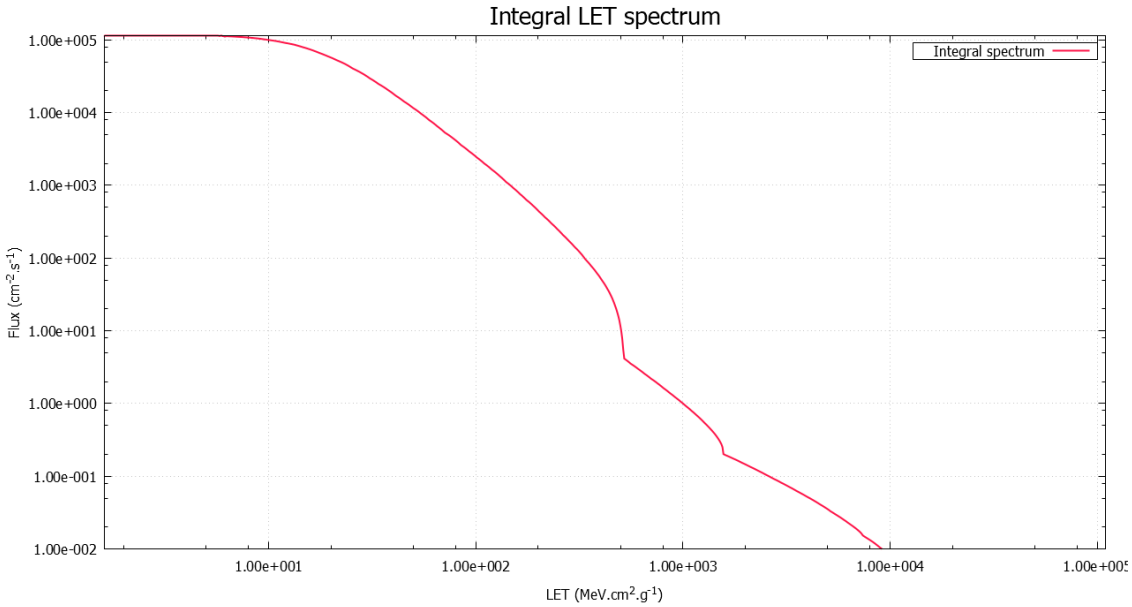


Figure 5.14: MEO (Galileo orbit) - LET spectrum

	In-flare rate (upsets per device per day)			Out-of-flare rate (upsets per device per day)		
	Heavy ions	**Protons**	**Total**	**Heavy ions**	**Protons**	**Total**
PL – CRAM SEUs	539	1010	1549	0.801	0.135	0.936
PL – BRAM SEUs	237	412	649	0.369	0.055	0.424
PL – FF SEUs	31	13	44	0.062	0.002	0.064
PL – SRL/DRAM SEUs	112	143	254	0.191	0.019	0.210
PS – OCM SEUs	9.3×10^{-4}	94.5	94.5	1.2×10^{-6}	0.013	0.013
PS – L1 SEUs	0.03	5.81	5.84	0.3×10^{-4}	7.8×10^{-4}	8.1×10^{-4}
PS – SDCs L1 only	6.0×10^{-7}	1.19	1.19	7.6×10^{-10}	0.00016	0.00016
PS – SDCs L2 only	1.5×10^{-7}	9.69	9.69	1.9×10^{-10}	0.0013	0.0013
PS – SDCs L1&L2	1.6×10^{-6}	10.2	10.2	2.0×10^{-9}	0.00137	0.00137

Table 5.27: MEO (Galileo orbit) SEE rate estimation

Comparing the MEO upset rates with those of LEOs, the following are observed:

- The out-of-flare rates are greater than Earth Observation LEO rates and less than ISS LEO rates for the PL components, while the PS out-of-flare rates are less than both LEO orbits.
- In both PL and PS components, the in-flare upset rates are about two orders of magnitude higher than the out-of-flare rates for heavy ions, while for protons, the

in-flare rates are about three orders of magnitude higher than the out-of-flare rates. In MEO, protons are generally the dominant source of radiation-induced upsets due to their abundance from the trapped particle environment (i.e., inner Van Allen belt). However, heavy ions, while less frequent, can cause more severe SEE, especially in sensitive or high-reliability systems.

- In MEO orbit, the proton out-of-flare rates are two order of magnitude lower than those calculated for the LEO Earth observation orbit. The in-flare rates (for heavy ions and protons) and the heavy ion out-of-flare rates are of the same order of magnitude for both MEO and LEO Earth observation orbits.
- In addition, comparing the upset rates of this orbit with the two other LEO orbits, the in-flare total rates are three times higher than the rates in the Earth Observation orbit, while they are 100 times higher than the rates in the ISS orbit. Although out-of-flare rates are low, in-flare rates are too high in this orbit, because it lies outside Earth's protective magnetic shield. In this orbit, a solar flare causes an immediate and massive spike in component failures, compared to the nominal baseline during non-flare periods.
- The in-flare rates are greater than both LEO orbits, starting from 1 upset per day and reaching up to 1549 upsets per day for the PL CRAM. Since this CRAM upset rate means about 1 upset per device per minute during solar flares, a sophisticated mitigation method is recommended, as the one presented in chapter 6, that improves error correction while keeping error correction latency and hardware costs low.

6 Configuration Memory Scrubbing

Though modern COTS SRAM-based FPGAs have penetrated the space market in the last decades, their vulnerability to radiation effects – particularly the susceptibility of their configuration memory to SEUs caused by the space ionizing radiation – imposes a significant barrier to their use in high-reliability space applications. Thus, it is crucial to study efficient hardening techniques to ensure their reliable operation in space.

Several fault tolerance methods have been applied to COTS SRAM-based FPGA devices. Such methods, as outlined in section 3.3, involve redundancy techniques, i.e., resource replication to perform the same task in parallel, to detect the error along with majority voting circuitry to choose the correct result. Memory scrubbing is also employed to correct memory upsets and prevent fault accumulation by reading the current memory state and writing back the correct values, typically stored in an external rad-hard memory. Additionally, error correction codes (ECCs), i.e., parity data, are used in addition to user data to detect and correct errors within memories. Most mitigation approaches proposed so far rely on hardware redundancy techniques, often combined with configuration memory scrubbing to prevent the accumulation of faults.

Most SRAM-based FPGA families, including Zynq-7000, incorporate ECCs into their configuration frames to facilitate the scrubbing mechanism. However, these frame-level ECCs usually have single-error correction double-error detection (SECDED) capabilities, as these COTS devices are primarily intended for terrestrial applications where MBUs are not a real threat. Consequently, these ECCs are insufficient for correcting MBUs, i.e., a single event affects two or more memory cells within a single configuration frame, which are not uncommon when the Zynq-7000 devices operate on a space radiation environment, as demonstrated by our radiation experiments presented in chapter 4.

SEM IP [12] is the hardware solution suggested by AMD to mitigate soft errors and, thus, protect the FPGA designs from radiation-induced faults. Exploiting the embedded SECDED capabilities of the AMD Xilinx FPGAs, the SEM IP can correct single-bit errors in the configuration memory frames when it operates in repair mode. It uses the built-in frame-level ECC syndrome to identify the exact location of the error in a frame. Thus, the SEM IP in repair mode guarantees the correction of all SBU events.

Additionally, when it is configured to operate in enhanced repair mode, in addition to the ECC code, it uses a frame-level CRC scheme capable of correcting two-bit adjacent errors in the same frame. Thus, the SEM IP in enhanced repair mode guarantees the correction of all SBU events and all double-bit adjacent upset events. In both cases, the SEM IP reads the frame containing the error, inverts the relevant bit(s), and writes back the frame.

The hybrid scrubbing method discussed in [1] and [132] combines the aforementioned internal ECC-based scrubbing mechanism with an external scrubber. The internal scrubber continuously monitors the configuration frames, correcting SBUs and detecting MBUs. The external scrubbing mechanism communicates with the internal scrubber to identify frames with MBUs, retrieves the correct frame data from an external storage medium, and rewrites these frames. This approach relies on an external radiation-hardened memory to store the complete golden configuration bitstream. This extra memory, which is accessed only when multibit errors occur, increases the memory storage requirements, the error correction latency and the system cost.

Some researchers have suggested frame-level redundant configuration scrubbing techniques to enhance FPGA correction capabilities for multibit errors [139], [56]. The redundancy techniques involve replicating the design, creating multiple copies of the configuration frames, and using majority voting among these redundant frames to correct upsets. In [139], a TMR circuit is created using a customized design flow, ensuring that the configuration frames in each TMR domain are identical, meaning each domain is synthesized, placed, and routed in the same way. Only the configuration frames of the TMR circuit are scanned, and the majority vote is performed among the three replicated frames during each scrubbing cycle to produce a fault-free frame. In [56], the configuration frames of the original design are replicated N times to unused frames after the initial configuration of the device; this replication is performed out of the FPGA layout tools. Not all replicas need to be active; only the original circuit must be operational, thereby reducing power consumption compared to [139]. A scrubber performs a detection and correction cycle by checking the consistency of the corresponding frames of each redundant sequence. At each iteration, the scrubber reads the redundant frames and fixes the incorrect ones. The advantage of these methods is that they do not require extra storage for the golden bitstream, while their disadvantage is that they are only applicable to designs with low utilization of the programmable resources and require regions with identical programmable resources to host the redundant configurations.

On the other hand, other researchers have focused on designing more advanced ECC schemes for multibit error correction. Their basic idea is to treat FPGA configuration memory as a two-dimensional (2-D) array and use sophisticated codes, such as 2-D hamming codes or bit-interleaved parity codes combined with erasure codes. Specifically, in [112], each configuration frame is treated as a 2-D memory window, utilizing a 2-D hamming code for protection. This method has been shown to effectively

correct MBUs within a configuration frame using a set of row and column directional error correction trials, capable of correcting up to ten errors in a 1024-bit frame organized as a 32x32-bit window. The performance of this ECC depends on the size of the 2-D window and the number of iterations; smaller windows and more iterations enhance the ECC's effectiveness but increase memory overhead and correction latency, respectively. In [147], each configuration frame is divided into a fixed number of subframes using bit interleaving. A hamming code is generated for each subframe, with the non-essential bits of the frame used to store its parity bits. This bit-interleaved hamming scheme enhances ECC performance. An external memory is not required if all ECC bits can be embedded within the configuration frames, depending on the availability of non-essential bits. However, if the hamming codes cannot be embedded, the memory overhead and circuit complexity increase significantly. In [42], a parity scheme known as stepped parity is introduced for memory arrays. In this scheme, each memory cell is linked to multiple parity bits, ensuring that the most common MBU patterns are detectable by at least one parity bit. This method provides higher MBU coverage than the virtually interleaved parity while it imposes less memory overhead. Finally, in [120], [43], [98], and [99], interleaved parity codes are combined with erasure codes to protect the FPGA configuration memory from MBUs. In [120], a virtually interleaved 2-D (I2D) parity scheme is presented, where each horizontal or vertical parity bit is the XOR of the bits in multiple rows or columns. This I2D parity scheme reduces memory overhead compared to a typical 2-D parity scheme. MBU detection coverage is enhanced by increasing the interleaving distances both horizontally and vertically. A parity-based erasure code is employed with one redundant block containing the parity of data stored in a cluster of blocks. In [43], the I2D parity scheme is extended to three dimensions (I3D), including horizontal, vertical, and diagonal parity bits, improving MBU coverage with minimal additional hardware overhead. In [98], the authors introduce a parity-based ECC called the interleaved modified matrix code (IMMC) [23], along with a type of erasure code using EVENODD coding [29], to enhance error correction capabilities and reduce area overhead compared to [43], while in [99], the same authors improve the IMMC scheme by combining the selective bit placement technique from [123] with frame interleaving, increasing the coverage of clustered errors in adjacent configuration frames. All these approaches do not account for the ECC schemes already embedded in modern SRAM-based FPGAs and thus require additional hardware resources and memory storage.

In this thesis, a scrubbing approach is presented that improves the multiple-bit error correction capabilities of the on-chip ECC schemes while it keeps low the error correction latency and the hardware cost. To achieve this, the concept of hybrid scrubbing [1], [132] is combined with the concept of 2-D coding [42], [43], [98], [99], [112], [120] and [147], building a mixed 2-D coding technique. The proposed approach utilizes the embedded ECC scheme as well as an interframe, interleaved parity code creating a 2-D EDC technique. Assuming a 2-D arrangement of the FPGA configuration memory, the on-chip ECC detects SBUs and MBUs per frame while the combination of the ECC code

and the parity code, in the vertical and horizontal directions, respectively, manages to correct the vast majority of SBUs and MBUs in one or more frames.

The mixed 2-D coding technique can be applied in all the SRAM FPGAs featuring on-chip ECC, such as the AMD Xilinx FPGA (e.g., Virtex-5, Virtex-6, 7-Series and UtraScale) and Intel FPGA families (e.g., Stratix and Arria series). However, since this thesis was motivated by the need to facilitate the deployment of Zynq-7000 devices in radiation environments, the technique was adapted and validated for this FPGA family. Based on the detailed characterization of Zynq-7000, as presented in chapter 4, the mixed 2-D EDC approach was designed, developed, validated under irradiation and compared with the state-of-the-art. The efficiency of the proposed approach was demonstrated through heavy ion radiation experiments, where all the SBUs and MBUs observed in the configuration memory were corrected. In addition, to compare it with the state-of-the-art, we implemented two versions of the scrubbing scheme for the Zynq-7000 FPGA architecture. The first version is an external scrubber comprising a microcontroller that runs the correction algorithm and communicates with the Zynq-7000 via the JTAG port. The second version is an internal scrubber that fully integrates the proposed solution on-chip using the ICAP configuration port.

6.1 Error Detection and Correction Mechanism

The proposed mitigation technique combines two coding schemes: the on-chip ECC, which is used to detect the upsets that occur in the configuration frames and the external error correction algorithm, which undertakes to correct all SBUs and most MBUs. The former is based on built-in ECC mechanisms, for example, the built-in Readback CRC supported by the majority of the AMD Xilinx FPGA families (e.g., Virtex-5, Virtex-6, 7-Series and UltraScale), and achieves fast error detection without extra cost. In contrast, the latter monitors the internal ECC's syndromes and combines them with externally stored parity data for error correction. The main advantages of the proposed approach are that a) it removes the requirement of externally holding the golden bitstream; only the parity bits need to be stored in a separate memory, whether internal or external, b) it ensures that the worst-case error correction latency remains bounded, as the correction process fully depends on the 2-D coding scheme, eliminating the need to upload the golden copy, and c) it accurately distinguishes odd-numbered MBUs from SBUs, thereby preventing incorrect corrections that occur when the internal scrubber of the hybrid approach in [132] and [1] mistakenly repairs an odd-numbered MBU interpreted as an SBU due to limitations of the built-in ECC mechanism, and thus its external scrubber needs extra time to identify this situation and scrub the entire frame.

6.1.1 Built-in ECC Feature

Newer FPGA families have progressively enhanced their upset mitigation mechanisms with each subsequent release. For example, the AMD Xilinx mitigation mechanisms have evolved over generations, starting with the Virtex-4 series [21] and reaching the Versal series [20]. Key features include an individual ECC word for each configuration frame and a device-wide CRC mechanism. The individual frame ECCs allow for SBU correction and two-bit upset detection, while the CRC identifies nearly all MBUs. A crucial aspect of the ECC and CRC encodings is the internal scan hardware engine that actively employs them, which is called Readback CRC. The Readback CRC continuously scans all configuration frame data, performing ECC checks for each frame and a CRC comparison for the entire device. Utilizing and interpreting the output from this internal scan demands careful attention to specific details.

Each frame includes an embedded ECC word used by the Readback CRC to provide SECDED functionality. The Readback CRC calculates the ECC for each frame, generating a syndrome value from all the words in that frame. In the case of Zynq-7000, each frame consists of 100 data words plus 1 ECC word, totaling 101 words with 32 bits each, amounting to 3,232 bits per frame. The 51st word in these frames is the ECC word. Given that the Zynq-7000 has a 13-bit ECC syndrome, there are 2^{13} (8192) possible syndrome values or 8191 unique error representations. Since the syndrome can pinpoint the exact location of an SBU, 3,232 of these values must indicate unique single-bit locations. As the code has SECDED capability, the two-bit upsets can be detected but not corrected. All odd-numbered MBUs will be detected by the ECC, but the syndrome value will not indicate any specific upset bits. Finally, most even-numbered upsets - with multiplicity higher than two bits - will also be detected, although in rare cases, e.g., when upsets satisfy ECC, detection is not guaranteed. Bit [12] of the syndrome serves as the parity bit, used to determine whether the detected upset is odd- or even-numbered. If this bit is the only upset bit, a parity error has occurred. The possible combinations of the other syndrome bits and the parity bit are summarized in Table 6.1.

ECC syndrome		**Type of upset**
bit [12]	**bit [11:0]**	
0	0	ECC satisfied - No error or even-numbered MBU
1	≠ 0	Odd-numbered MBU In the case of SBU, the syndrome value indicates the upset location
0	≠ 0	Even-numbered MBU
1	0	Parity error - upset in the ECC word

Table 6.1: Zynq-7000 ECC syndrome

The syndrome identifies the exact location of the upset, allowing the repair mechanism, such as the Readback CRC, to correct it. However, for an odd-numbered MBU, the syndrome is unlikely to point to one of the actual error bits. Instead, it indicates the bit that, when flipped, would satisfy the ECC and yield an all-zero syndrome. Thus, if a three-bit upset occurs and the bit indicated by the syndrome is flipped, it will result in a four-bit upset, and the ECC will likely fail to detect the error, as it now appears satisfied. In such cases, more advanced mitigation techniques are required to handle these upsets. The FRAME_ECCE2, which is described in subsection 2.3.2.2, is a dedicated hardware primitive that monitors and outputs the built-in Readback CRC error detection and correction signals (Table 2.4) that indicate the status of its circuitry.

The Readback CRC has the four operating modes described below:

- Continue mode: If an ECC error is detected in the current frame, the scan continues to the next frame while asserting the corresponding error signals (i.e., SYNDROMEVALID, SYNDROME, ECCERROR, etc.) along with the CRCERROR regardless of the error type (Table 2.4). However, no correction of errors is performed.
- Halt mode: If an ECC error is detected in the current frame, readback is stopped, and the corresponding error signals are asserted. The CRCERROR will also be asserted, regardless of the error type detected.
- Correct and halt mode: This mode is similar to the halt mode, but in addition, SBUs are corrected.
- Correct and continue mode: If an ECC error is detected, the corresponding error signals are asserted, SBUs are corrected without asserting CRCERROR, and readback restarts from the beginning of the device to continue scanning frames.

