Reliability Analysis of TMR Configurations in SRAM-Based FPGAs Using Static Evaluation

Reliability Analysis of TMR Configurations in SRAM-Based FPGAs Using Static Evaluation

Invited Paper

Davide Nicolini
Politecnico di Torino
Turin, Italy
davide.nicolini@polito.it

Corrado De Sio
Politecnico di Torino
Turin, Italy
corrado.desio@polito.it

Luca Sterpone
Politecnico di Torino
Turin, Italy
luca.sterpone@polito.it

Luca Baldanzi
IngeniArs S.r.l.
Pisa, Italy
luca.baldanzi@ingeniars.com

Leonardo Ligabue
IngeniArs S.r.l.
Pisa, Italy
leonardo.ligabue@ingeniars.com

Alessandro Zubani
IngeniArs S.r.l.
Pisa, Italy
alessandro.zubani@ingeniars.com

Filomena Decuzzi
ESA
Noordwijk, Netherlands
filomena.decuzzi@esa.int

Abstract

SRAM-based FPGAs have been increasingly adopted in recent years due to their high performance and power efficiency. As a result, they are also being considered for space missions. However, SRAM-based FPGAs are inherently vulnerable to radiation effects in space environments. High-energy particles can induce bit flips in the configuration memory, potentially altering the functionality of the implemented design. To address this issue, several mitigation techniques have been proposed, among which Triple Modular Redundancy (TMR) is the most widely used. Although effective, TMR significantly increases resource utilization. This overhead may limit its applicability on certain devices and can also raise the probability of radiation-induced faults affecting active resources. In this paper, we investigate different levels of granularity for applying TMR in custom designs implemented on a Xilinx KU060 FPGA, with the aim of balancing fault tolerance and resource efficiency.

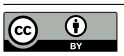
CCS Concepts

• **Hardware → Redundancy**.

Keywords

FPGA, TMR, SEU, Reliability

ACM Reference Format:
Davide Nicolini, Corrado De Sio, Luca Sterpone, Luca Baldanzi, Leonardo Ligabue, Alessandro Zubani, and Filomena Decuzzi. 2026. Reliability Analysis of TMR Configurations in SRAM-Based FPGAs Using Static Evaluation: Invited Paper. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF Companion '26), May 19–21, 2026, Catania, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3801488.3808245



CF Companion '26, Catania, Italy

ACM ISBN 979-8-4007-2569-2/26/05
https://doi.org/10.1145/3801488.3808245

1 Introduction

Field-Programmable Gate Arrays (FPGAs) have become a key technology for implementing high-performance and energy-efficient digital systems [8]. In particular, SRAM-based FPGAs are widely adopted due to their flexibility, reconfigurability, and superior performance compared to traditional radiation-hardened devices [11]. These characteristics make them attractive for space applications, where adaptability and computational capability are increasingly important.

However, the use of SRAM-based FPGAs in radiation-rich environments introduces significant reliability challenges. In space, high-energy particles can cause Single Event Upsets (SEUs), leading to bit flips in the configuration memory. Since the configuration memory defines the implemented hardware, such upsets may result in transient or permanent malfunctions of the system [1]. In addition, the increasing integration density of modern FPGA devices further exacerbates this issue, as a larger number of configuration bits become susceptible to radiation effects, potentially increasing the likelihood of functional failures.

To mitigate these effects, several fault-tolerance techniques have been proposed. Among them, Triple Modular Redundancy (TMR) is one of the most commonly adopted approaches, as it allows correct system operation even in the presence of faults by replicating logic and using majority voting. Despite its effectiveness, TMR comes with a substantial cost in terms of resource utilization, power consumption, and design complexity. Moreover, the increased use of resources may itself raise the susceptibility of the design to radiation-induced faults [5]. For this reason, the way TMR is applied—particularly in terms of granularity, partitioning, and voter placement—plays a crucial role in determining the overall reliability of the system.