Each mode has a different behaviour. The selection of the appropriate mode depends on the operating conditions and the reliability requirements of the system. For example, in the case of a terrestrial radiation environment, where SBUs occur at a slow rate, the correct and continue mode is the suitable choice to handle a single upset at a time. On the other hand, in environments where multi-bit upsets may occur due to either accumulated or single events, the continue is more appropriate than the correct and continue mode. In this mode, the Readback CRC only detects upsets and another mechanism takes care of correcting the errors. This idea is adopted by the scrubbing approach of this thesis, where the error correction algorithm, described in the following subsection, takes over this job.

6.1.2 Error Correction Algorithm

The main concepts of the proposed error correction algorithm are depicted in Figure 6.1, i.e., the 2-D coding scheme and the interleaved parity code. Let's consider the FPGA configuration memory as a 2-D matrix, where the configuration frames (green cells in

Figure 6.1) represent the columns of the matrix. The algorithm relies on a 2-D coding scheme consisting of:

- the on-chip frame-level ECC code (vertical direction). Assuming that it is a SECDED code, it can detect all SBUs and most MBUs per configuration frame (column). In the case of SBUs, the syndrome indicates the precise location of the errors within the frame; otherwise, it just shows the existence of errors.
- an interframe, interleaved parity code (horizontal direction). The configuration frames are divided into groups, as in [43]. Each group G contains N consecutive frames, R_0, R_1, … R_{N-1}, which are further divided into two or more subgroups in an interleaved manner. In the example of Figure 6.1, the group is divided into two subgroups, $G_A = \{R_0, R_2, R_4, \ldots, R_{N-2}\}$, i.e., even-numbered frames and $G_B = \{R_1, R_3, R_5, \ldots, R_{N-1}\}$, i.e., odd-numbered frames. The data of each bit location of the frames of a subgroup (row) are XORed to calculate a parity bit, thus all the frames (columns) of each subgroup produce the two parity frames, P_A and P_B (orange cells in Figure 6.1). For example, $P_A = R_0 \oplus R_2 \oplus \ldots \oplus R_{N-2}$ and $P_B = R_1 \oplus R_3 \oplus \ldots \oplus R_{N-1}$.

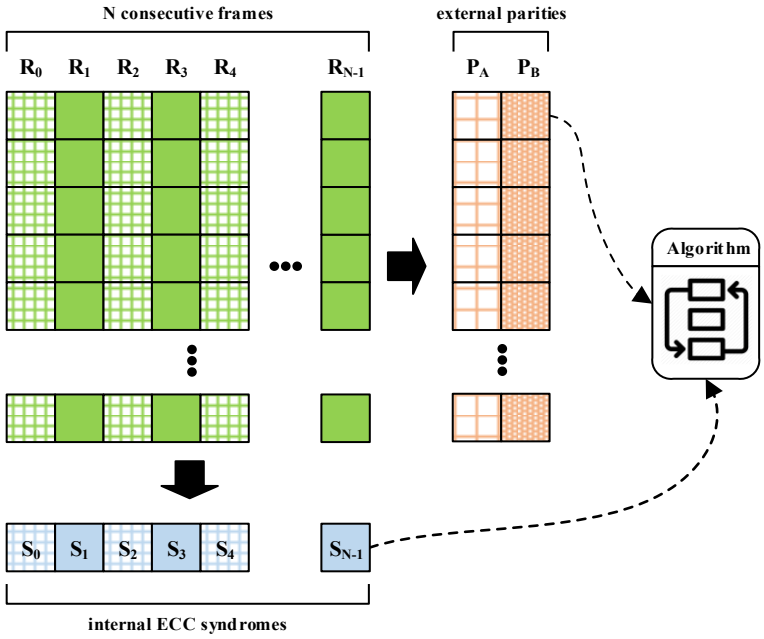


Figure 6.1: Mixed 2-D coding scrubbing approach

The concept of dividing each group into interleaved subsets of frames is motivated by the physical interleaving of configuration frames. Especially in the case of Zynq-7000, the interleaving schemes we identified in the detailed MCU analysis of the heavy ion radiation experiments, described in subsection 4.2.1.4 and Table 4.17, led to the design

of the algorithm architecture. The key idea behind the interleaving schemes is that the MCU events do not upset adjacent bits in one frame, but they affect multiple bits on adjacent frames, making the built-in ECC more effective, e.g., double MBUs in single frames are converted in SBUs affecting logically adjacent frames. The interleaved parity code of the mitigation technique guarantees that the adjacent frames are allocated into different subgroups and protected by different parity frames, reducing the possibility of fault masking due to multiple upsets in more than one frame of the same subgroup.

```
/* Single erroneous frame */
If M = 1
    correct all upsets in F0 based on E                    /* case A: SBU or MBU of single frame in a subgroup */
/* Multiple erroneous frames */
else
    for every frame Fi in F  i = 0 to M-1 do
        /* when syndrome matches parity */
        if Si belongs to E
            correct upset Si in Fi                         /* case B: SBU in each Fi */
        /* when syndrome does not match parity */
        else
            for every Fj in F  j = i+1 to M-1 do
                /* 2 SBUs in the same row */
                if Si = Sj
                    correct upset Si in Fi and Fj          /* case C: SBUs in Fi and Fj with the same syndrome*/
```

Figure 6.2: Error correction algorithm code

Let us explain how the error correction algorithm works. Assume that a single event has affected M frames $F = \{F_0, F_1, \ldots, F_{M-1}\}$, all belonging to subgroup G_A, and the on-chip ECC has produced the following non-zero syndromes $S = \{S_0, S_1, \ldots, S_{M-1}\}$. Notice that the error correction process is performed per subgroup. Thus, if there are erroneous frames in more than one subgroups, located either in the same group or in different groups, the scrubber creates one fault list per subgroup and runs the error correction process as many times as the number of affected subgroups. Then, the scrubber readbacks all the N frames of group G, calculates the parity frame P_A, and compares it with the golden parity frame P_A. Let us also assume that the comparison between the parity frames shows errors in K bit positions $E = \{E_0, E_1, \ldots, E_{K-1}\}$. Then, the error correction algorithm, shown in Figure 6.2, locates the upsets and corrects them by flipping the specific bits in the affected frames, using the syndromes (S) and the parity errors (E).

Figure 6.3 depicts six different examples of MCUs, which result in SBUs and MBUs in one or more frames, to demonstrate the error correction capabilities of the proposed algorithm.

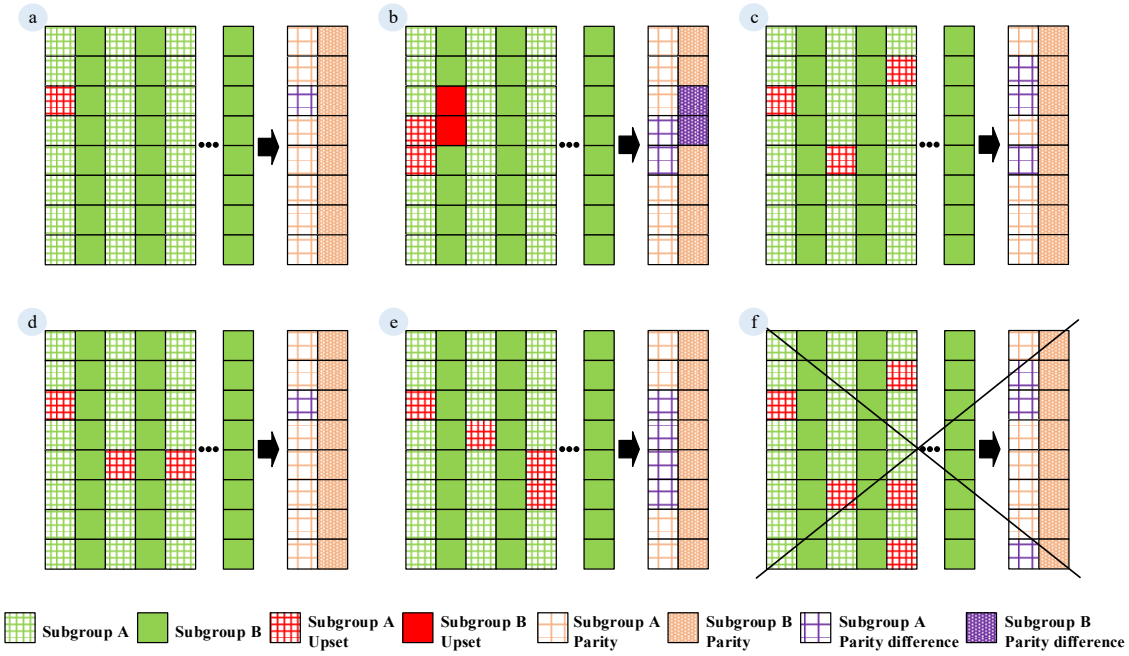


Figure 6.3: Error correction capability of the proposed algorithm

The algorithm guarantees the correction of upsets when:

i. A single frame in a subgroup has been affected by SBU or MBU: In the case where an SBU (Figure 6.3a) or MBU (Figure 6.3b) occurs in a single frame in a subgroup, the algorithm can locate the errors using the external parity data only and without the frame syndrome, running the case A of the code in Figure 6.2. This case also includes all MCU patterns, which have been generated by a single event and have affected two adjacent frames, since these frames belong to different subgroups. Thus, the proposed algorithm can correct all the heavy ion MCU patterns in Table 4.15 and the vast majority of the proton MCU patterns in Table 4.24. Remember that the errors in different subgroups are corrected by running the algorithm separately, since a parity frame is retained for each subgroup.

ii. Multiple frames in a subgroup have been affected by SBUs: In Figure 6.3c, where all the affected rows contain a single upset, the algorithm locates the errors by matching a syndrome in S with a bit position in E (case B of the code in Figure 6.2). In Figure 6.3d, where one row contains two upsets, the error is masked in the parity frame, thus, the algorithm locates the errors by detecting the cause of

masking, i.e., two syndromes in S with the same value (case C of the code in Figure 6.2). These cases include all the remaining proton MCU patterns in Table 4.24 that can not be corrected by the first case.

iii. Multiple frames in a subgroup have been affected by SBUs and one frame by MBU: The algorithm guarantees the correction of the errors, assuming that the MBU has occurred in different rows than the SBUs (Figure 6.3e). The correction will be performed in two stages: first, the algorithm will correct all the frames with SBU (cases B or C of the code in Figure 6.2), and then it will correct the only remaining frame affected by MBU (case A of the code in Figure 6.2). Note that such cases did not appear during radiation tests as described in chapter 4; they are rare and are likely to be caused by accumulated events.

Finally, there is a case where the algorithm does not guarantee the correction of errors. This is when MBUs and SBUs in different frames of the same subgroup have affected the same row(s), and thus, the algorithm cannot combine the syndromes and parity data to definitely locate the upset position. Let us look at the example in Figure 6.3f. The algorithm will try to fix all the frames with SBU by matching the location indicated by the syndromes to the bit position indicated by the parities. The first SBU can be corrected, but the second SBU cannot, as the error has been masked in the parity frame. Since the algorithm cannot locate the upsets, the FPGA must be fully reconfigured. Notice that none of the heavy ion and proton MCU patterns observed during our radiation experiments belong to this category of upsets.

Some algorithm parameters have been chosen to be customizable according to the physical layout and the dimensions of the memory in which the fault tolerance method will be applied and the radiation environment where the target device will operate. Such parameters are a) the frame-level interleaving scheme of the algorithm, i.e., the number of interleaved subgroups of a group, and b) the group size, i.e., the total number of frames in a group combined with the number of groups in which the memory is divided.

The frame-level interleaving scheme of the algorithm must be customized according to the configuration memory interleaving of the target device architecture. For example, in the AMD Xilinx 7-Series architecture, the configuration bits of adjacent configuration frames are physically interleaved with a depth of two [63], [151]. Based on this, the frames of each group are interleaved with a depth of two, producing two parity frames as explained above. The adjacent frames are allocated into different subgroups and parity frames, while the error correction process runs separately for each subgroup. On the contrary, the configuration memory in the AMD Xilinx UltraScale families uses a 2-way physical interleaving scheme, similar to the 7-Series, combined with a 4-way virtual interleaving, where every four consecutive bits of a physical frame are protected by four separate ECCs [34]. In this case, the algorithm will create two parity frames per group similar to the 7-Series, but each parity frame will be divided into four parity subframes following the 4-way virtual interleaving scheme of the UltraScale architecture. Using this

scheme, the error correction algorithm can locate the upsets and correct them by flipping the specific bits in the affected subframes, using the syndromes of the specific ECC and the parity errors of the specific subframe.

Another algorithm parameter that must be customized to achieve the best trade-off between the error correction latency and the memory overhead is the group size, i.e., the number of frames in a group. The selection of the group size depends on the total size of the memory and the radiation environment. In the case of a large memory, if the memory is divided into many groups, the algorithm will need a large number of total parity frames. On the other hand, in the case of radiation environments with high upset rates, if the memory is divided into many groups, the algorithm will take a short time to execute since each group will consist of a small number of frames. The two corner cases are: a) the group size is equal to the total number of device frames, i.e., all the configuration frames of the device are assigned in one group, which minimizes the memory overhead and the error coverage, and b) each frame is assigned into a separate group, which minimizes the error correction latency and maximizes error coverage. Thus, as the group size increases, the memory overhead is reduced at the cost of error correction latency increase.

6.2 Scrubber Implementations

Two different versions of the proposed scrubbing scheme were implemented. The first is an external scrubber suitable for systems incorporating an external microcontroller. The scrubber interfaces with the target device through a JTAG connection to access the contents of the configuration memory and the ECC logic. This solution combines the popularity of the JTAG interface with the software versatility provided by the external microcontroller. The second version is a full-hardware solution, where the scrubber is integrated within the target SoC-FPGA device (i.e., internal scrubber) by embedding the 2-D EDC algorithm on-chip. The primary benefit of the internal scrubber, which removes the necessity for an external controller, is the accelerated execution of the EDC algorithm. This leads to improved system availability.

6.2.1 External Scrubber Implementation

The external scrubber consists of two main components, the on-chip logic embedded in the target FPGA fabric and the EDC mechanism implemented in an external microcontroller, as shown in Figure 6.4. The on-chip logic is based on the AMD Xilinx Readback CRC to detect erroneous configuration frames, while the microcontroller communicates with the on-chip logic through a standard boundary-scan JTAG port and runs the 2-D EDC algorithm.

The on-chip logic uses the Readback CRC feature in conjunction with the FRAME_ECCE2 primitive to gain access to the internal ECC logic. The Readback CRC

is set to Continue mode to detect the presence of upsets in the configuration memory. It performs readbacks continuously and calculates the frame-level ECC syndromes and the global CRC without performing correction. The FRAME_ECC generates the frame-level ECC error flag, the syndrome, the current frame address, and the CRC error flag. An 512x64-bit asynchronous FIFO fitted in a 36Kb BRAM is used to store the FRAME_ECC outputs when an ECC or CRC error occurs, i.e., it retains up to 512 erroneous configuration frames. The FIFO data and the FIFO status (empty and full flags) are read through the JTAG serial interface. If the FIFO is empty, there is no upset and the external scrubber remains idle. If the FIFO is not empty, the external scrubber reads all the erroneous frames and starts the correction process. Otherwise, if the FIFO is full, it is likely that more than 512 frames have upsets and valuable information has been lost, and thus a reconfiguration of the entire device is required.

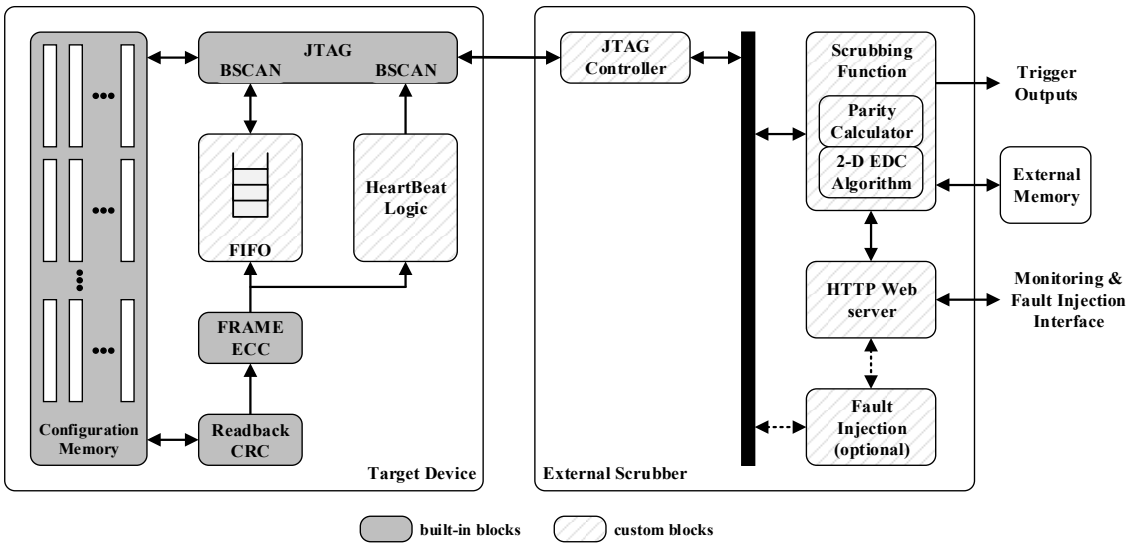


Figure 6.4: External scrubber architecture

The FIFO data write and read operations use different clock domains since the write and read clocks are not synchronized. The asynchronous FIFO helps to synchronize data flow between the Readback CRC, which uses the internal configuration clock, and the JTAG interface, which is clocked by the JTAG controller of the external scrubber. Note that the configuration clock is a signal output from the dedicated internal ring oscillator with a typical 65 MHz frequency. Also, note that the STARTUPE2 primitive [5] is the only possible way to obtain access to the internal configuration clock and make it available on the user design.

The HeartBeat logic provides watchdog functionality to alarm malfunctions of the frame ECC logic by storing the state of the FRAME_ECC at the end of each device scan cycle. The logic is implemented by a simple flip-flop, which is set to high at the end of the

configuration memory scan and is reset asynchronously in the case of an external JTAG access (i.e., the flip-flop is first read and then auto-cleared). Thus, if the internal logic stops scanning the configuration memory for any reason and never reaches the end of memory, the flip-flop is never set to '1', i.e., remains to '0', and the external scrubber will detect this malfunction, performing successive flip-flop reads. For example, in the case of Zynq-7020 device, which consists of 8,000 configuration frames with 101 words per frame, the Readback CRC needs 8,000 x 101 = 808,000 clock cycles, i.e., approximately 12.5 msec (using 65 MHz configuration clock), for a full scan cycle of the device, so the HeartBeat flip-flop is refreshed periodically every 12.5 msec. Consequently, the external read must be performed with a period twice the above refresh period to ensure that the HeartBeat is updated.