In this context, the evaluation of mitigation strategies requires accurate and efficient analysis methodologies. While radiation testing campaigns and dynamic fault injection provide realistic insights into system behavior, they are often time-consuming and costly.

Alternatively, static analysis techniques enable the estimation of design sensitivity to SEUs by examining the relationship between configuration memory and the implemented hardware resources. These approaches allow for rapid exploration of design solutions and early identification of critical vulnerabilities, supporting informed design decisions prior to experimental validation.

This work is conducted within a collaborative project with the European Space Agency (ESA), aimed at assessing the reliability of a custom design implemented on a dedicated board featuring a Xilinx KU060 FPGA. In this context, we explore how different levels of granularity in the application of TMR affect system reliability. The analysis is carried out through static evaluation of the implemented design, enabling a comparative assessment of the fault tolerance achieved under different redundancy strategies. In particular, the study investigates how coarse-grain and more distributed TMR approaches, combined with different voter placement schemes, influence the trade-off between reliability improvement and resource overhead.

2 Related Work

The reliability of SRAM-based FPGAs in radiation environments has been extensively studied, particularly due to their susceptibility to Single Event Upsets (SEUs) affecting configuration memory. Several surveys provide a comprehensive overview of fault mechanisms and mitigation techniques, highlighting redundancy and configuration scrubbing as the dominant approaches for space applications[9].

Triple Modular Redundancy (TMR) is one of the most widely adopted techniques for SEU mitigation in FPGA-based systems. It relies on the replication of logic and the use of majority voting to mask faults, significantly improving system reliability under radiation exposure[4]. Experimental studies have demonstrated that TMR can provide substantial reliability gains, although at the cost of increased resource utilization, power consumption, and performance degradation[12].

Despite its effectiveness, several works have shown that the reliability improvement achieved by TMR strongly depends on implementation details, such as physical placement and routing[10]. In particular, common-mode failures caused by shared configuration resources can reduce the effectiveness of redundancy, motivating placement-aware techniques to improve fault isolation between replicas[2]. Similarly, the distribution of clock and routing resources has been identified as a critical factor influencing the SEU tolerance of TMR-protected designs [13].

Another important aspect addressed in the literature is the granularity and partitioning of TMR. Different levels of redundancy—from coarse-grain replication of entire modules to fine-grain or distributed TMR—have been explored to balance reliability and overhead. Studies based on reliability modeling and fault injection have shown that the number of partitions and the placement of voters significantly affect system robustness, enabling trade-offs between area, performance, and fault coverage[6] . In particular, the positioning and frequency of voters play a key role in preventing fault propagation and improving overall effectiveness [7].

To evaluate the impact of SEUs and the effectiveness of mitigation techniques, both dynamic and static analysis approaches have been proposed. Dynamic methods rely on fault injection or radiation testing to measure the functional sensitivity of a design, often expressed in terms of error rates or cross-sections[3] . In contrast, static analysis techniques aim to estimate design vulnerability by analyzing the mapping between configuration memory and hardware resources, enabling faster and more exhaustive evaluation without requiring physical testing. These approaches are particularly useful during early design stages, where they can guide the selection and optimization of mitigation strategies.

In this context, the present work focuses on the evaluation of TMR at different levels of granularity through static analysis, providing a comparative assessment of reliability and resource trade-offs on a modern SRAM-based FPGA platform.

3 AMATERASU project

The AMATERASU project, initiated in August 2024 in collaboration with Ingeniars and the European Space Agency (ESA), is focused on the reliability assessment and enhancement of SRAM-based FPGA designs for radiation-prone environments. The project involves the development of a custom digital design and a dedicated hardware platform integrating a Xilinx KU060 FPGA.