The FIFO and the HeartBeat logic are triplicated to reduce the likelihood of malfunction due to radiation effects. BSCANE2 primitives [5] are instantiated to provide general-purpose communication ports between the JTAG interface and the on-chip resources. Notice that separate BSCANs are used to provide access to the different resources instead of multiplexing all into a single primitive. This way, we simplify the routing between the BSCAN and the corresponding components, facilitating the AMD Xilinx Isolation Design Flow (IDF) typically used to design hardened circuits [9].

Table 6.2 shows the resource utilization of the on-chip logic implemented in the Zynq-7020 device. It is obvious that this logic occupies a small space in the FPGA and does not limit the use of programmable resources by any user design.

Resource	Utilization	Utilization %
LUT	351	0.66
FF	306	0.29
BRAM	3	2.14

Table 6.2: On-chip logic resource utilization for Zynq-7020

The external microcontroller should be able to act as JTAG master controller that performs FPGA configuration memory functions through JTAG and execute in software the EDC algorithm. The low-cost Digilent Zybo board [38] integrating an AMD Xilinx Zynq-7010 APSoC device was chosen to demonstrate the external scrubbing solution. The heterogeneous architecture of Zynq-7000, which incorporates a processing system based on the ARM Cortex-A9 CPU [24] and a programmable logic, can easily support these functions. A space-grade microcontroller, such as the Cobham Gaisler GR716 [83] or the BAE Systems RAD750 [27], could also fulfill this role, albeit with a slight decrease in performance. Space-grade CPUs are significantly slower than COTS CPUs; for instance, the Cortex-A9 offers 2.50 DMIPS/MHz, whereas the GR716 and RAD750 provide only 1.83 and 1.80 DMIPS/MHz, respectively. Nevertheless, the overall

performance reduction from using a space-grade processor will be minimal. This is because only a small portion of the error correction latency is attributed to the ECC algorithm executed on the processor, as detailed in section 6.3. It is important to note that if a non space-grade microcontroller is chosen, radiation hardening techniques should be implemented.

To achieve better performance, the system runs under a real-time operating system named FreeRTOS. FreeRTOS is an open-source, real-time operating system (RTOS) designed for embedded systems. It provides a lightweight, scalable, and efficient solution for managing multiple tasks in a microcontroller or microprocessor environment. FreeRTOS is a powerful and versatile RTOS that provides essential real-time capabilities for embedded systems.

The ARM Cortex-A9 CPU runs the three following tasks on top of the FreeRTOS: the scrubbing function, the HTTP web server, and the optional fault injection function. The scrubbing task communicates with the target device through the JTAG controller by sending JTAG commands and reading the FIFO and HeartBeat periodically to monitor the Readback CRC status. In case an ECC error is detected, the scrubber performs the 2-D EDC algorithm. Specifically, it reads the corresponding group frames, calculates the current parity frames, XORes them with the corresponding golden parity frames stored in the external memory, and stores the results locally. Then, it runs the error correction algorithm and writes back the corrected configuration frames. These tasks are performed for all the erroneous frame addresses stored in the FIFO. In case the algorithm cannot locate the upsets or a CRC-only error is detected in the target FIFO but not an ECC error, the scrubbing function asserts an external trigger output pin to reset the target FPGA. If the JTAG communication is lost or the watchdog function raises an alarm, the scrubbing function asserts a second trigger output pin to power cycle the FPGA.

Given that the platform provides all the configuration memory functions required by an FPGA fault injection environment, it optionally integrates a fault injection module. This module supports the injection of single-bit or multi-bit errors in the CRAM, BRAMs, LUTRAMs, and FFs of the target FPGA for testing purposes. In other words, the same infrastructure can be used to build a fault injection platform for SRAM-based FPGAs.

Finally, a simple web server has been implemented to provide the HTTP protocol. It is used to control and monitor the embedded platform through a web browser, which passes commands and data to the scrubbing functions using the interprocess communication methods supported by FreeRTOS. The scrubbing mechanism can be started and stopped by the user. The fault injection can be also configured through a simple webpage. In addition, statistics relative to the scrubbing operation or fault injection functionality, e.g., the number of erroneous frames detected, the number of frames corrected, the number of upsets injected, are exposed to the user. It must be noted that the external scrubbing platform can operate in standalone mode without a web-based supervisor.

The communication between the ARM CPU and the JTAG port of the target device is achieved through an on-chip high-performance AXI interface, which connects the PS with the PL, and an AXI-to-JTAG controller implemented on the PL side (Figure 6.5). The controller implements a JTAG master node and bridges the PS AXI bus with the JTAG interface. It works as an internal buffer that stores multiple JTAG commands and data to be transferred via JTAG. The AXI-to-JTAG controller can run at various clock speeds, making it compatible with a wide range of target devices, as well as different cable lengths using low-voltage differential signaling (LVDS), which reduces signal noise over long distances.

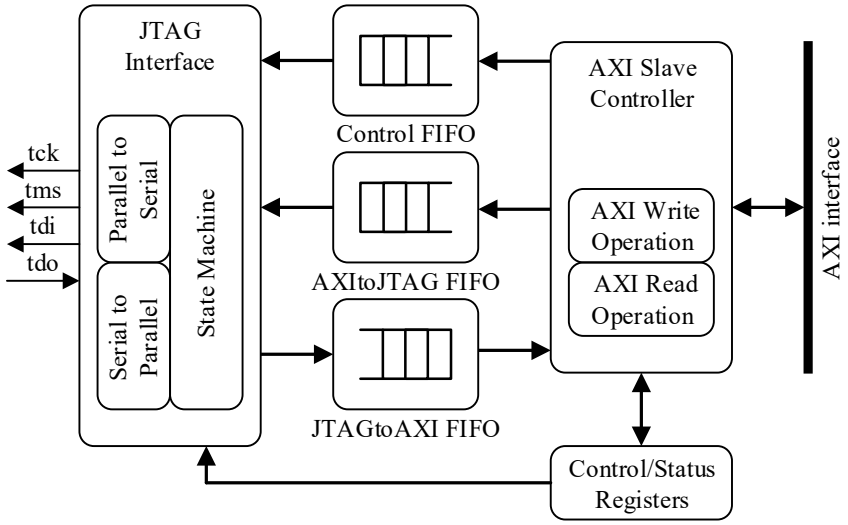


Figure 6.5: AXI-to-JTAG controller architecture

The AXI-to-JTAG controller consists of the following blocks:

- AXI Slave Controller: It implements the AXI slave write and read operations. It has access to the FIFOs or Register bank according to the AXI address.
- Control FIFO: It is a 128x32-bit FIFO which stores control commands. Each FIFO entry contains a command for a number of JTAG transitions to be shifted in the target device. A number of write accesses to AXItoJTAG FIFO corresponding to the number specified here must be made.
- AXItoJTAG FIFO: It is a 1024x32-bit FIFO. Each FIFO entry contains up to 16 tms and tdi bit transitions to be shifted in the target device.
- JTAGtoAXI FIFO: It is a 512x32-bit FIFO. Each FIFO entry contains up to 32 tdo bit transitions to be shifted out of the target device.
- Control/Status Registers: Two different registers are supported, the Calibration Register and Status Register.

The Calibration Register configures the JTAG clock frequency and determines the delay of returning data, adjusting the data capture mechanism accordingly. While shorter cables allow for higher speeds, longer cables may be necessary for applications, such as radiation experiments, where the external scrubber needs to be far away from the target device. However, using longer cables increases the delay of returning data and decreases the signal integrity due to electromagnetic interference. The calibration process is performed upon clock frequency configuration by reading the JTAG ID code of the target device. If the ID code is read successfully, then there is no calibration. If the selected clock speed is high, then it is possible that the ID code read is not valid because the data returns a clock cycle after it is expected. To overcome this, the data capture mechanism is delayed by a number of clock cycles resulting in a valid data capture.

The Status Register contains useful information for the CPU. It indicates if the controller is in idle state, where all the JTAG transitions stored in the Control FIFO are completed, i.e., the Control FIFO is empty and the JTAG interface state machine is in idle state. In addition, it includes three different fields that indicate how many entries exist in the three FIFOs.

- JTAG Interface: It reads the available command by the Control FIFO and the tms and tdi vectors by the AXItoJTAG FIFO and feeds the parallel-to-serial component with up to 16 bit shifts. It also loads to the JTAGtoAXI FIFO, the tdo transitions gathered by the serial-to-parallel component. Parallel-to-serial takes the 16-bit tms and tdi vectors and splits them into 16 serial transitions, while the serial-to-parallel takes a series of tdo transitions and creates a 32-bit tdo vector.

Table 6.3 shows the resource utilization of the AXI-to-JTAG controller implemented in the Zynq-7010 device.

Resource	Utilization	Utilization %
LUT	427	2.43
FF	561	1.59
BRAM	2	3.33

Table 6.3: AXI-to-JTAG controller resource utilization for Zynq-7010

6.2.2 Internal Scrubber Implementation

In the internal scrubber, the 2-D EDC algorithm is implemented on-chip, as shown in Figure 6.6. It uses the Readback CRC and the FRAME_ECCE2 primitives to gain access to the internal ECC, as in the external scrubber. The Syndromes Handler monitors the FRAME_ECC outputs and stores the configuration frames indicated as erroneous in the

Syndromes Memory. The Syndromes Handler also triggers the Parity Calculator to start calculating the parity frames of a group in which at least one of its frames has been recorded into the Syndromes Memory as erroneous. The Parity Calculator stores the calculation results in the Calculated Parity Memory. The 2-D EDC Algorithm reads the ECC syndromes from the Syndromes Memory and the parity data from the Calculated Parity Memory and executes the algorithm to locate the upsets in the erroneous frames of the group. Finally, it reads these frames and writes back the corrected data through the ICAP port.

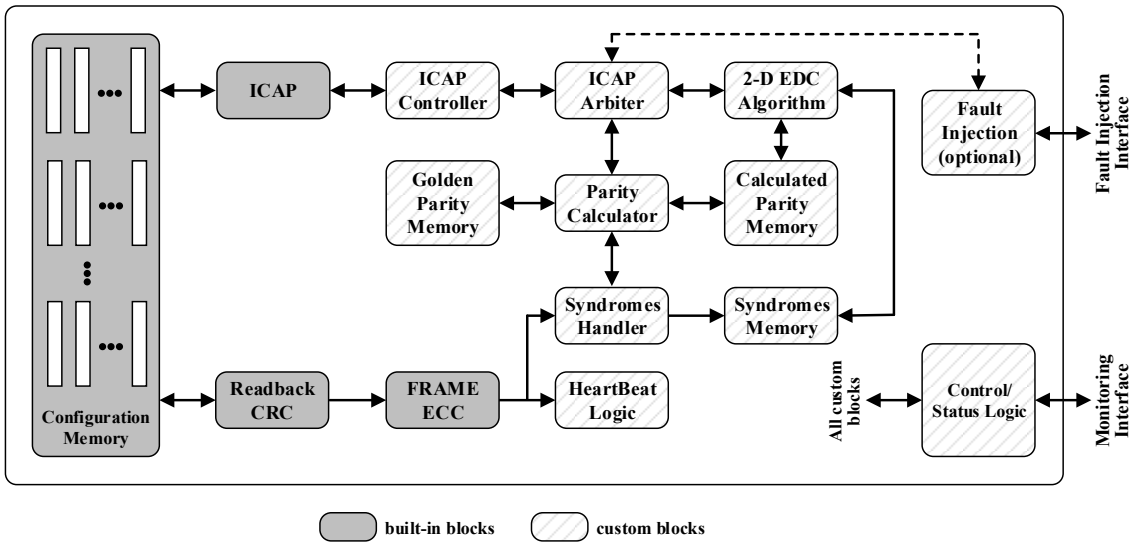


Figure 6.6: Internal scrubber architecture

In addition, the internal scrubber integrates control and status monitoring logic and an optional Fault Injection module. Namely, the ICAP controller provides access to the configuration memory through the ICAP port, while the ICAP Arbiter handles requests from various blocks to access the configuration memory. The HeartBeat module checks that the Readback CRC is alive. The Control/Status module provides an external port for enable/reset, error monitoring (e.g., CRC-only, uncorrectable ECC, heartbeat, ICAP synchronization errors, etc.), and logging purposes. It must be noted that the internal scrubber can operate in a standalone fashion, eliminating the need for an external controller. Finally, through the Fault Injection module, an external user can obtain access to the configuration memory and execute fault injection functions. The clock frequency of the design is limited by the maximum ICAP frequency, and thus, it can be up to 100 MHz. It must also be noted that all the soft cores of the internal scrubber have been protected against soft errors using the TMR technique, and the built-in ECC feature of the BRAM has been activated.

The internal scrubber consists of the following blocks:

- Syndromes Handler: It continuously monitors the configuration memory scan operation performed by the Readback CRC, checking the Frame ECC output. When an ECC error occurs, Syndromes Handler stores the address and the ECC syndrome of the erroneous frame into the Syndromes Memory. It also integrates a FIFO, each entry of which corresponds to a group of frames with at least one corrupted frame. Each entry contains the frame address offset, i.e. the address of the first frame in the group, the number of erroneous frames in the group and the subgroups that have frames in error. When the Readback CRC completes the scan of a specific number of groups, in which at least one of their frames has been recorded into the Syndromes Memory as erroneous, the Parity Calculator is triggered to start calculating the parity frames of the groups. In addition, a timeout counter ensures that the detected erroneous frames will not remain uncorrected for more than a specified time period; when the counter expires, the parity calculation is started even if the specific number of groups with erroneous frames has not been collected, but a subset of them has been stored in the Syndromes Memory. Finally, it reports the CRC-only error, i.e., when the Readback CRC has detected a CRC error but there is no ECC frame error. In this case, full reconfiguration is required.
- Syndromes Memory: It holds the address and the ECC syndrome of the erroneous frames detected by the internal ECC mechanism. The frames are stored in the Syndromes Memory by the Syndromes Handler and read by the 2-D EDC Algorithm to perform the correction process. Its size is configurable according to the number of erroneous groups that trigger the start of the correction process. For example, for a group which contains N frames, if the correction process starts when we detect up to N erroneous frames in that group, then the Syndromes Memory must, in the worst case, be able to store the address and syndrome of N frames.
- Parity Calculator: The Syndromes Handler signals this block to start the parity calculation of a specific group by passing the frame group address. The Parity Calculator reads the configuration frames of a group through the ICAP port and if these frames are the first frames of a subgroup, then XORes them with the corresponding golden parity frames, which are fetched by the Golden Parity Memory, and stores the resulting parity frames of each subgroup to the Calculated Parity Memory. It also XORes the remaining frames of the group that it reads from memory with the so far calculated parity frames and writes back to the Calculated Parity Memory the updated parity. This process is repeated until the entire group is read through the ICAP port. At the end of the process, the Calculated Parity Memory will contain the result of the comparison between the golden parity frames and the current parity frames. In addition, after the boot of the device and since the Golden Parity Memory is not preloaded with the golden parity frames upon device configuration, the Parity Calculator can calculate the

golden parity frames with the above procedure once and for the entire configuration memory and store the result of the calculation in the Golden Parity Memory.

- Golden Parity Memory: It holds the golden parity frames of the entire configuration memory. Assuming a configuration memory with F frames in total as well as groups that contain N frames and are divided into S subgroups, the total number of parity frames is equal to S*F/N; this is because the number of parity frames for every group corresponds to the number of its subgroups.
- Calculated Parity Memory: It holds the calculated parity frames of the group that is currently being processed for correction.
- 2-D EDC Algorithm: This basic unit implements the error detection and correction algorithm. It reads the ECC syndromes of the erroneous frames of a specific group from the Syndromes Memory and the parity frames from the Calculated Parity Memory and executes the algorithm to identify the upsets in the erroneous frames of the group. It also reads these frames, flips the bit upsets and writes back the corrected frames through the ICAP port.
- Fault Injection: This optional block provides an interface that allows an external user to perform fault injection for testing purposes. After it injects a fault, the scrubber takes over and executes the correction algorithm. In this way, a fault injection campaign can easily be carried out to prove that a user design will safely operate in a fault condition caused by a random hardware failure without the need for radiation testing.
- ICAP Controller: It provides a custom interface for any other component (e.g. Parity Calculator, 2-D EDC Algorithm, Fault Injection) that wants to write and read configuration memory through the ICAP port. It implements the configuration read and the reconfiguration (write) procedures [5], including manipulation of the DUMMY, SYNC and NOOP words. For instance, to write a configuration frame, only the frame address and frame data have to be provided through the custom interface and the ICAP controller takes care of generating the entire command sequence by inserting the above special words where required. Note that the device's packet buffer is flushed by writing DUMMY words, a special SYNC word is used to allow configuration logic to align at a 32-bit word boundary and at least one NOOP word is required at various points in the configuration procedure.
- ICAP Arbiter: It resolves requests from different blocks that want to access the configuration memory through the ICAP port, such as the Parity Calculator, 2-D EDC Algorithm and Fault Injection blocks, as shown in Figure 6.6, and also provides adaptation to the ICAP Controller interface. It receives requests from these blocks to read or write configuration frames from/to CRAM.

- HeartBeat Logic: It monitors the Readback CRC liveliness and informs Control/Status Logic module accordingly. If the configuration memory scan process suddenly stops, then the HeartBeat Logic is triggered, and a status signal is set to high to indicate this malfunction.
- Control/Status Logic: It provides an interface to externally control the scrubber, i.e. to enable and reset the scrubber, as well as to determine the CRAM region to be scrubbed. Note that the user design may occupy a specific region in the FPGA, thus, it would not be efficient to run the correction algorithm for upsets that occur in an "unused" fabric area. This block also provides an interface with a convenient set of decoded outputs that indicate the scrubber status, i.e. the status of the state machine of the different modules, the frame address and the data to be corrected.

Considering, for example, the Zynq-7000 architecture that uses a 2-way physical interleaving scheme for its configuration memory and a group with 64 configuration frames, the resource utilization of the internal scrubber implemented in the Zynq-7020 target device is presented in Table 6.4. The selection of the fundamental algorithm parameters, i.e., the total number of frames in a group and the number of interleaved subgroups of a group, is discussed in detail in subsection 6.3.1. The Synopsys Synplify software was used to apply the distributed TMR in all the custom blocks, while the AMD Xilinx built-in ECC option of the BRAMs was enabled.

Resource	Utilization	Utilization %
LUT	8083	15.19
FF	5778	5.43
BRAM	48	34.28

Table 6.4: Internal scrubber resource utilization for Zynq-7020

6.3 Reliability Analysis

This section first presents the reliability analysis of the mixed 2-D coding mitigation technique using the two scrubbing implementations for the target Zynq-7000 FPGA. Then, it compares these two implementations with other state-of-the-art techniques in terms of error coverage, latency and memory overhead. Finally, it discusses the adaptability of the proposed approach to different SRAM-based FPGA device families and its applicability to artificial terrestrial radiation environments.

6.3.1 Zynq-7000 FPGA Analysis

Two solutions for the AMD Xilinx Zynq-7000 FPGA architecture were implemented, as mentioned above: the external scrubber and the internal scrubber. The Zynq-7020 FPGA device was chosen as the target FPGA for both implementations, while the XC7Z010 FPGA as the external scrubber's microcontroller. The low-cost boards Avnet Zedboard [25], featuring a Zynq-7020 APSoC device, and Digilent Zybo [38], featuring a Zynq-7010 APSoC device, were selected to demonstrate the two solutions.

Remember that the scrubbing approach, presented in this thesis, protects the static CRAM frames, i.e., the frames scanned by the Readback CRC logic (block types 0, 2 & 3 in Table 2.3). On the contrary, the dynamic frames, i.e., BRAM content frames (block type 1 in Table 2.3), cannot be scrubbed since they may be modified during normal circuit operation. However, they can be protected using different mitigation methods, such as TMR or built-in ECC.

The two basic parameters of the algorithm that need to be adjusted for the target FPGA architecture are the number of parity frames per group (i.e., the number of subgroups per group) and the group size, as analyzed in subsection 6.1.2. The number of parity frames per group is set to 2, following the heavy ion and proton MBU/MCU analysis and the interleaving scheme of the Zynq-7000 architecture, presented in subsections 4.2.1.4 and 4.2.2.3. Note that all the MCU patterns shown in Table 4.15 and Table 4.24, which were observed during heavy ion and proton radiation tests, can be corrected using two subgroups per group.

On the other hand, the optimal group size is determined based on the following trade-off analysis. Let's take as an example the MCU patterns of the heavy ion irradiation experiment presented in Table 4.15. Figure 6.7 and Figure 6.8 depict how the number of frames per group affects the error correction latency and the memory overhead of the external and internal scrubber, respectively. The average error correction latency (ECL_{avg}) is calculated based on the following equation:

$$ECL_{avg} = \sum_{i=1}^{n} (ECL_i * freq_i) \quad \text{[Eq. 6.1]}$$

where ECL_i is the time to correct the MCU pattern i since the embedded ECC detects it and $freq_i$ is its frequency of occurrence, as measured in a radiation experiment. Suppose that the MCU has affected a single group, the error correction latency of the external (ECL_{ext}) and internal scrubber (ECL_{int}) can be calculated as:

$$ECL_{ext} = (ErrFr + 1) * FIFORd + GrSize * ParCal + ErrFr * (AlgoET + FrWr) \quad \text{[Eq. 6.2]}$$

$$ECL_{int} = GrScan + GrSize * ParCal + ErrFr * (AlgoET + FrWr) \quad \text{[Eq. 6.3]}$$

where *ErrFr* is the number of erroneous frames in the group, *FIFORd* is the FIFO read access time, *GrSize* is the group size, *ParCal* is the parity calculation latency per frame including the frame read time, *AlgoET* is the execution time of a single run of the EDC algorithm and *FrWr* is the time required to write back the correct frame data. Notice that the external scrubber must access the FIFO *ErrFr* times to read the erroneous frames plus once more to ensure that the Readback CRC has scanned the entire group. On the other hand, the internal scrubber requires *GrScan* time to initiate the error correction process, which is the time between the first erroneous frame is detected and the Readback CRC scans the last frame of the group. *GrScan* depends on the location of the erroneous frame and, on average, is the half scanning time of the group.

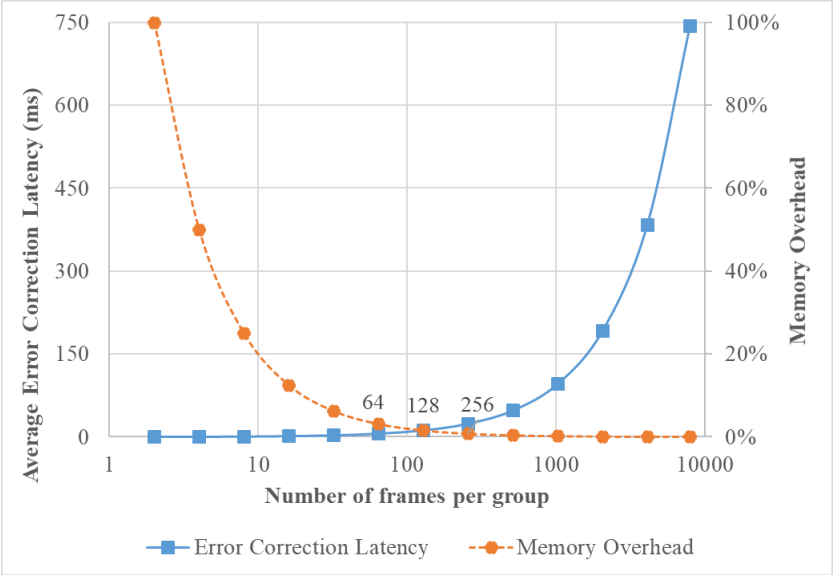


Figure 6.7: External scrubber –error correction latency vs memory overhead for different group sizes

The memory overhead is determined by comparing the memory size needed for storing group parity frames to the total group memory size. Figure 6.7 and Figure 6.8 illustrate that a decrease in group size significantly lowers the error correction latency but increases the memory overhead. The selection of the scrubber version, internal or external, and its design parameters, such as group size, should be based on the bit error rate and the reliability requirements of the target application. The internal scrubber excels over the external scrubber in terms of error correction latency, making it a suitable option for applications requiring high availability due to its shorter mean repair time. Conversely, if the error correction speed is not important or if integrating the scrubber on-chip is not

feasible due to limited programmable resources or power constraints, the external scrubber is the best choice. In such scenarios, employing a space-grade microcontroller for the external scrubber is recommended, as previously discussed. Regardless of the choice, Figure 6.7 and Figure 6.8 indicate that a group size ranging from 64 to 256 frames offers a balanced trade-off between error correction latency and memory overhead.

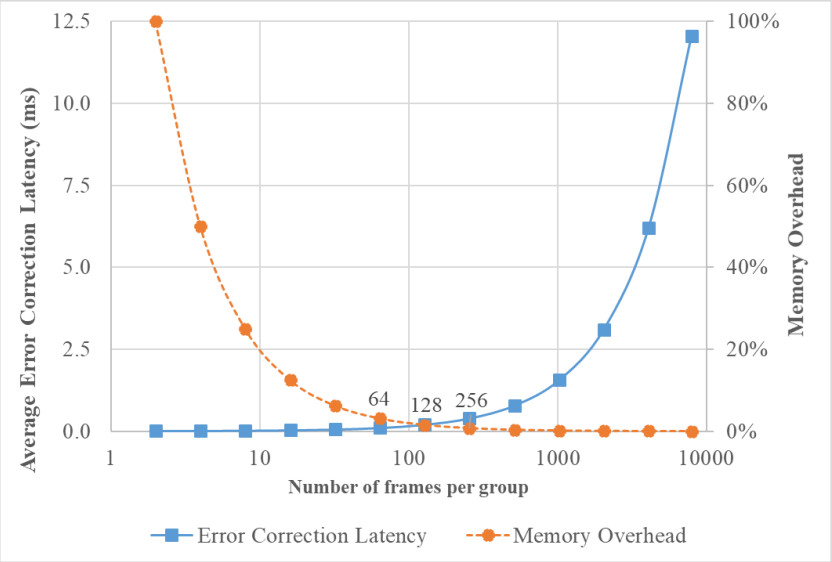


Figure 6.8: Internal scrubber –error correction latency vs memory overhead for different group sizes

In addition, the group size parameter can be affected by the configuration memory structure of the target FPGA family. For instance, in AMD Xilinx FPGAs, configuration frames are organized into columns based on their resource category, such as CLB columns, DSP columns, BRAM columns, etc. Each column type contains a specific number of frames; for example, in the Zynq-7000 architecture, CLB columns have 36 frames, DSP and BRAM columns have 28 frames, IOB columns have 42 frames, and CLK columns have 30 frames [5]. The addressing scheme of the configuration frames, which involves the column type, i.e., the column address field, and the frame offset, i.e., the minor address field, affects their read access time. Thus, it is more time-efficient to form groups that include an integer multiple of columns, resulting in variable group sizes, e.g., groups of one CLB column with 36 frames and groups of one DSP column with 28 frames, rather than maintaining a fixed size for all groups. Consequently, choosing an address space of 64 frames, the algorithm runs considering up to 64 frames per group, i.e., one column per group, and the two parity frames per group are calculated based on the frames of each column.

6.3.2 Comparison with the State-of-the-Art

Table 6.5 compares the two scrubbing implementations presented in this thesis with eight state-of-the-art scrubbing techniques, such as AMD Xilinx SEM IP [12], hybrid scrubbing [132], lightweight and fully testable SEU mitigation (LTFSM) system [1], frame-level redundancy (FLR) scrubbing [139], redundant configuration (RC) scrubbing [56], 2-D Hamming product code (HPC) [112], bit-interleaved embedded Hamming scheme [147], and I3D parity scheme with erasure code [43]. The comparison is performed in terms of error correction coverage, error correction latency (min, max, and average), and memory overhead. To calculate the error correction coverage and latency, let's consider the list of MCU patterns observed in the heavy-ion radiation experiments presented in Table 4.15, at LET of 8.8 MeVcm^2/mg and 12.45 MeVcm^2/mg. It should also be noted that the parameters in Table 6.5 for the external and internal scrubbers have been calculated based on two parity frames and one column per group, as described in the previous subsection.

The AMD Xilinx SEM IP mechanism, when operating in enhanced repair mode [12], is capable of correcting single-bit errors or adjacent double-bit errors within a frame; however, it is not equipped to handle MBUs with a multiplicity greater than two. A notable drawback of this method is the substantially high correction latency for double-bit errors, which is over 30 times longer than the latency for single-bit error correction. This increased latency is primarily due to the time-intensive nature of the CRC-based algorithm used to correct double-bit errors in adjacent frames. On the other hand, the memory overhead, measured by BRAM utilization, is relatively low compared to the total number of configuration bits.

Both the hybrid scrubbing and the LTFSM mechanisms are divided into internal FPGA logic and external logic. The internal logic rapidly corrects errors but is limited to SBUs. In contrast, the external logic handles the correction of all other error types, though this comes with increased latency due to the time required to retrieve golden frame data from an external storage device. This process incurs significant memory overhead because the entire golden bitstream must be stored. In addition, all odd-numbered MBUs are mistakenly identified as SBUs by the internal logic. Odd-numbered MBUs are subdivided into in-bounds and out-of-bounds MBUs. An odd-numbered in-bounds MBU suggests that the syndrome value, while not directly pointing to any actual upset bits, still corresponds to a valid word number, e.g., 0 to 100 for Zynq-7000. In contrast, an odd-numbered out-of-bounds MBU has a syndrome value that points to an invalid word number, e.g., greater than 101 for Zynq-7000. In the former case, the odd-numbered in-bounds MBUs are improperly repaired by the internal logic, introducing an additional error into the configuration memory. The external logic eventually detects this issue and provides the correct repair, but this prolongs the latency. On the other hand, the odd-numbered out-of-bounds MBUs do not actually cause changes to the configuration frame since the upset, indicated by the syndrome value, does not exist. The external logic

detects the erroneous frame and performs a configuration scrub by writing back the golden frame. The maximum correction latency of 42.8 ms in Table 6.5 occurs in the case where a multi-bit odd upset affects adjacent frames whose syndromes indicate out-of-bounds errors.

Error Correction Approach	Correction Coverage (%)	Correction Latency (msec)	Memory Overhead (%)
SEM IP [12]	99.22 / 98.17	min: 0.61 max: 37.58 avg: 1.38 / 2.59	0.64
hybrid scrubbing [132]	100.00 / 100.00	min: 0.0045 max: 42.8 avg: 0.32 / 0.72	Golden bitstream
LTFSM [1]	100.00 / 100.00	min: 0.0042 max: 25.84 avg: 0.25 / 0.59	Golden bitstream
FLR scrubbing [139]	100.00 / 100.00	0.005 ms	
RC scrubbing [56]	100.00 / 100.00	low for internal and high for external scrubbing	
2-D HPC [112]	99.41 / 97.26	moderate due to multiple 2-D ECC iterations	25.68
embedded Hamming scheme [147]	100.00 / 100.00	low due to single-frame access	< 3.25
I3D parity with erasure code [43]	100.00 / 100.00	moderate due to access in all the cluster frames	2.00
external scrubber	100.00 / 100.00	min: 3.78 max: 3.98 avg: 3.88 / 3.90	5.56
internal scrubber	100.00 / 100.00	min: 0.043 max: 0.083 avg: 0.062 / 0.063	5.56

Table 6.5: Comparison of error correction approaches

The FLR and RC scrubbing techniques offer extensive error correction coverage and quick repair times without needing a golden memory for the configuration bitstream.

However, their downside is that they limit user design to only 1/3 of the FPGA fabric. Additionally, the FPGA subregions dedicated to host redundant configurations must contain identical programmable resources, further reducing the area available for the user design.

The 2-D HPC without using bit interleaving provides relatively high error coverage, since it fails to recover the data only when multibit errors occur in both directions, as mentioned in the introduction of chapter 6. However, the memory needed to store the Hamming bits on-chip is extremely large. For example, if each configuration frame is treated as a 101x32-bit matrix, the horizontal and vertical Hamming codes would require 101x6 and 32x8 bits, respectively, resulting in 830 Hamming bits per frame, which translates to a 25.68% memory overhead for the Zynq-7020 device. Additionally, the error correction latency increases when multiple iterations of the 2-D ECC algorithm are needed to fix multibit errors.

The embedded Hamming scheme interleaves the bits of each frame into 13 subframes and uses non-essential bits within the configuration frames to store the parity bits. This approach is only feasible if there are enough non-essential bits to support a Hamming-compliant code; otherwise, the Hamming code must be stored in internal BRAMs or external memory. As a result, the memory overhead of this scheme depends heavily on its embedding efficiency. Additionally, the non-essential bits used to store the Hamming code are also susceptible to radiation effects, which can reduce the coding scheme's effectiveness. For instance, if an 8-bit Hamming code is applied to each subframe and a Zynq-7000 frame is divided into 13 subframes, the memory overhead could be as high as 3.25%. When the Hamming bits are embedded directly within the frames, the correction latency remains low since only a single frame access is required.

The I3D parity scheme utilizes 10 parity bits per frame, i.e., 5 bits horizontally, 2 vertically, and 3 diagonally, to achieve maximum detection coverage of 100% with a minimal memory overhead. When an error is detected, the corrupted frame is always recovered using a 50-cluster erasure code, i.e., one erasure block per 50 frames, ensuring 100% error correction coverage. While the I3D parity bits can be stored in the 19 upper spare bits of the ECC word within the same configuration frame, additional memory is required to store the redundant erasure blocks, resulting in a 2% memory overhead. The error correction latency is moderate, depending on the access time of the cluster frames.

This thesis introduces an approach that integrates the advantages of SEM, hybrid scrubbing, and LTFSM approaches, which utilize built-in frame-level ECC, with a 2-D coding scheme inspired by the efficient ECCs proposed in 2-D HPC, bit-interleaved embedded Hamming scheme and I3D parity scheme with erasure code. These combined features ensure 100% error coverage, successfully correcting all upsets observed in the heavy ion and proton experiments presented in subsections 4.2.1.4 and 4.2.2.3, while maintaining low to moderate error correction latency. Both external and internal scrubbers exhibit a nearly constant error correction latency, unlike SEM, hybrid

scrubbing, and LTFSM approaches, where latency can vary significantly. Specifically, the upper latency limit for correcting MCUs is substantially higher than that for SBUs. It's important to note that the correction latency for common SBU cases is higher with the external scrubber. However, this comparison is not entirely fair, as the external scrubber's JTAG configuration interface operates at a slower speed (e.g., 40 MHz) compared to the AXI and PCAP interfaces used by the hybrid scrubbing and the LTFSM approaches. Despite the slower JTAG interface, the external scrubber achieves a maximum correction latency of 10.7 times and 6.5 times lower than that of the hybrid scrubbing and LTFSM approaches, respectively. In addition, the approaches that utilize the internal ARM to manage the correction mechanism are more vulnerable to radiation effects, as described in chapter 5 of this thesis. On the contrary, the external scrubber can depend on a space-grade microcontroller for this task, as outlined in subsection 6.2.1. The external scrubber also eliminates the need to store the entire golden bitstream, unlike hybrid scrubbing and LTFSM. On the other hand, the internal scrubber offers lower error correction latency compared to the SEM, 2-D HPC, and I3D parity approaches. While both the external and internal scrubbers may involve higher error correction latency and memory overhead than the embedded Hamming technique, the additional circuitry required by the latter to map parity bits to non-essential bits is substantial, even with 100% embedding efficiency.

6.3.3 Universal Application of the Algorithm

6.3.3.1 Adaptation to other FPGA Device Families

The proposed approach was demonstrated for the AMD Xilinx Zynq-7000 family, but it can also be applied in all the FPGA architectures that support on-chip ECC, e.g., AMD Xilinx 7-Series, Virtex-5 and Virtex-6 and Intel Arria II & V, Stratix III, V & 10, Cyclone V. The only requirement is that the scrubber design must be adapted to the interface and features of the ECC module of the corresponding FPGA family. Table 6.6 summarizes these features of the various FPGA families.

Each ECC module supports different error detection and correction capabilities and produces different results that must be accessed and processed by the scrubber. AMD Xilinx Readback CRC is based on the frame-level ECC, i.e., 12 bits in Virtex-5 and 13 bits in Virtex-6 and 7-Series, and can detect SBUs and double-, odd- and most even-MBUs. The FRAME ECC hardware primitive monitors and outputs the Readback CRC information related to the detected errors, i.e., frame address, syndrome and error type.