The adopted methodology investigates the application of Triple Modular Redundancy (TMR) at different levels of granularity within the design. For each TMR configuration, the implemented system is evaluated through both static and dynamic analyses. Static analysis is used to estimate the sensitivity of the configuration memory to Single Event Upsets (SEUs), for instance by identifying critical bits and analyzing fault propagation paths. Dynamic analysis is performed through fault injection campaigns, allowing the evaluation of the functional impact of configuration upsets on the system

The reliability metrics obtained from these analyses—such as error rates, fault coverage, and sensitivity cross-sections—are subsequently validated through radiation testing campaigns (e.g., proton or heavy-ion irradiation). This step enables the correlation between predicted and observed behavior under realistic radiation conditions.

Based on the experimental results, advanced mitigation techniques are investigated to improve the robustness of the design, potentially including optimized redundancy schemes and selective hardening strategies. The refined implementations are then re-evaluated and subjected to further radiation campaigns, following an iterative design–analysis–validation loop aimed at achieving an optimal trade-off between reliability and resource utilization

4 Methodology

To assess the reliability of a specific design implemented in an SRAM-based FPGA, a dedicated analysis tool is employed. The tool performs static analysis of the implemented design by examining the utilized hardware resources and their mapping onto the configuration memory. VeriPlace is a static analysis framework developed to evaluate the fault tolerance of FPGA designs. It targets AMD SRAM-based devices (e.g., Zynq-7020, Kintex UltraScale) and relies on the analysis of the implemented netlist together with its mapping onto the configuration memory. The tool operates by importing the placed-and-routed design and building a graph-based internal

representation of the architecture, referred to as the Physical Design Description (PDD). In this model, each node corresponds to a specific FPGA resource, including LUTs, flip-flops, BRAMs, PIPs, clock networks, and routing elements. The configuration bitstream is decoded in detail to establish a direct association between each CRAM bit and its corresponding hardware resource.

Exploiting this bit-to-resource mapping, VeriPlace enables the simulation of radiation-induced Single Event Upsets (SEUs) by injecting bit flips into the configuration memory and evaluating their impact on the design structure. In the case of TMR-protected implementations, the tool incorporates mechanisms to identify faults that simultaneously affect multiple redundant domains, potentially defeating the majority voting scheme. Such faults are classified as critical, as they may propagate to the outputs despite redundancy.

The analysis flow consists of several stages. First, a virtual configuration memory database is generated, describing how the design is mapped onto the FPGA configuration, with each bit annotated in terms of location, function, and associated resource. Next, a complete architectural mapping is established by linking each configuration bit to its corresponding element in the netlist. Fault injection is then performed by introducing single or multiple bit flips in the CRAM, with the resulting modifications propagated through the PDD graph to determine their effect on observable outputs. This approach also supports the analysis of multiple-bit upsets (MBUs) by considering cumulative architectural changes.

Reliability evaluation is carried out through a Monte Carlo campaign, where combinations of SEUs with increasing multiplicity (from 1 up to 500 simultaneous bit flips) are injected over up to 100,000 iterations per configuration. For each case, the tool determines whether the fault leads to an observable error at the output. At the end of the simulation, a report is generated, summarizing the fault classification:

- Accumulated bits: Total number of injected SEUs across all iterations.
- Misses: Faults that hit unused or inactive configuration bits.
- Errors: Faults that propagate to at least one output node.
- Filtered: Faults that affect active resources but whose effects are masked or contained.

Overall, this static framework provides an effective methodology for pre-characterizing the radiation sensitivity of FPGA designs prior to irradiation campaigns. It supports informed design choices and mitigation strategies, and complements dynamic fault injection approaches by enabling exhaustive fault coverage and fast evaluation of resource-level vulnerabilities.

5 Experimental setup

5.1 Benchmark overview

The benchmark design is based on the CCSDS 131.2-B transmitter IP core developed by Ingeniars. The core processes a predefined set of input data consisting of 1024 patterns of 32 bits each, which are cyclically fetched from an internal memory buffer and fed to the transmitter. For each input pattern, the core generates an output stream of 198 bits. The produced output is transmitted to a secondary board connected via FMC, which operates in parallel to further process the data. However, within the scope of this work, only the board hosting the target design is considered for reliability analysis.