	AMD Xilinx 7-Series, Zynq-7000, Virtex-5 & 6	**Intel Stratix 10**	**Intel Arria II & V, Stratix III, IV & V, Cyclone V**
ECC module	Readback CRC	Error Detection and Correction (EDC)	error detection CRC
ECC interface	FRAME ECC	error message queue	error message register
Error detection capability	SBUs, double-, odd- and most even-MBUs	SBUs, up to 8 bits in a rectangular area	all error types
Internal configuration port	read/write – ICAP	write-only – SDM	write-only – PR-IP
JTAG configuration port	read/write	read/write	read/write
JTAG access to internal logic	BSCAN	Virtual JTAG IP	Virtual JTAG IP

Table 6.6: FPGA families features

On the other hand, Intel Stratix 10 FPGAs [75] features the on-chip error detection and correction (EDC) circuitry to detect soft errors, using a 32-bit frame-level CRC to detect SBUs or up to 8 CRAM bits that fit in a rectangular box of 8 CRAM bits (i.e., 8x1, 4x2, 1x8 or 2x4 errors). The error message queue of the Intel Stratix 10 device stores the error messages when detecting an SEU error. The 8-entry error message queue is capable of storing a maximum of four different 2-entry messages. Each error message contains information about the sector address, number of errors detected in the sector, error type (i.e., single-bit or multi-bit), correction status (i.e., not corrected or corrected), and error location for single-bit errors. For multiple-bit errors, the error bit positions are not provided.

Similarly, Intel Arria II & V, Stratix III, IV & V and Cyclone V FPGAs [70] rely on the error detection CRC module. The width of the CRC field is 16 bits in Arria II, Stratix III, and Stratix IV devices and 32 bits in Arria V, Cyclone V, and Stratix V devices. When an error occurs, the error detection state machine identifies the error type and error location. All types of CRC errors can be detected in a frame. The errors are stored in the error message register. The error message register contains information on the error type, the location of the error, and the actual syndrome. This register is 46 bits wide in Arria II, Stratix III, and Stratix IV devices and 67 bits wide in Arria V, Cyclone V, and Stratix V devices. The types and locations of errors in the Arria II, Stratix III, and Stratix IV devices are reported for the single- and double-adjacent bit errors, while in the Arria V, Cyclone V, and Stratix V devices, they are reported for the single-, double-, triple-, and quadruple-adjacent bit errors. The location for other types of errors is not identified by the error message register. The contents of the error message register are updated when one or more errors occur.

In addition to the on-chip error detection and correction features, the scrubber must have access to the FPGA configuration memory to read the erroneous frames and write back the corrected ones. All the AMD Xilinx FPGAs families support both an ICAP and a JTAG external interface to implement the internal and external scrubber, respectively. On the contrary, Intel FPGA families can only support the external solution because their internal configuration port does not provide a direct interface to the FPGA fabric for reading configuration memory. The secure device manager (SDM) [73] in Intel's Stratix 10 FPGAs manages the access and security of the configuration memory internally, offering a more secure and controlled approach to handling configuration data. This design choice prioritizes security and reliability over user-driven flexibility in accessing the configuration memory, contrasting with AMD Xilinx's ICAP, which does allow direct access to the configuration memory from the FPGA fabric. In Intel FPGA families, Arria II & V, Stratix III, IV, & V, and Cyclone V, partial reconfiguration IP (PR-IP) [72] enables dynamic reconfiguration of specific regions of the FPGA device while other regions continue to operate without interruption. But this feature, also, does not provide an interface to the user logic to directly access and manipulate the configuration memory from within the FPGA fabric.

Both the AMD Xilinx and Intel FPGAs provide access to the on-chip user logic through the JTAG port by instantiating BSCAN primitives and the Virtual JTAG IP core [76], respectively. The Virtual JTAG is a specialized IP core designed for Intel FPGAs that enables the use of the JTAG protocol within the FPGA fabric. This IP core allows users to create virtual JTAG interfaces for various purposes, including debug, configuration, and communication, using JTAG signals directly within the FPGA.

6.3.3.2 Application to other radiation environments

Although the algorithm presented in this thesis mainly targets the space domain, its application can be extended to FPGA systems operating on other artificial terrestrial radiation environments, such as particle colliders, proton-therapy centers, robotics for nuclear decommissioning, etc. In such an application scenario, the metrics that must be prior evaluated are the memory overhead and the error correction coverage and latency.

First, the proposed approach imposes moderate memory overhead, which, in general, can be considered affordable even for constrained systems. Second, the error correction coverage has been measured based on the MBU/MCU patterns shown in Table 4.15 and Table 4.24, which we observed during the heavy ion and proton radiation experiments, but it can be safely assumed that the algorithm parameters, i.e., the number of parity frames per group and the group size, can be adjusted to cover MCU patterns caused by different radiation sources (i.e., heavy ions, protons, neutrons), and energies and expanded to more than two adjacent frames [33].

On the other hand, the error correction latency could affect the effectiveness of the approach in artificial radiation environments with very high upset rates. Specifically,

when the mean time between upsets is shorter than the mean repair time, faults will be accumulated and may cause the error correction algorithm to fail. For example, the algorithm may incorrectly repair an erroneous frame because the syndromes obtained from the FIFO are not consistent with the contents of the configuration frames; this may happen when a new upset occurs after the scrubbing mechanism reads the syndromes for the previous errors and before it reads back the targeted group. Let's assume these two realistic scenarios: a) the maximum mean flux of the ultra-high energy heavy-ion beam in the CERN accelerator complex is 2×10^3 ions*cm^{-1}*s^{-1} [3] and b) the maximum proton flux in proton therapy facilities for typical treatments is 10^9 particles*s^{-1} [138]. Based on the cross sections reported for the CRAM for heavy ions [151] and protons [136], the upset rates for these two radiation environments can be measured up to 68.52 upsets/sec (MTBF = 14.6 ms) for CERN and 16.62 upsets/sec (MTBF = 60.2 ms) for proton therapy center. Given that the maximum repair time equals a full Readback CRC scan cycle plus the error correction latency, i.e., 8 ms + 3.96 ms = 11.96 ms (for the external scrubber), it is lower than the MTBF in both cases. However, note that the effectiveness of the approach for larger devices must be recalculated since both the CRC scan cycle and the device cross section are analogous to the device size.

6.4 Evaluation of the Technique

The algorithm was evaluated in the CERN Super Proton Synchrotron North Area (SPS-NA) [3] in Geneva, Switzerland under heavy ions (Pb) irradiation. The current radiation experiment was performed in parallel with the experiment for the SEE characterization on the Zynq-7000 PL described in chapter 4. For the reader's convenience, let’s remember here some of the features of the beam (for more information, the reader can refer to chapter 4): the beam arrives at the facility in the form of spills of length between 4.5 sec and 10 sec, with a periodicity of ~41 sec. During the tests, the beam size was set to ~40 mm × 40 mm, and the beam flux was varied between 1×10^2 and 2×10^3 ions/cm^2 per spill. The test was performed for two different angles of incidence (θ), 0^o and 45^o, obtaining two effective LETs of 8.8 and 12.45 MeVcm^2/mg.

6.4.1 Radiation Test Setup

The external scrubber solution was used to evaluate the proposed mitigation approach. The radiation tests were performed in the Avnet ZedBoard [25], which integrates an AMD Xilinx Zynq-7020 APSoC as the target FPGA. The test setup is the same as those described in chapter 4, and it is shown in Figure 4.2. The ZedBoard is connected through the JTAG port with the Control Laptop (CL) located in the beam room, where the external scrubber runs. Note that at the time of the radiation experiments, the scrubber version based on the external microcontroller was not fully implemented; thus, an emulated version running in Python on the CL was evaluated. For the communication between the

CL and the on-chip logic in the target FPGA via the JTAG port and the execution of the configuration memory functions required by the scrubbing process, the open-source FREtZ framework [125], [145], described in subsection 4.1.2, was used. The beam trigger signal, which is monitored by the portable USB oscilloscope, indicates the start and the end of the beam-on periods. It triggers the external scrubber to read the on-chip logic and the target FPGA FIFO and run the error correction algorithm.

The main test operations are summarized below:

- The ZedBoard is powered by the DC PSU (Keysight N6705) remotely controlled by the remote laptop (RL) 2 located in the control room. RL 2 monitors the Ethernet-interfaced PSU to power cycle the board during, if any, high-current events.
- CL configures the target FPGA, communicates with its on-chip logic through the JTAG port and runs the correction algorithm.
- CL is remotely controlled and monitored by RL 1.
- The beam line is monitored by the USB oscilloscope connected to the CL. A beam trigger signal indicates the start and the end of the beam-on periods. When the Oscilloscope detects a falling edge in the beam trigger signal, it actually identifies the end of spill. This event detection triggers the host application running on CL to allow DUT configuration, data acquisition and logging while the beam is off.
- All system clocks are synchronized before the experiments with the facility's clock. All instruments' (PSU, CL, RLs) logs are time-stamped for post processing.

6.4.2 Circuit Under Test

The synthetic benchmark described in subsection 4.1.3 for the XC7Z020 device is used as circuit under test (CUT) for the evaluation tests. It instantiates all FFs, LUTs, BRAMs and DSPs, but a specific region of the device is left unused, i.e., no chain is allocated in this region to host the on-chip logic. The CUT communicates with RL 2 via BSCANE2 primitives and the JTAG interface, as shown in Figure 6.4. The on-chip scrubbing logic uses the Readback CRC feature in conjunction with the FRAME ECC primitive to gain access to the internal ECC logic. The Readback CRC is set to the CONTINUE mode in order to detect the presence of upsets in the configuration memory. It performs readbacks continuously and calculates the frame-level ECC syndromes and the global CRC. The FRAME ECC provides the frame-level ECC error flag, the syndrome, the current frame address and the CRC error flag. Note that the FRAME ECC does not correct the detected upsets. HeartBeat logic stores the state of the FRAME ECC at the end of each device scan cycle. A FF register is set to high, when the Readback CRC mechanism reads the last configuration frame and is auto-cleared when read through the JTAG interface. This

enables the external scrubber to perform a watchdog process in order to alarm the malfunction of the frame ECC.

Unlike the CUT used in SEE characterization, the current CUT is clocked. A simple logic is implemented to monitor the status of the long-register, BRAM and DSP chains. Specifically, the logic stores the toggle action at the end of the long-register chain, given that the chain is driven by a toggle bit logic, and it is auto-cleared when read through the JTAG interface. The BRAM and DSP chains are also monitored using similar logic. The external scrubber periodically accesses these logic flags through the BSCAN primitive in order to perform a watchdog functionality. Reading a zero flag means that the corresponding chain does not operate correctly. CL logs the errors into a file and the test sw GUI sets the Logic status LED shown in Figure 4.7.

In addition, the GLUTMASK configuration option is enabled in order to mask the changeable memory cell readback LUTRAM value, in contrast to the SEE characterization tests where the current state of LUTRAMs must be stored in the configuration memory cells. Remember that the scrubbing approach protects the static and not the dynamic configuration data.

Table 6.7 presents details about the configuration bitstream of the CUT for the XC7Z020 device.

Bit Type	XC7Z020	
	Bits	**%**
Configuration bits (non-masked)	19,257,500	59.53
CLB FF bits (masked)	85,740	0.26
CLB SRL/DRAM bits (masked)	898,368	2.78
Other unknown bits (masked)	152,184	0.47
BRAM content bits (masked)	4,796,416	14.83
Unused masked bits (e.g., PS area)	7,158,880	22.13
Total bitstream bits	32,349,088	100.00

Table 6.7: Circuit under test – bit type details

6.4.3 Test Flow

The test flow is shown in Figure 6.9. It performs the following steps:

- The ZedBoard is first powered and configured through the JTAG interface.
- During the irradiation period, no configuration action, i.e., readback or scrubbing, is performed to avoid the injection of errors due to abnormal behavior of the configuration interface.

- When the beam gets off, a beam-off timer is activated, which counts the time that the beam is off, and the error correction process initiates.
- A readback is performed to obtain the configuration memory image before error correction to evaluate the correct operation of the algorithm. For example, if the algorithm makes a wrong correction in memory, the memory image will help to find the faults injected by the algorithm.
- The FIFO of the on-chip logic, which holds the erroneous frames of the target FPGA, is cleared. Immediately afterwards, the Readback CRC is activated to scan the entire configuration memory and load the FIFO in case there are upsets.
- The status of the user logic, i.e., the long-register, BRAM and DSP chains, and the HeartBeat are checked. In case the HeartBeat alarms a system malfunction, a DUT power cycle is performed (power-off, power-on and board configuration).
- Otherwise, the FIFO is read and in case of ECC errors, the correction process is performed. It reads the corresponding frame groups, runs the error correction algorithm and writes back the corrected configuration frames.
- Two different scenarios were tested. In the first scenario, called EC beam-off, the error correction process runs only during the beam-off period of ~30 sec, i.e., no more upsets occur to the configuration memory while the correction algorithm is running. In this case, a beam-off timer is set, and if it expires before the error correction process is completed, but there are still uncorrected errors in the FIFO, the CUT is fully reconfigured and the process is repeated.
- In the second scenario, called EC cont, the error correction process runs continuously (i.e., the beam-off timer is set to a maximum value so that it never expires). When all the faults of the FIFO list are corrected, the FIFO is read again and the correction process continues. The risk of this scenario is that during the correction steps (and before all upsets are corrected), new soft errors may occur due to irradiation causing the algorithm to fail.
- In both cases, if a CRC-only error is read from the FIFO, which means that the frame-level ECC failed to detect an MBU, the DUT is fully reconfigured. This also occurs when the error correction process fails to correct some errors as described in subsection 6.1.2.

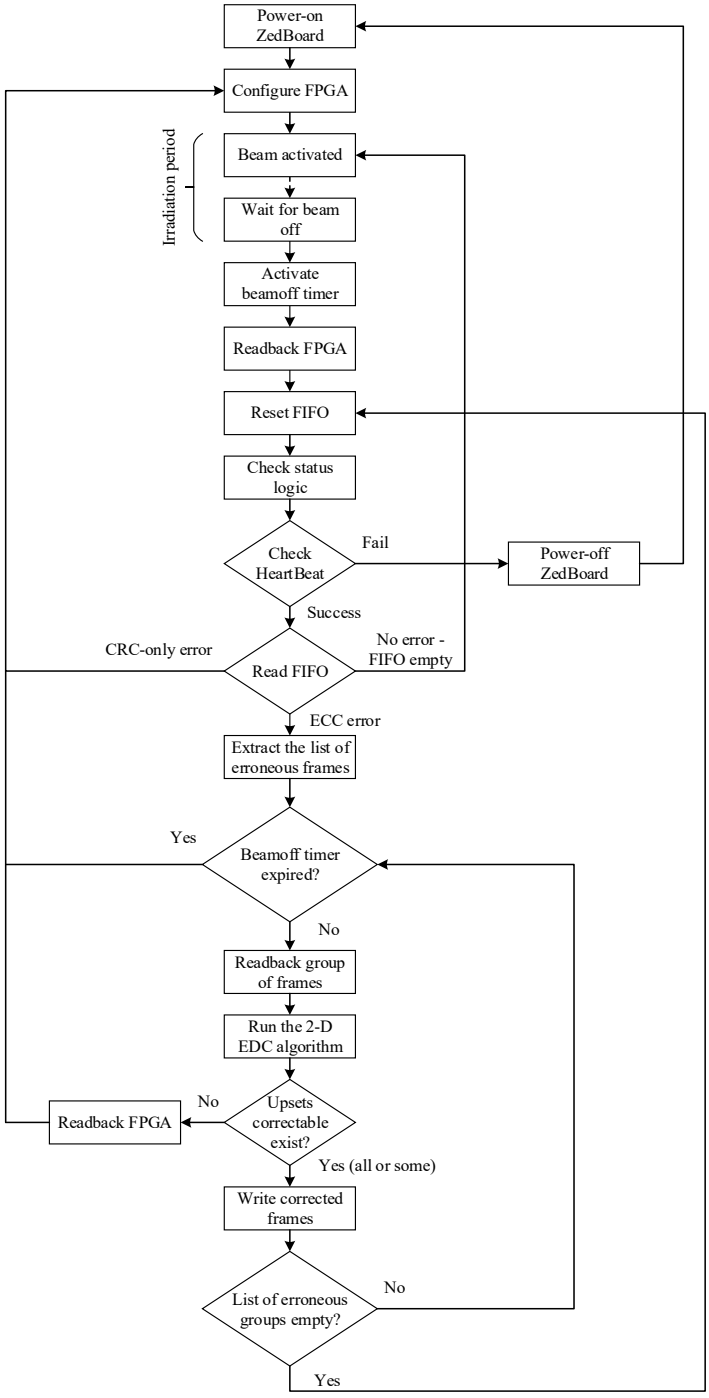


Figure 6.9: Test flow

Different test sessions were performed, as shown in Table 6.8. The EC beam-off test scenario was carried out for the two thetas, 0º and 45º, while the EC cont for theta 0º. The CRAM bit upsets and the corresponding configuration bit cross section were calculated; note that the values are in accordance with the results presented in [151].

	EC beam-off ($\theta = 0^o$)	**EC beam-off ($\theta = 45^o$)**	**EC cont ($\theta = 0^o$)**
Effective LET [MeVcm^2/mg]	8.80	12.45	8.80
Number of beam periods	349	41	571
Effective fluence [ions/cm^2]	3.19x10^5	0.31x10^5	5.23x10^5
Number of bit upsets	10944	1246	16077
Cross section [cm^2/bit]	1.90x10^{-9}	2.24x10^{-9}	1.71x10^{-9}

Table 6.8: Test sessions & cross sections

6.4.4 Results

Table 6.9 illustrates the effectiveness of the proposed scrubbing scheme, highlighting the number of erroneous cases detected and corrected by the EDC algorithm. Errors are classified based on the number of frames within each subgroup that were affected. During the EC beam-off sessions (as shown in columns 2 and 3 of Table 6.9), the scrubber achieved to complete the error correction process, successfully repairing all single and double erroneous frames. However, in the EC cont session (column 4 of Table 6.9), the scrubber did not manage to correct a small fraction of errors, 40 out of 11,665 cases. We believe that the algorithm failed in these cases because, during the time window between the point that the scrubber reads the on-chip FIFO and the point that it initiates the error correction process i.e., the time it requires to readback the group frames, a new upset might hit another frame in the same subgroup, resulting to a combination of MBUs and MCUs that the correction algorithm could not handle. This time interval lasts a few milliseconds. However, remember that the maximum upset rate in Galileo MEO orbit, is approximately 1 upset per minute. So, the algorithm does not actually fail; this occurs due to the accelerated radiation environment. The likelihood of this situation arising in a non-accelerated radiation environment is very low and becomes even smaller using either the internal or the external implementation of the scrubber, where the error detection latency of the algorithm is further reduced.

Frames per subgroup	**EC beam-off ($\theta = 0^o$)**	**EC beam-off ($\theta = 45^o$)**	**EC cont ($\theta = 0^o$)**
1	3315 (0)	107 (0)	9820 (0)
2	151 (0)	1 (0)	1602 (17)
3			178 (22)
4			21 (1)
5			4 (0)

Table 6.9: Number of corrected (uncorrected) cases

In summary, the algorithm detected and corrected a total of 17,388 configuration frames across 15,239 erroneous events in all test sessions. The majority of these events (i.e., 13,242) involved single-frame errors (see Figure 6.3 a & b), demonstrating that the algorithm can identify errors using only external parity data, regardless of the number of upsets. For multi-frame errors (refer to Figure 6.3 c, d, e & f), only a small subset of events (40, involving 104 frames in total) could not be corrected. Post-process analysis confirmed that the specific error type shown in Figure 6.3f was not among the 40 uncorrectable events, further supporting the explanation provided earlier.

7 Conclusions and Future Work

7.1 Conclusions

This thesis presents a comprehensive investigation into the SEEs induced by radiation in COTS APSoC devices, focusing on the AMD Xilinx Zynq-7000 APSoC. Through an extensive experimental and analytical study, this work explores the susceptibility of both the PL and PS to radiation effects and introduces an effective mitigation strategy to enhance the reliability of such devices in radiation-prone environments, such as space.

The first major contribution of the thesis lies in the systematic characterization of the PL part of the Zynq-7000, specifically under heavy-ion radiation using ultra- and very-high-energy ion beams at CERN and GSI. The study revealed critical insights into the SEE vulnerability of the PL by analyzing various types of embedded memories such as CRAM, LUTRAM, BRAM, and FFs. It was demonstrated that these memory types exhibit distinct susceptibilities to radiation, with the CRAM essential bits being particularly critical due to their direct influence on hardware functionality. This work established the cross sections for all relevant memory types, further refining the understanding of radiation-induced failure modes.

A detailed categorization of SEE phenomena, ranging from transient errors like SET-induced upsets in FFs to destructive SEFIs, was conducted. Notably, phenomena that compromise redundancy-based mitigation techniques, such as TMR, were observed and analyzed. This emphasized the necessity for advanced mitigation methods capable of addressing correlated or multi-bit upsets.

Subsequent experimental campaigns extended the characterization efforts to include proton irradiation, covering a broad range of energies (i.e., 30 - 250 MeV) and evaluating the PS part of the Zynq-7000 for the first time. These studies captured the SEU cross sections for SRAM arrays (e.g., OCM, L1/L2 caches) and provided an application-level perspective by executing diverse processor benchmarks under varying cache configurations. The analysis underscored how system-level behavior and error resilience are intricately tied to architectural and operational parameters.

The thesis culminates in the estimation of in-orbit error rates for various space mission profiles using OMERE software. The SEE rate predictions for different error categories

provide mission designers with valuable data to assess and manage the reliability of Zynq-7000-based systems in LEO and MEO orbits.

Finally, one of the most significant outcomes of this research is the development of an innovative SEE mitigation technique for SRAM-based FPGAs. The proposed scrubbing strategy introduces a mixed 2-D coding approach that synergistically combines the on-chip ECC with an interleaved, interframe parity mechanism. This method offers notable advantages over conventional hybrid scrubbing schemes, including enhanced correction of MBUs, minimized dependency on external golden bitstreams, and consistent error correction latency. Two implementations, an external and an internal scrubber, were realized and evaluated. The internal solution, with its fast on-chip execution, targets applications demanding high availability, while the external solution, leveraging software flexibility, suits reliability-centric applications.

The efficacy of this scrubbing approach was validated through irradiation tests at CERN, demonstrating successful correction of all observed SBUs and MBUs in the configuration memory. Comparative analysis with state-of-the-art techniques revealed superior performance in terms of error correction latency and area overhead.

In conclusion, this work significantly advances the understanding of radiation-induced SEEs in APSoCs and proposes a practical and scalable mitigation technique. The insights, methodologies, and mitigation strategy developed here contribute directly to the reliable deployment of COTS APSoCs in harsh radiation environments, bridging the gap between performance-rich commercial electronics and the stringent reliability demands of aerospace systems.

7.2 Future Work

This thesis lays the groundwork for enhancing the reliability of COTS processors, focusing on the dual-core ARM Cortex-A9 PS of the Zynq-7000 APSoC. The SEE characterization and experimental results presented form a solid foundation for future research into fault-tolerant computing architectures based on loosely-coupled lockstep execution.

Unlike tightly-coupled lockstep, which requires strict cycle-level synchronization, often impractical for modern COTS processors, the loosely-coupled approach allows independent execution with periodic synchronization and state comparison. This direction is particularly suited to the Zynq-7000, where the APU cores lack hardware support for fine-grained synchrony. Future work could involve developing and validating such a system, where each core executes redundant software independently and synchronizes at checkpoints. A checker circuit implemented in the PL will handle output comparisons and manage rollback operations.

The PS experimental findings in this thesis provide critical insights for designing synchronization intervals, checkpointing strategies, and rollback mechanisms. They represent an initial step towards realizing a practical lockstep system that balances reliability, performance, and integration complexity.

Further research could include validation through fault injection, while heavy-ion radiation testing could be pursued to confirm the system's resilience. Finally, demonstrating the approach on RTOS-based benchmarks could assess its practicality and influence future adoption in high-reliability embedded systems.

In summary, this thesis and its PS-level results can serve as a strong foundation for the continued development of loosely-coupled lockstep systems on COTS platforms.

Abbreviations – Acronyms

ACP	Accelerator coherency port
ALU	Arithmetic logic unit
API	Application programming interface
APU	Application processing unit
ARM	Advanced RISC machines
ASIC	Application-specific integrated circuits
AU	Astronomical unit
AXI	Advanced extensible interface
APSoC	All programmable system on a chip
BRAM	Block random-access memory
CLB	Configuration logic block
CMOS	Complementary metal oxide semiconductor
CMT	Clock management tile
COTS	Commercial off-the-shelf
CPU	Central processing unit
CRAM	Configuration random-access memory
CRC	Cyclic redundancy check
CUT	Circuit under test
DD	Displacement damage
DDR	Double data rate
DMA	Direct memory access
DMR	Dual modular redundancy
DRAM	Dynamic random-access memory

DSP	Digital signal processing
DUT	Device under test
ECC	Error correction code
EDAC	Error detection and correction
EMIO	Extended multiplexed I/O
ESA	European Space Agency
EU	European Union
FF	Flip-flop
FIFO	First-in first-out
FPGA	Field programmable gate array
FPU	Floating point unit
FREtZ	FPGA reliability evaluation through JTAG
GCR	Galactic cosmic rays
GEO	Geostationary orbit
GNC	Guidance, navigation and control
GNSS	Global navigation satellite system
GOES	Geostationary operational environmental satellites
GPS	Global positioning system
GUI	Graphical user interface
HDL	Hardware description language
HPSC	High performance spaceflight computing
ICAP	Internal configuration access port
I/O	Input / Output
IOB	Input / Output block
IP	Intellectual property
ISS	International space station
JTAG	Joint test action group
LCL	Latching current limiters
LEO	Low earth orbit
LET	Linear energy transfer

LUT	Lookup table
LUTRAM	Lookup table random-access memory
LVDS	Low-voltage differential signaling
MAC	Multiply and accumulate
MBU	Multiple-bit upset
MCU	Multiple-cell upset
MEO	Medium earth orbit
MIO	Multiplexed I/O
MMCM	Mixed-mode clock manager
MMU	Memory management unit
NASA	National Aeronautics and Space Administration
OBC	On-board computer
OCM	On-chip memory
OS	Operating System
OTP	One-time programmable
PC	Program counter
PCAP	Processor configuration access port
PCB	Printed circuit board
PL	Programmable logic
PLL	Phase-locked loop
PS	Processing system
PSU	Power supply unit
RAM	Random-access memory
RDBMS	Relational database management system
RHBD	Radiation hardened by design
RHBP	Radiation hardened by process
ROM	Read-only memory
RTL	Register-transfer level
SAMPEX	Solar anomalous and magnetospheric particle explorer
SBU	Single-bit upset

SCU	Snoop control unit
SDC	Silent data corruption
SEM	Soft error mitigation
SECDEC	Single-error correction, double error detection
SEE	Single event effect
SEFI	Single event function interrupt
SEL	Single-event latch-up
SET	Single event transient
SEU	Single event upset
SQL	Structured query language
SMEX	Small explorer
SoC	System on a chip
SRAM	Static random-access memory
SRL	Shift register look-up table
SSO	Sun-synchronous orbit
SWaP	Size, weight, and power
TAP	Test access port
TID	Total ionizing dose
TNID	Total non-ionizing dose
TSS	Tiangong space station
TMR	Triple modular redundancy
US	United States

References

[1] F. Abid, D. Jayasinghe, S. Somsavaddy, and S. Parameswaran, “LFTSM: Lightweight and fully testable SEU mitigation system for Xilinx processor-based SoCs,” in Proc. 30th International Conference on Field-Programmable Logic and Applications (FPL), Gothenburg, Sweden, Aug. 2020, pp. 162–168, doi: https://doi.org/10.1109/FPL50879.2020.00036

[2] J. H. Adams, R. Silberberg and C. H. Tsao, “Cosmic Ray Effects on Microelectronics. Part 1. The Near-Earth Particle Environment,” NRL Memorandum Report 4506, Naval Research Laboratory, Washington DC 20375-5000, USA, Aug. 1981. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA103897

[3] R. G. Alía, P. F. Martínez, M. Kastriotou, M. Brugger, J. Bernhard, M. Cecchetto, F. Cerutti, N. Charitonidis, S. Danzeca, L. Gatignon, A. Gerbershagen, “Ultraenergetic Heavy-Ion Beams in the CERN Accelerator Complex for Radiation Effects Testing,” IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 458-465, Jan. 2019, doi: https://doi.org/10.1109/TNS.2018.2883501

[4] G. Allen, F. Irom, and M. Amrba, “Zynq SoC Radiation Test Results and Plans for the Altera MAX10,” Proceedings of the NEPP Electronic Technology Workshop, Greenbelt, MD, USA, June 2015. [Online]. Available: https://nepp.nasa.gov/workshops/etw2015/talks/25%20-%20Thu/0950%20-%20Allen%20ETW%202015.pdf

[5] AMD Xilinx, 7-Series FPGAs Configuration – User Guide, UG470, v1.16, Feb 2023. [Online]. Available: https://docs.amd.com/v/u/en-US/ug470_7Series_Config

[6] AMD Xilinx, 7-Series FPGAs Configurable Logic Block – User Guide, UG474, v1.8, Sep 2016, [Online]. Available: https://docs.amd.com/r/en-US/ug474_7Series_CLB

[7] AMD Xilinx, 7-Series DSP48E1 Slice – User Guide, UG479, v1.10, Mar 2018, [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug479_7Series_DSP48E1

[8] AMD Xilinx, Vivado Design Suite 7 Series FPGA and Zynq 7000 SoC Libraries Guide, UG953, v2023.2, Oct 2023. [Online]. Available: https://docs.xilinx.com/r/en-US/ug953-vivado-7series-libraries

[9] AMD Xilinx, Isolation Design Flow for Xilinx 7 Series FPGAs or Zynq-7000 SoCs (Vivado Tools), XAPP1222, v1.4, Dec 2020, [Online]. Available: https://docs.amd.com/v/u/en-US/xapp1222-idf-for-7s-or-zynq-vivado

[10] AMD Xilinx, JTAG to AXI Master, [Online]. Available: https://www.amd.com/en/products/adaptive-socs-and-fpgas/intellectual-property/jtag_to_axi_master.html

[11] AMD Xilinx, QPro Virtex-II 1.5V Radiation-Hardened QML Platform FPGAs. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds124

[12] AMD Xilinx, Soft Error Mitigation Controller v4.1 LogiCORE IP Product Guide v4.1. [Online]. Available: https://docs.xilinx.com/r/en-US/pg036_sem

[13] AMD Xilinx, Space-grade Virtex 4QV FPGA. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/virtex-4qv.html

[14] AMD Xilinx, Space-grade Virtex 5QV FPGA. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/virtex-5qv.html

[15] AMD Xilinx, Zynq-7000 APSoC. [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html

[16] AMD Xilinx, Zynq-7000 APSoC Technical Reference Manual, UG585, v1.14, Jun 2023. [Online]. Available: https://docs.xilinx.com/r/en-US/ug585-zynq-7000-SoC-TRM

[17] AMD Xilinx Zynq-7000 SoC ZC706 Evaluation Kit. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html

[18] AMD Xilinx, Zynq Ultrascale+ MPSoc. [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

[19] AMD Xilinx, Versal Adaptive SoC. [Online]. Available: https://www.xilinx.com/products/silicon-devices/acap/versal.html

[20] AMD Xilinx, Versal Adaptive SoC Technical Reference Manual, AM011, v1.6, Oct. 2023. [Online]. Available: https://docs.amd.com/r/en-US/am011-versal-acap-trm

[21] AMD Xilinx, Virtex-4 FPGA Configuration User Guide, UG071, v1.12, June 2017. [Online]. Available: https://docs.amd.com/v/u/en-US/ug071

[22] M. Amrbar, F. Irom, S. M. Guertin, and G. Allen, “Heavy ion single event effects measurements of Xilinx Zynq-7000 FPGA,” in Proc. IEEE Radiation Effects Data

Workshop (REDW), Boston, MA, USA, Jul. 2015, pp. 1–4, doi: https://doi.org/10.1109/REDW.2015.7336714

[23] C. Argyrides, D. K. Pradhan and T. Kocak, "Matrix Codes for Reliable and Cost Efficient Memory Chips," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 3, pp. 420-428, March 2011, doi: https://doi.org/10.1109/TVLSI.2009.2036362

[24] ARM, Cortex-A9 – Technical Reference Manual, Revision r3p0, 2011. [Online]. Available: https://developer.arm.com/documentation/ddi0407/g

[25] AVNET, ZedBoard Product Brief (Datasheet). [Online]. Available: https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/zedboard/

[26] G.D. Badhwar, "The radiation environment in low-earth orbit" Radiation Research, vol. 148 no. 5, pp. S3-S10, Nov. 1999, doi: https://doi.org/10.2307/3579710

[27] BAE Systems Electronics, RAD750 family of radiation-hardened products. [Online]. Available: https://www.baesystems.com/en-media/uploadFile/20210404044504/1434555689265.pdf

[28] D. N. Baker, G. M. Mason, O. Figueroa, G. Colon, J. G. Watzin and R. M. Aleman, "An overview of the Solar Anomalous, and Magnetospheric Particle Explorer (SAMPEX) mission," in IEEE Transactions on Geoscience and Remote Sensing, vol. 31, no. 3, pp. 531-541, May 1993, doi: https://doi.org/10.1109/36.225519

[29] M. Blaum, J. Brady, J. Bruck and Jai Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures," in IEEE Transactions on Computers, vol. 44, no. 2, pp. 192-202, Feb. 1995, doi: https://doi.org/10.1109/12.364531

[30] C. Boatella Polo, "SEE effect, mechanism and testing," Radiation Environment and its Effects in EEE Components and Hardness Assurance for Space Applications CERN - ESA - SSC Workshop, ESA, Zurich, Switzerland, May 2017, Available: https://indico.cern.ch/event/635099/contributions/2570672/attachments/1456364/2249943/Single_Event_Effecs_Radiation_Course_May_2017_SEE_CBP.pdf

[31] T. Bradshaw, J. McCasland, J. Donckels, J. Meub and John Dickinson, "Development and Delivery of a Custom Versal Board for Space Applications," Single Event Effects (SEE) Symposium and Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, San Diego, USA, May 2023. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=5684&context=smallsat

[32] G. Bruni, P. Rech, L. A. Tambara, G. Nazar, F. L. Kastensmidt, R. Reis, A. Paccagnell, "Power dissipation effects on 28 nm FPGA-based system on chips

neutron sensitivity," in Proc. 22nd International Conference on VLSI and System-on-Chip (VLSI-SoC), Playa del Carmen, Mexico, Oct. 2014, pp. 1–4, doi: https://doi.org/10.1109/VLSI-SoC.2014.7004195

[33] P. Celis, J. Andres, "Statistical Method for Extracting Radiation-Induced Multi-Cell Upsets and Anomalies in SRAM-Based FPGAs," Theses and Dissertations, 9287, Brigham Young University, Nov. 2021, https://scholarsarchive.byu.edu/etd/9287

[34] K. Chapman, "SEU Mitigation Techniques for SRAM based FPGAs," Topical Workshop on Electronics for Particle Physics (TWEPP), Lisbon, Portugal, Sep. 2015. [Online]. Available: https://indico.cern.ch/event/357738/contributions/848841/attachments/1161500/1675463/Chapman_ID240.pdf

[35] L. H. Crockett, R. A. Elliot, M. A. Enderwitz, and R. W. Stewart, "The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC" First Edition, Strathclyde Academic Media, July 2014, ISBN:978-0-9929787-0-9

[36] P. Davis, D. S. Lee, M. Learn and D. Thorpe, "Single-Event Characterization of the 16 nm FinFET Xilinx UltraScale+TM RFSoC Field-Programmable Gate Array under Proton Irradiation," 2019 IEEE Radiation Effects Data Workshop, San Antonio, TX, USA, July 2019, pp. 1-5, doi: https://doi.org/10.1109/REDW.2019.8906566

[37] F. Davoli, Ch. Kourogiorgas, M. Marchese, A. Panagopoulos, F. Patrone, "Small satellites and CubeSats: Survey of structures, architectures, and protocols" International Journal of Satellite Communications and Networking, vol. 37, no. 4, pp. 343-359, Special Issue: Satnex IV, Jul./Aug. 2019, doi: https://doi.org/10.1002/sat.1277

[38] Digilent, Zybo Z7: Zynq-7000 ARM/FPGA SoC Development Board. [Online]. Available: https://digilent.com/shop/zybo-z7-zynq-7000-arm-fpga-soc-development-board

[39] B. Du, L. Sterpone, S. Azimi, D. Merodio Codinachs, V. Ferlet-Cavrois, C. Boatella Polo, R. G. Alía, M. Kastriotou, and P. Fernandez-Martínez, "Ultrahigh energy heavy ion test beam on Xilinx Kintex-7 SRAM-based FPGA," IEEE Transactions on Nuclear Science, vol. 66, no. 7, pp. 1813 - 1819, Jul. 2019, doi: https://doi.org/10.1109/TNS.2019.2915207

[40] X. Du, S. Liu, D. Luo, Y. Zhang, X. Du, C. He, X. Ren, W. Yang and Y. Yuan "Single event effects sensitivity of low energy proton in Xilinx Zynq-7010 system-on chip," Microelectronics Reliability, vol. 71, pp. 65–70, Apr. 2017, doi: https://doi.org/10.1016/j.microrel.2017.02.014