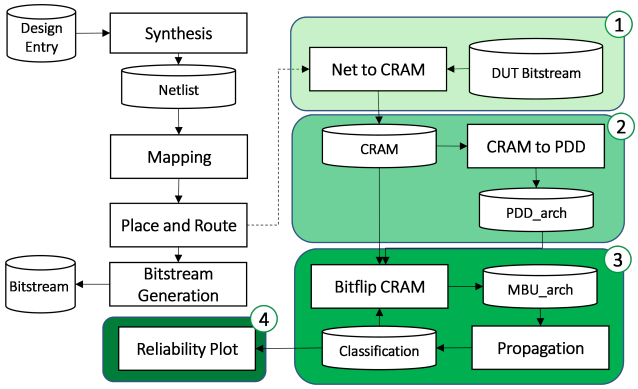


Figure 1: VeriPlace fault simulation flow

5.2 Benchmark versions

Different levels of TMR granularity are evaluated on the benchmark design, including a baseline implementation without redundancy. In particular, two TMR strategies are considered: a coarse-grain approach, in which the entire design is triplicated, and a modular-level approach, where redundancy is applied selectively at the module level within the design. A conceptual representation is shown in Fig. 2. It should be noted that the figure illustrates both coarse-grain and modular-level TMR implementations; however, only one level is applied per design, as combining both approaches was not feasible due to resource constraints. For each TMR configuration, two voter placement strategies are considered. In the first approach, a single voter is inserted at the final output of the triplicated design. In the second approach, voters are distributed across the design, such that each triplicated module is followed by a voter whose output feeds the subsequent triplicated stage. An example is shown in Fig. 3. This latter configuration enables error correction at intermediate points in the data path, potentially limiting fault propagation across the design.

Table 1: Resource utilization for the evaluated design configurations

Resource	Plain	Coarse TMR	Modular TMR
LUT	65600 (19.78%)	65091 (19.62%)	65563 (19.77%)
LUTRAM	1692 (1.15%)	1692 (1.15%)	1704 (1.16%)
FF	120363 (18.14%)	120128 (18.11%)	120751 (18.20%)
BRAM	634.5 (58.75%)	514.50 (47.64%)	514.50 (47.64%)
DSP	312 (11.30%)	312 (11.30%)	312 (11.30%)
IO	60 (9.62%)	56 (8.97%)	56 (8.97%)

6 Experimental analisys

This section reports preliminary results obtained from three design configurations analyzed using VeriPlace. The first configuration

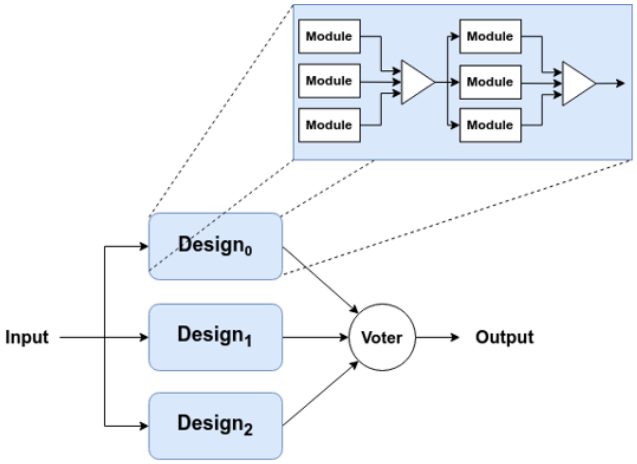


Figure 2: Different level of TMR implementation

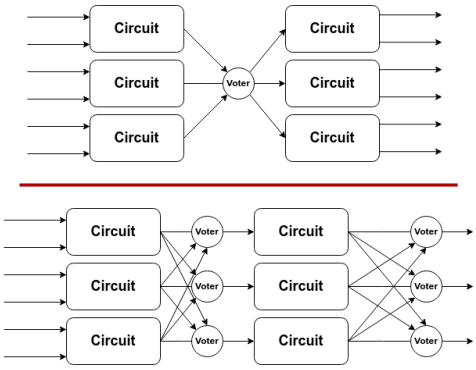


Figure 3: Implementation with one voter vs triple voters

corresponds to the baseline design, implemented without any fault-tolerance mechanisms. The second introduces a reliability-oriented solution based on coarse-grain TMR applied at the top level, combined with a single voter at the output. The third configuration extends this approach by incorporating a distributed voting scheme, where the coarse-grain TMR is complemented by multiple voting stages. Additional reliability-enhanced designs are currently under development and will be evaluated using the same methodology to ensure a fair and consistent comparison.

The analysis is performed by progressively injecting configuration memory bit flips, increasing the number of SEUs up to a maximum of 400. For each accumulation level, 60,000 fault injection experiments are carried out, with injection locations randomly selected across the design.

For each configuration, the tool generates reliability curves that account for the effectiveness of TMR. In particular, it distinguishes between the behavior of the circuit with and without the voting stage. For instance, if a fault affects only one replica while the others remain correct, the configuration without voting is classified as erroneous, since the incorrect value propagates to the output.

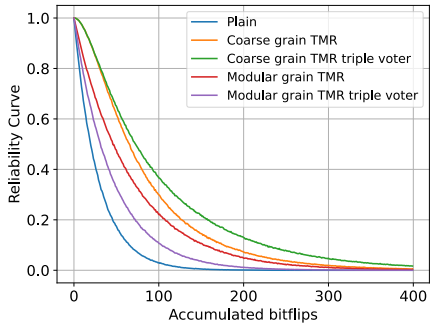


Figure 4: Reliability curves for each design version

Conversely, when the voting mechanism is enabled, the same condition is masked by majority voting and is therefore not counted as a failure.

Fig. 4 summarizes the results obtained with VeriPlace, comparing all design configurations, including both single- and triple-voter implementations, as well as the baseline (plain) version. Among the evaluated strategies, the coarse-grain TMR with distributed (triple) voting achieves the best reliability performance, whereas the modular-level TMR with triple voters exhibits the lowest effectiveness.

This behavior can be explained by the resource utilization characteristics of the modular-level approach (as reported in Table 1), which increases further when triple voters are introduced. Although the overall resource overhead is not dramatically higher compared to the coarse-grain implementation, the finer granularity leads to a wider distribution of utilized resources across the device. This results in a larger set of configuration bits actively contributing to the design, thereby increasing the probability that radiation-induced faults affect critical portions of the circuit.

Consequently, while distributed voting improves fault masking capabilities, in the modular-level design this benefit does not fully compensate for the increased exposure of sensitive configuration bits.

In contrast, the coarse-grain approach concentrates redundancy at a higher level of abstraction, requiring fewer additional structures such as voters. In this case, triplicating the voter introduces only a limited overhead while still improving fault masking, leading to a net gain in reliability.

Fig. 6 focuses on the two coarse-grain TMR configurations, comparing each design with and without the voting stage. When the voter is disabled, simple replication of the module leads to a degradation in reliability, resulting in performance that is even worse than the baseline design. These results underline the critical role of the voting stage, demonstrating that redundancy alone is insufficient and that effective fault masking through voting is essential for achieving reliability improvements with TMR.

Fig. 5 presents the corresponding results for the modular-level TMR configurations, comparing each design with and without the voting stage. Similarly to the coarse-grain case, disabling the voter leads to a significant degradation in reliability, confirming that replication alone is not sufficient to ensure fault tolerance. When