[41] A. Dufour, J. Carron, F. Pierron, M. Fongral, D. Dangla, G. Bascoul, F. Bezerra, J. Mekki, F. Malou and P. Maillard, "7nm FinFET technology heavy ion SEL evaluation using Xilinx Versal as case study," 2021 21th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Vienna, Austria, Sept 2021, pp. 1-6, doi: https://doi.org/10.1109/RADECS53308.2021.9954564

[42] M. Ebrahimi and M. B. Tahoori, "Stepped parity: A low-cost multiple bit upset detection technique," in Proc. IEEE International Test Conference (ITC), Anaheim, CA, USA, Oct. 2015, pp. 1–8, doi: https://doi.org/10.1109/TEST.2015.7342410

[43] M. Ebrahimi, P. M. B. Rao, R. Seyyedi, and M. B. Tahoori, "Low-cost multiple bit upset correction in SRAM-based FPGA configuration frames," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 3, pp. 932–943, Mar. 2016, doi: https://doi.org/10.1109/TVLSI.2015.2425653

[44] Embedded Systems Laboratory (ESLab) - University of Piraeus. http://eslab.cs.unipi.gr

[45] European Cooperation for Space Standardization (ECSS), "Space engineering – Methods for the calculation of radiation received and its effects, and a policy for design margins," ESA-ESTEC, Noordwijk, The Netherlands, ECSS-E-ST-10-12C, Nov 2008. [Online]. Available: https://ecss.nl/standard/ecss-e-st-10-12c-methods-for-the-calculation-of-radiation-received-and-its-effects-and-a-policy-for-design-margins

[46] European Cooperation for Space Standardization (ECSS), "Space engineering – Space environment," ESA-ESTEC, Noordwijk, The Netherlands, ECSS-E-ST-10-04C, Nov 2008. [Online]. Available: https://ecss.nl/standard/ecss-e-st-10-04c-space-environment

[47] European Cooperation for Space Standardization (ECSS), "Space product assurance: Techniques for radiation effects mitigation in ASICs and FPGAs handbook," ESA-ESTEC, Noordwijk, The Netherlands, ECSS-Q-HB-60-02A, September 2016. [Online]. Available: http://microelectronics.esa.int/asic/ECSS-Q-HB-60-02A1September2016.pdf

[48] European Space Components Coordination (ESCC), "Single Event Effects Test Method and Guidelines – ESCC Basic Specification No. 25100," European Space Agency (ESA), Issue 2, Oct 2014. [Online]. Available: https://escies.org/download/webDocumentFile?id=62690

[49] J. C. Fabero, H. Mecha, F. J. Franco, J. A. Clemente, G. Korkian, S. Rey, B. Cheymol, M. Baylac, G. Hubert, and R. Velazco, "Single event upsets under 14-MeV neutrons in a 28-nm SRAM-based FPGA in static mode," IEEE Transactions on Nuclear Science, vol. 67, no. 7, pp. 1461 - 1469, Jul. 2020, doi: https://doi.org/10.1109/TNS.2020.2977874

[50] B. Fiethe, H. Michalik, C. Dierker, B. Osterloh and G. Zhou, "Reconfigurable System-on-Chip Data Processing Units for Space Imaging Instruments," 2007 Design, Automation & Test in Europe Conference & Exhibition (DATE), Nice, France, April 2007, pp. 1-6, doi: https://doi.org/10.1109/DATE.2007.364420

[51] S. T. Fleming and D. Thomas, "Injecting FPGA Configuration Faults in Parallel," 2018 International Conference on Field-Programmable Technology (FPT), Naha, Japan, Dec. 2018, pp. 198-205, doi: https://doi.org/10.1109/FPT.2018.00037

[52] G. Furano, A. Tavoularis, L. Santos, V. Ferlet-Cavrois, C. Boatella, R. G. Alía, P. Fernandez Martinez, M. Kastriotou, V. Wyrwoll, S. Danzeca, M. Tali, D. Gacnik, I. Kramberger, L. Juul, K. Maragos and G. Lentaris, "FPGA SEE test with ultra-high energy heavy ions," in Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Chicago, IL, USA, Oct. 2018, pp. 1–4, doi: https://doi.org/10.1109/DFT.2018.8602958

[53] G. Furano and A. Menicucci, "Roadmap for On-Board Processing and Data Handling Systems in Space," Dependable Multicore Architectures at Nanoscale, Cham, Switzerland: Springer, Cham, 2018, pp. 253-281, doi: https://doi.org/10.1007/978-3-319-54422-9_10

[54] Gaisler, "GR740 Technical Note on Benchmarking and Validation," Doc. No GR740-VALT-0010, Issue 3.4, April 2019 [Online]. Available: https://www.gaisler.com/doc/gr740/GR740-VALT-0010.pdf

[55] A. D. George and C. M. Wilson, "Onboard Processing with Hybrid and Reconfigurable Computing on Small Satellites," Proceedings of the IEEE, vol. 106, no. 3, pp. 458-470, Mar 2018, doi: https://doi.org/10.1109/JPROC.2018.2802438

[56] R. Giordano, S. Perrella, V. Izzo, G. Milluzzo, and A. Aloisio, "Redundant-configuration scrubbing of SRAM-based FPGAs," IEEE Transactions on Nuclear Science, vol. 64, no. 9, pp. 2497–2504, Sep. 2017, doi: https://doi.org/10.1109/TNS.2017.2730960

[57] R. Glein, F. Rittner, A. Becher, D. Ziener, J. Frickel, J. Teich, A. Heuberger, "Reliability of space-grade vs. COTS SRAM-based FPGA in N-modular redundancy," 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Montreal, QC, Canada, June 2015, pp. 1-8, doi: https://10.1109/AHS.2015.7231159

[58] M. Glorieu, A. Evans, T. Lange, A. -D. In, D. Alexandrescu, C. Boatella-Polo, R. G. Alía, M. Tali, C. U. Ortega, M. Kastriotou, P. Fernández-Martínez and V. Ferlet-Cavrois, "Single-Event Characterization of Xilinx UltraScale+® MPSOC under Standard and Ultra-High Energy Heavy-Ion Irradiation," July 2018 IEEE Radiation Effects Data Workshop (REDW), Waikoloa, HI, USA, 2018, pp. 1-5, doi: https://doi.org/10.1109/NSREC.2018.8584296

[59] J. Gonzalo, D. Domínguez, D. López, "On the challenge of a century lifespan satellite," Progress in Aerospace Sciences, vol. 70, pp. 28-41, Oct. 2014, doi: https://doi.org/10.1016/j.paerosci.2014.05.001

[60] E. Gregersen, "Heliosphere," Encyclopedia Britannica, Sep. 2018. https://www.britannica.com/science/heliosphere (accessed Jan. 21, 2024)

[61] P.K.F. Grieder, "Cosmic Rays at Earth – Researcher's Reference Manual and Data Book," Elsevier (2001), ISBN: 0-444-50710-8

[62] M. Happe, A. Traber and A. Keller, "Preemptive Hardware Multitasking in ReconOS," Applied Reconfigurable Computing – 11th International Symposium (ARC), Bochum, Germany, pp. 79–90, March 2015, doi: https://doi.org/10.1007/978-3-319-16214-0_7

[63] A. Harding and M. Wirthlin, "Improving the Reliability of Xilinx 7 Series FPGAs through Configuration Scrubbing," 20th Annual Fellowship Symposium Utah NASA Space Grant Consortium Hill Aerospace Museum, April 2014. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1365&context=spacegrant

[64] D. M. Hiemstra and V. Kirischian, "Single Event Upset Characterization of the Zynq-7000 ARM® Cortex™-A9 Processor Unit Using Proton Irradiation," 2015 IEEE Radiation Effects Data Workshop (REDW), Boston, MA, USA, 2015, pp. 1-3, doi: https://doi.org/10.1109/REDW.2015.7336735

[65] D. M. Hiemstra, V. Kirischian and J. Brelski, "Single Event Upset Characterization of the Zynq UltraScale+ MPSoC Using Proton Irradiation," 2017 IEEE Radiation Effects Data Workshop (REDW), New Orleans, LA, USA, July 2017, pp. 1-4, doi: https://doi.org/10.1109/NSREC.2017.8115448

[66] K. Huey, "Xilinx Virtex-5QV Update and Space Roadmap," in 3rd SpacE FPGA Users Workshop (SEFUW), March 2016. [Online]. Available: https://indico.esa.int/event/130/contributions/717/

[67] J. Hussein and G. Swift, "Mitigating Single-Event Upsets - White Paper: 7 Series FPGAs," WP395, v1.1, May 2015. [Online]. Available: https://docs.xilinx.com/v/u/en-US/wp395-Mitigating-SEUs

[68] K. Iniewski, "Radiation Effects in Semiconductors," (1st. ed.), 2011, CRC Press Taylor & Francis Group, Boca Raton, Florida, USA, Hardback ISBN: 9781439826942

[69] Intel, Agilex SoC. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/agilex.html

[70] Intel, “AN 539: Test Methodology of Error Detection and Recovery using CRC in Intel FPGA Devices,” Aug 2019. [Online]. Available: https://cdrdv2-public.intel.com/666501/an539-683075-666501.pdf

[71] Intel, Cyclone SoC. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/cyclone.html

[72] Intel, “Partial Reconfiguration IP Core,” April 2019. [Online]. Available: https://cdrdv2.intel.com/v1/dl/getContent/666325?fileName=ug-partrecon-683404-666325.pdf

[73] Intel, “Stratix 10 Configuration User Guide,” UG-S10CONFIG, April 2024. [Online]. Available: https://cdrdv2.intel.com/v1/dl/getContent/772137?fileName=ug-s10-config-683762-772137.pdf

[74] Intel, Stratix 10 SoC FPGA. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/stratix/10.html

[75] Intel, “Intel Stratix 10 SEU Mitigation User Guide,” UG-S10SEU, Sep 2020. [Online]. Available: https://cdrdv2-public.intel.com/704708/ug-s10-seu-20-2-683602-704708.pdf

[76] Intel, “Virtual JTAG Intel® FPGA IP Core User Guide”, UG-SLDVRTL, Aug 2021. [Online]. Available: https://cdrdv2-public.intel.com/666577/ug_virtualjtag-683705-666577.pdf

[77] International Organization for Standardization, “ISO 15390:2004 - Space environment (natural and artificial) - Galactic cosmic ray model,” June 2004, [Online]. Available: https://www.iso.org/standard/37095.html

[78] X. Iturbe, D. Keymeulen, P. Yiu, D. Berisford, K. Hand, R. Carlson and E. Ozer, “Towards a generic and adaptive System-on-Chip controller for space exploration instrumentation,” 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Montreal, QC, Canada, 2015, pp. 1-8, doi: https://doi.org/10.1109/AHS.2015.7231151

[79] X. Iturbe, D. Keymeulen, P. Yiu, D. Berisford, R. Carlson, K. Hand and E. Ozer, “On the Use of System-On-Chip Technology in Next-Generation Instruments Avionics for Space Exploration. In: Shin, Y., Tsui, C., Kim, JJ., Choi, K., Reis, R. (eds) VLSI-SoC: Design for Reliability, Security, and Low Power. VLSI-SoC 2015. IFIP Advances in Information and Communication Technology, vol 483. Springer, Cham, doi: https://doi.org/10.1007/978-3-319-46097-0_1

[80] JEDEC, “Test Method for Beam Accelerated Soft Error Rate,” JESD89-3A, Nov. 2007. [Online]. Available: https://www.jedec.org/sites/default/files/docs/jesd89-3a.pdf

[81] JEDEC, “Test Procedures for the Measurement of Single-Event Effects in Semiconductor Devices from Heavy Ion Irradiation,” JESD57A, Nov. 2017. [Online]. Available: https://www.jedec.org/system/files/docs/JESD57A.pdf

[82] D. C. Jensen and J. C. Cain, “An interim geomagnetic field,” J. Geophys. Res., 67, 3568, 1962. [Online]. Available: https://garfield.library.upenn.edu/classics1979/A1979HZ37500001.pdf

[83] F. Johansson, S. Habinc, J. Andersson F. Sturesson, N. Wessman, S. Redant, L. Berti, G. Thys, C. Monteleoneet, “Status update on GR716 Rad-Hard Microcontroller for Space Applications,” 7th International Workshop on Analogue and Mixed-Signal Integrated Circuits for Space Applications (AMICSA), Leuven, Belgium, June 2018, pp. 179–183. [Online]. Available: https://indico.esa.int/event/222/contributions/2065

[84] A. Johnson, S. Aaron, J. Chang, Y. Cheng, J. Montgomery, S. Mohan, S. Schroeder, B. Tweddle, N. Trawny, J. Zheng, “The Lander Vision System for Mars 2020 entry descent and landing,” AAS Guidance Navigation and Control Conference, Breckenridge, Colorado, February 2017. [Online]. Available: https://ntrs.nasa.gov/citations/20190025658

[85] A. Johnston, “Reliability and Radiation Effects in Compound Semiconductors,” NASA Jet Propulsion Laboratory, California Institute of Technology, USA, April 2010, doi: https://doi.org/10.1142/7331

[86] S. Kanekal, K. Ogasawara, N. Paschalidis, E. Christian, M. Desai, Q. Schiller, L. Blum, A. Greeley, “The Compact Radiation belt Explorer, CeREs: A CubeSat mission to study electron microbursts”, 42nd COSPAR Scientific Assembly, Pasadena, California, USA, July 2018

[87] J. Karp, M. J. Hart, P. Maillard, G. Hellings and D. Linten, “Single-Event Latch-Up: Increased Sensitivity From Planar to FinFET,” in IEEE Transactions on Nuclear Science, vol. 65, no. 1, pp. 217-222, Jan. 2018, doi: https://doi.org/10.1109/TNS.2017.2779831

[88] F. L. Kastensmidt, L. Carro, R. Reis, “Fault-Tolerance Techniques for SRAM-based FPGAs,” Springer New York, doi: https://doi.org/10.1007/978-0-387-31069-5

[89] M. Kastriotou, P. Fernandez-Martinez, R. G. Alía, C. Cazzaniga, M. Cecchetto, A. Coronetti, G. Lerner, M. Tali, N. Kerboub, V. Wyrwoll, J. Bernhard, “Single Event Effect Testing With Ultrahigh Energy Heavy Ion Beams,” IEEE Transactions on Nuclear Science, vol. 67, no. 1, pp. 63-70, Jan. 2020, doi: https://doi.org/10.1109/TNS.2019.2961801

[90] K. A. LaBel and R. Ladbury, “Proton Particle Test Fluence: What's the Right Number?” JEDEC JC-13 Joint Electron Device Engineering Council (JEDEC),

Committee Meeting. No. GSFC-E-DAA-TN26472. Columbus, OH, 2015. [Online]. Available: https://nepp.nasa.gov/files/27741/NEPP-CP-2015-LaBel-Pres-JEDEC-TN26472.pdf

[91] D. S. Lee, G. R. Allen, G. Swift, M. Cannon, M. Wirthlin, J. S. George, R. Koga and K. Huey, "Single-Event Characterization of the 20 nm Xilinx Kintex UltraScale Field-Programmable Gate Array under Heavy Ion Irradiation," 2015 IEEE Radiation Effects Data Workshop (REDW), Boston, MA, USA, July 2015, pp. 1-6, doi: https://doi.org/10.1109/REDW.2015.7336736

[92] D. S. Lee, M. King, W. Evans, M. Cannon, A. Pérez-Celis, J. Anderson, M. Wirthlin and W. Rice, "Single-Event Characterization of 16 nm FinFET Xilinx UltraScale+ Devices with Heavy Ion and Neutron Irradiation," 2018 IEEE Radiation Effects Data Workshop (REDW), Waikoloa, HI, USA, July 2018, pp. 1-8, doi: https://doi.org/10.1109/NSREC.2018.8584313

[93] D. S. Lee, M. Wirthlin, G. Swift and A. C. Le, "Single-Event Characterization of the 28 nm Xilinx Kintex-7 Field-Programmable Gate Array under Heavy Ion Irradiation," 2014 IEEE Radiation Effects Data Workshop (REDW), Paris, France, July 2014, pp. 1-5, doi: https://doi.org/10.1109/REDW.2014.7004595

[94] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona and G. Furano, "High-performance embedded computing in space: Evaluation of platforms for vision-based navigation," Journal of Aerospace Information Systems, vol. 15, no. 4, pp 178-192, April 2018, doi: https://doi.org/10.2514/1.I010555

[95] A. Lesea, W. Koszek, G. Steiner, G. Swift, and D. White, "Soft error study of ARM SoC at 28 nanometers," in Proc. IEEE Workshop Silicon Errors Logic-Syst. Effects, Apr. 2014, pp. 1–4

[96] A. Lindoso, M. García-Valderas, L. Entrena, Y. Morilla and P. Martín-Holgado, "Evaluation of the Suitability of NEON SIMD Microprocessor Extensions Under Proton Irradiation," in IEEE Transactions on Nuclear Science, vol. 65, no. 8, pp. 1835-1842, Aug. 2018, doi: https://doi.org/10.1109/TNS.2018.2823540

[97] T. M. Lovelly and A. D. George, "Comparative analysis of present and future space-grade processors with device metrics," Journal of Aerospace Information Systems, vol. 14, no. 3, pp 184-197, 2017, doi: https://doi.org/10.2514/1.I010472

[98] S. Mandal, R. Paul, S. Sau, A. Chakrabarti, and S. Chattopadhyay, "Efficient dynamic priority based soft error mitigation techniques for configuration memory of FPGA hardware," Microprocessors and Microsystems, vol. 51, pp. 313–330, Jun. 2017, doi: https://doi.org/10.1016/j.micpro.2016.12.003

[99] S. Mandal, A. Chakrabarti, and S. Bodapati, "Clustered error resilient SRAM-based reconfigurable computing platform," IEEE Transactions on Aerospace and

Electronic Systems, vol. 57, no. 3, pp. 1768–1779, Jun. 2021, doi: https://doi.org/10.1109/TAES.2021.3054070

[100] M. Marinescu, "Simple and efficient algorithms for functional RAM testing," 1982 IEEE Test Conference, Philadelphia, (Nov.). IEEE Computer Society, pp. 236-239

[101] M. Matney, A. Vavrin, A. Manis, "Effects of CubeSat Deployments in Low-Earth Orbit," 7th European Conference on Space Debris, April 2017, ESA's Space Operations Centre, Darmstadt, Germany, [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/509/SDC7-paper509.pdf

[102] C. Maxfield, "The Design Warrior's Guide to FPGAs: Devices, Tools and Flows," (1st. ed.), April 2004, Newnes, USA, Paperback ISBN: 9780750676045

[103] Microchip, PolarFire SoC FPGAs. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/system-on-chip-fpgas/polarfire-soc-fpgas

[104] Microchip, ProASIC 3 FPGA. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/fpgas/proasic-3-fpgas

[105] Microchip RTAX-S/SL/DSP Radiation-Tolerant FPGAs. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/radiation-tolerant-fpgas/rtax-s

[106] Microchip, SmartFusion-2 FPGAs. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/system-on-chip-fpgas/smartfusion-2-fpgas

[107] NanoXplore, BRAVE NG-Ultra SoC FPGA. [Online]. Available: https://nanoxplore.org/index.php/product/ng-ultra

[108] NASA, High Performance Spaceflight Computing (HPSC). [Online]. Available: https://www.nasa.gov/directorates/spacetech/game_changing_development/projects/HPSC

[109] NASA Science Mars Exploration, "NASA's Mars 2020 Rover," nasa.gov. https://mars.nasa.gov/resources/21377 (accessed Jan. 21, 2024).

[110] NASA – National Aeronautics and Space Administration, "NASA Space Science Data Coordinated Archive – SAMPEX," nasa.gov. https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1992-038A (accessed Jan. 21, 2024).

[111] P. M. O'Neill, G. D. Badhwar and W. X. Culpepper, "Risk assessment for heavy ions of parts tested with protons," in IEEE Transactions on Nuclear Science, vol. 44, no. 6, pp. 2311-2314, Dec. 1997, doi: https://doi.org/10.1109/23.659052

[112] S. P. Park, D. Lee, and K. Roy, "Soft-error-resilient FPGAs using built-in 2-D Hamming product code," Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp. 248–256, Feb. 2012, doi: https://doi.org/10.1109/TVLSI.2010.2095435

[113] D. Petrick, A. Geist, D. Albaijes, M. Davis, P. Sparacino, G. Crum, R. Ripley, J. Boblitt and T. Flatley, "SpaceCube v2.0 space flight hybrid reconfigurable data processing system," 2014 IEEE Aerospace Conference, Big Sky, MT, USA, March 2014, pp. 1-20, doi: https://doi.org/10.1109/AERO.2014.6836226

[114] D. Petrick, N. Gill, M. Hassouneh, R. Stone, L. Winternitz, L. Thomas, M. Davis, P. Sparacino, and T. Flatley, "Adapting the SpaceCube v2.0 data processing system for mission-unique application requirements," 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Montreal, QC, Canada, 2015, pp. 1-8, doi: https://doi.org/10.1109/AHS.2015.7231153

[115] pgAdmin – PostgreSQL tools. [Online]. Available: https://www.pgadmin.org

[116] C. Poivey, "Radiation effects in space electronics," 6th EIROforum School on Instrumentation, May 2019, ESTEC, Noordwijk, The Netherlands. [Online]. Available: https://indico.cern.ch/event/777129/contributions/3249529/

[117] PostgreSQL: The World's Most Advanced Open Source Relational Database. [Online]. Available: https://www.postgresql.org

[118] H. Quinn, "Radiation effects in reconfigurable FPGAs," Semiconductor Science and Technology, vol. 32, no. 4, 044001, March 2017, doi: https://iopscience.iop.org/article/10.1088/1361-6641/aa57f6

[119] H. Quinn, T. Fairbanks, J. L. Tripp, G. Duran, and B. Lopez, "Single-event effects in low-cost, low-power microprocessors," in Proc. IEEE Radiation Effects Data Workshop (REDW), Paris, France, Jul. 2014, pp. 1–9, doi: https://doi.org/10.1109/REDW.2014.7004596

[120] P. M. B. Rao, M. Ebrahimi, R. Seyyedi, M. B. Tahoori, "Protecting SRAM-based FPGAs against multiple bit upsets using erasure codes," in Proc. 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, Jun. 2014, pp. 1241–1246, doi: https://doi.org/10.1145/2593069.2593191

[121] D. Rudolph, C. Wilson, J. Stewart, P. Gauvin, A. George, H. Lam, G. Crum, M. Wirthlin, A. Wilson, and A. Stoddard, "CSP: A multifaceted hybrid architecture for space computing," Proc. 28th AIAA/USU Conference Small Satellites, Aug. 2014, SSC14-III-3 pp. 1–7. [Online]. Available: https://ntrs.nasa.gov/citations/20150000168

[122] "SAMPEX (Solar Anomalous and Magnetospheric Particle Explorer)," eoportal.org powered by ESA. https://www.eoportal.org/satellite-missions/sampex (accessed Jan. 21, 2024).

[123] A. Sanchez-Macian, P. Reviriego and J. A. Maestro, “Enhanced Detection of Double and Triple Adjacent Errors in Hamming Codes Through Selective Bit Placement,” in IEEE Transactions on Device and Materials Reliability, vol. 12, no. 2, pp. 357-362, June 2012, doi: https://doi.org/10.1109/TDMR.2012.2186965

[124] A. D. Santangelo, “LinkStar-X, An integrated UltraScale+ Based Flight Computer and S-/X-Band Software Defined Radio For Use In LEO and Deep Space,” AIAA SPACE and Astronautics Forum and Exposition, Sep 2017, Orlando, FL, doi: https://doi.org/10.2514/6.2017-5249

[125] A. Sari, V. Vlagkoulis and M. Psarakis, “An Open-source Framework for Xilinx FPGA Reliability Evaluation,” Workshop on Open Source Design Automation (OSDA) 2019, in conjunction with Design, Automation and Test in Europe Conference (DATE), pp 1-6, Florence, Italy, March 2019. [Online]. Available: https://osda.gitlab.io/19/3.2.pdf

[126] D. M. Sawyer and J.I. Vette, “AP8 Trapped Proton Environment For Solar Maximum and Solar Minimum,” National Space Science Data Center / World Data Center A for Rockets and Satellites , 76-06, Dec. 1976, [Online]. Available: https://ntrs.nasa.gov/citations/19770012039

[127] D. Selva and D. Krejci, “A Survey and Assessment of the Capabilities of Cubesats for Earth Observation,” Acta Astronautica, vol. 74, May - June 2012, pp. 50–68, doi: https://doi.org/10.1016/j.actaastro.2011.12.014

[128] F. Siegle, T. Vladimirova, J. Ilstad, and O. Emam, “Mitigation of radiation effects in SRAM-based FPGAs for space applications,” ACM Computing Surveys, vol. 47, no. 2, pp. 1–34, Jan. 2015, doi: https://doi.org/10.1145/2671181

[129] C. De Sio, S. Azimi, L. Bozzoli, B. Du and L. Sterpone, “Radiation-induced Single Event Transient effects during the reconfiguration process of SRAM-based FPGAs,” Microelectronics Reliability, volumes 100–101, Sept 2019, 113342, doi: https://doi.org/10.1016/j.microrel.2019.06.034

[130] C. D. Sio, S. Azimi, L. Sterpone and D. M. Codinachs, “Analysis of Proton-induced Single Event Effect in the On-Chip Memory of Embedded Process,” 2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Austin, TX, USA, Oct 2022, pp. 1-6, doi: https://doi.org/10.1109/DFT56152.2022.9962341

[131] E. G. Stassinopoulos and J. P. Raymond, “The space radiation environment for electronics,” in Proceedings of the IEEE, vol. 76, no. 11, pp. 1423-1442, Nov 1988, doi: https://doi.org/10.1109/5.90113

[132] A. Stoddard, A. Gruwell, P. Zabriskie, and M. J. Wirthlin, “A hybrid approach to FPGA configuration scrubbing,” IEEE Transactions on Nuclear Science, vol. 64, no. 1, pp. 497–503, Jan. 2017, doi: https://doi.org/10.1109/TNS.2016.2636666

[133] G. Swift, "Micro-Latchup in 7-Series FPGAs and Other Ongoing XRTC Investigations," Single Event Effects (SEE) Symposium and Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, San Diego, USA, May 2015. [Online]. Available: https://www.seemapld.org/archive/2015/0519TUE/1620%20-%20Swift.pdf

[134] G. M. Swift, S. E. Stone, S. E. García, K. W. Wray, W. J. Rowe, K. H. Pfau, R. Liu, J. Holden, A. Angeles, B. L. Willits, K. P. Robinson, A. Pérez-Celis and M. J. Wirthlin, "Dynamic SEE Testing of Selected Architectural Features of Xilinx 28 nm Virtex-7 FPGAs," 17th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Geneva, Switzerland, Oct. 2017, pp. 1-6, doi: https://doi.org/10.1109/RADECS.2017.8696210

[135] L. A. Tambara F. L. Kastensmidt, N. H. Medina, N. Added, V. A. P. Aguiar, F. Aguirre, E. L. A. Macchione, and M. A. G. Silveira, "Heavy ions induced single event upsets testing of the 28 nm Xilinx Zynq-7000 all programmable SoC," in Proc. IEEE Radiation Effects Data Workshop (REDW), Boston, MA, USA, Jul. 2015, pp. 29–34, doi: https://doi.org/10.1109/REDW.2015.7336716

[136] L. A. Tambara, A. Akhmetov, D. V. Bobrovsky, and F. L. Kastensmidt, "On the characterization of embedded memories of zynq-7000 all programmable SoC under single event upsets induced by heavy ions and protons," in Proc. 15th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Moscow, Russia, Sep. 2015, pp. 126–129, doi: https://doi.org/10.1109/RADECS.2015.7365643

[137] L. A. Tambara, P. Rech, E. Chielle, J. Tonfat, and F. L. Kastensmidt, "Analyzing the impact of radiation-induced failures in programmable SoCs," IEEE Transactions on Nuclear Science, vol. 63, no. 4, pp. 2217–2224, Aug. 2016, doi: https://doi.org/10.1109/TNS.2016.2522508

[138] F. Tommasino, M. Rovituso, S. Fabiano, S. Piffer, C. Manea, S. Lorentini, S. Lanzone, Z. Wang, M. Pasini, W.J. Burger, C. La Tessa, E. Scifoni, M. Schwarz, M. Durante, "Proton beam characterization in the experimental room of the Trento Proton Therapy facility," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, De-tectors and Associated Equipment, vol. 869, pp. 15-20, Oct. 2017, doi: https://doi.org/10.1016/J.NIMA.2017.06.017

[139] J. Tonfat, F. L. Kastensmidt, P. Rech, R. Reis, and H. M. Quinn, "Analyzing the effectiveness of a frame-level redundancy scrubbing technique for SRAM-based FPGAs," IEEE Transactions on Nuclear Science, vol. 62, no. 6, pp. 3080–3087, Dec. 2015, doi: https://doi.org/10.1109/TNS.2015.2489601

[140] J. Tonfat, F. L. Kastensmidt, L. Artola, G. Hubert, N. H. Medina, N. Added, V. A. P. Aguiar, F. Aguirre, E. L. A. Macchione and M. A. G. Silveira, "Analyzing the Influence of the Angles of Incidence and Rotation on MBU Events Induced by Low

LET Heavy Ions in a 28-nm SRAM-Based FPGA," in IEEE Transactions on Nuclear Science, vol. 64, no. 8, pp. 2161-2168, Aug. 2017, doi: https://doi.org/10.1109/TNS.2017.2727479

[141] TRAD, OMERE Software, [Online]. Available: https://www.trad.fr/en/space/omere-software/

[142] G. Tsiligiannis, L. Dilillo, V. Gupta, A. Bosio, P. Girard, A. Virazel, H. Puchner, A. Bosser, A. Javanainen, A. Virtanen, C. Frost, F. Wrobel, L. Dusseau and F. Saigné, "Dynamic Test Methods for COTS SRAMs," in IEEE Transactions on Nuclear Science, vol. 61, no. 6, pp. 3095-3102, Dec. 2014, doi: https://doi.org/10.1109/TNS.2014.2363123

[143] A.J. Tylka, J. H. Adams, Jr., P. R. Boberg, B. Brownstein, W. F. Dietrich, E. O. Flueckiger, E. L. Petersen, M. A. Shea, D. F. Smart, and E. C. Smith, "CREME96: A Revision of the Cosmic Ray Effects on Micro-Electronics Code," in IEEE Transactions on Nuclear Science, vol. 44, no. 6, pp. 2150-2160, Dec. 1997, doi: https://doi.org/10.1109/23.659030

[144] University of Michigan - Electrical Engineering and Computer Science (EECS) Department, "MiBench Version 1.0," [Online]. Available: https://vhosts.eecs.umich.edu/mibench

[145] University of Piraeus - Embedded Systems Lab, "FREtZ (FPGA Reliability Evaluation through JTAG)," [Online]. Available: https://github.com/unipieslab/FREtZ

[146] R. Velazco, "Error-rate prediction for programmable circuits: methodology, tools and studied cases," Proc. SPIE 8725, Micro- and Nanotechnology Sensors, Systems, and Applications V, 872519, May 2013; doi: https://doi.org/10.1117/12.2015391

[147] S. Venkataraman, R. Santos, A. Das, and A. Kumar, "A bit-interleaved embedded Hamming scheme to correct single-bit and multi-bit upsets for SRAM-based FPGAs," in Proc. 24th International Conference on Field Programmable Logic and Applications (FPL), Munich, Germany, Sep. 2014, pp. 1–4, doi: https://doi.org/10.1109/FPL.2014.6927385

[148] J. I. Vette, "The AE-8 trapped electron model environment," vol, 91. no. 24, National Space Science Data Center (NSSDC), World Data Center A for Rockets and Satellites (WDC-AR &S), National Aeronautics and Space Administration, Goddard Space Flight Center, Nov. 1991, [Online]. Available: https://ntrs.nasa.gov/citations/19920014985

[149] V. Vlagkoulis, A. Sari, J. Vrachnis, G. Antonopoulos, N. Segkos, M. Psarakis, A. Tavoularis, G. Furano, C. Boatella-Polo, C. Poivey, V. Ferlet-Cavrois, M. Kastriotou, P. Fernandez Martinez and R. G. Alía, "Analysis of the Single Event

Upsets in the Programmable Logic of 28 nm Xilinx Zynq-7000 FPGA due to Heavy Ion Irradiation," Single Event Effects (SEE) Symposium and Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, San Diego, USA, May 2019. [Online]. Available: https://www.seemapld.org/archive/2019/0522_WED/1040%20-%20SEE-MAPLD19_Psarakis_Zynq_Radiation.pdf

[150] V. Vlagkoulis, A. Sari, J. Proko, D. Zografakis, M. Psarakis, A. Tavoularis, G. Furano, C. Boatella-Polo, C. Poivey, V. Ferlet-Cavrois, M. Kastriotou, P. Fernandez Martinez and R. G. Alía, "Configuration Memory Scrubbing of the Xilinx Zynq-7000 FPGA using a Mixed 2-D Coding Technique," 2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Montpellier, France, September 2019, doi: https://doi.org/10.1109/RADECS47380.2019.9745693

[151] V. Vlagkoulis, A. Sari, J. Vrachnis, G. Antonopoulos, N. Segkos, M. Psarakis, A. Tavoularis, G. Furano, C. Boatella-Polo, C. Poivey, V. Ferlet-Cavrois, M. Kastriotou, P. Fernandez Martinez, R. G. Alía, K.-O. Voss and C. Schuy, "Single Event Effects Characterization of the Programmable Logic of Xilinx Zynq-7000 FPGA Using Very/Ultra High-Energy Heavy Ions," in IEEE Transactions on Nuclear Science, vol. 68, no. 1, pp. 36-45, January 2021, doi: https://doi.org/10.1109/TNS.2020.3033188

[152] V. Vlagkoulis, A. Sari, G. Antonopoulos, M. Psarakis, A. Tavoularis, G. Furano, C. Boatella-Polo, C. Poivey, V. Ferlet-Cavrois, M. Kastriotou, P. Fernandez Martinez and R. G. Alía, "Configuration Memory Scrubbing of SRAM-Based FPGAs Using a Mixed 2-D Coding Technique," in IEEE Transactions on Nuclear Science, vol. 69, no. 4, pp. 871-882, April 2022, doi: https://doi.org/10.1109/TNS.2022.3151977

[153] V. Vlagkoulis, D. Agiakatsikas, A. Sari, I. Souvatzoglou, G. Antonopoulos, M. Psarakis, C. Boatella-Polo, A. Costantino, V. Gupta, A. Tavoularis, G. Furano, C. Poivey, V. Ferlet-Cavrois, T. Wagner and H. Wojciech, "Estimating the Error Rates of Xilinx Zynq-7000 APSoCs in Space Radiation Environments," in 5th SpacE FPGA Users Workshop (SEFUW), ESTEC, Noordwijk, The Netherlands, March 2023. [Online]. Available: https://indico.esa.int/event/439/contributions/7941/attachments/5183/8216/SEFUW_v1.1.pdf

[154] C. Wilson, J. Stewart, P. Gauvin, J. MacKinnon, J. Coole, J. Urriste, A. George, G. Crum, E. Timmons, J. Beck, T. Flatley, M. Wirthlin, A. Wilson and A. Stoddard, "CSP Hybrid Space Computing for STP-H5/ISEM on ISS", in Proc. of the AIAA/USU Conference on Small Satellites, 29th Annual AIAA/USU Conference on Small Satellites, August 2015. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2015/all2015/21/

[155] M. Wirthlin, D. Lee, G. Swift and H. Quinn, "A Method and Case Study on Identifying Physically Adjacent Multiple-Cell Upsets Using 28-nm, Interleaved and SECDED-Protected Arrays," in IEEE Transactions on Nuclear Science, vol. 61, no. 6, pp. 3080-3087, Dec. 2014, doi: https://doi.org/10.1109/TNS.2014.2366913

[156] M. Wirthlin, "High-Reliability FPGA-Based Systems: Space, High-Energy Physics, and Beyond," in Proceedings of the IEEE, vol. 103, no. 3, pp. 379-389, March 2015, doi: https://doi.org/10.1109/JPROC.2015.2404212

[157] M. A. Xapsos, G. P. Summers, J. L. Barth, E. G. Stassinopoulos and E. A. Burke, "Probability Model for Cumulative Solar Proton Event Fluences," IEEE Transactions on Nuclear Science, vol. 47, no. 3, pp. 486-490, June 2000, doi: https://doi.org/10.1109/23.856469

[158] W. Yang, X. Du, C. He, S. Shi, L. Cai, N. Hui, G. Guo, C. Huang, "Microbeam heavy-ion single-event effect on Xilinx 28-nm system on chip," IEEE Transactions on Nuclear Science, vol. 65, no. 1, pp. 545–549, Jan. 2018, doi: https://doi.org/10.1109/TNS.2017.2776244

[159] W. Yang, X. Du, J. Guo, J. Wei, G. Du, C. He, W. Liu, S. Shen, C. Huang, Y. Li, Y. Fan, "Preliminary single event effect distribution investigation on 28 nm SoC using heavy ion microbeam," Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, vol. 450, pp. 323–326, Jul. 2019, doi: https://doi.org/10.1016/j.nimb.2018.09.038