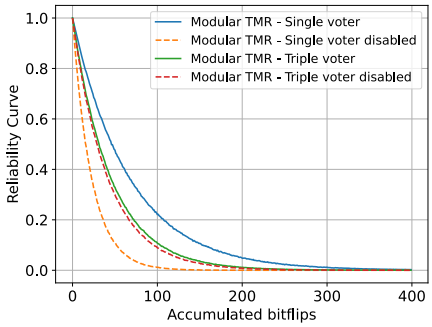


Figure 5: Reliability curves for modular grain TMR

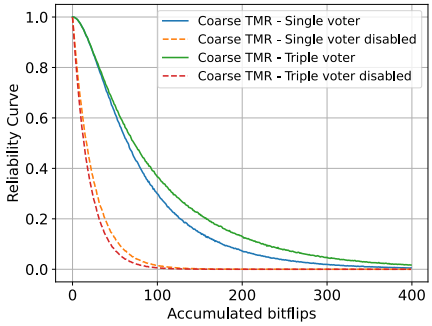


Figure 6: Reliability curves for coarse grain TMR

the voting stage is enabled, an improvement is observed; however, the overall reliability remains limited compared to the coarse-grain implementations. This behavior is mainly due to the higher resource utilization of the modular-level approach, which increases the number of sensitive configuration bits. As a result, even with distributed voting, the benefits of redundancy are partially offset by the increased vulnerability of the design.

A further observation is that, in the coarse-grain approach, disabling the voter results in a more pronounced degradation in reliability compared to the modular-level case. This suggests that, at higher levels of granularity, the overall robustness is strongly dependent on the presence of an effective voting stage. Conversely, as the granularity decreases, part of the robustness is inherently provided by the structural partitioning of the design, reducing, though not eliminating,the relative impact of the voting mechanism.

7 Conclusions

This work investigated the impact of TMR granularity and voter placement on the reliability of an SRAM-based FPGA design, within the context of the AMATERASU project. Through static analysis using the VeriPlace framework, multiple design configurations were evaluated, including baseline, coarse-grain TMR, and modular-level TMR implementations, with both single and distributed voting schemes.

The results confirm that TMR improves reliability against radiation-induced faults, but its effectiveness depends on how redundancy is implemented. Coarse-grain TMR with distributed voting provides the best performance, combining strong fault masking with limited overhead, whereas modular-level TMR increases resource usage and sensitivity to SEUs, reducing its overall benefit.

The study also highlights the critical role of voting: redundancy alone is insufficient, especially at coarse granularity, while finer-grain designs offer some intrinsic robustness but still rely on effective voting. Overall, optimizing TMR requires balancing granularity, voter placement, and resource utilization, with static analysis proving useful for exploring these trade-offs. Future work will extend the analysis and validate the results through radiation testing.

Acknowledgments

This research is part of AMATERASU (Design Mitigation Techniques Evaluation Of Advanced COTS FPGAS For ESA Missions), project funded by the European Space Agency (ESA). The project is conducted in collaboration with Politecnico di Torino, ESA, and Ingeniars S.r.l. The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency.

References

[1] G.-H. Asadi and M.B. Tahoori. 2005. Soft error mitigation for SRAM-based FPGAs. In *23rd IEEE VLSI Test Symposium (VTS'05)*. 207–212. doi:10.1109/VTS.2005.75

[2] Matthew Cannon, Andrew Keller, and Michael Wirthlin. 2018. Improving the Effectiveness of TMR Designs on FPGAs with SEU-Aware Incremental Placement. In *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. 141–148. doi:10.1109/FCCM.2018.00031

[3] Keller et al. 2018. Dynamic SEU Sensitivity of Designs on Two 28-nm SRAM-Based FPGA Architectures. *IEEE Transactions on Nuclear Science* 65, 1 (2018), 280–287. doi:10.1109/TNS.2017.2772288

[4] Ostler et al. 2009. SRAM FPGA Reliability Analysis for Harsh Radiation Environments. *IEEE Transactions on Nuclear Science* 56, 6 (2009), 3519–3526. doi:10.1109/TNS.2009.2033381

[5] Pratt et al. 2007. Fine-grain SEU mitigation for FPGAs using Partial TMR. In *2007 9th European Conference on Radiation and Its Effects on Components and Systems*. 1–8. doi:10.1109/RADECS.2007.5205468

[6] Khaza Anuarul Hoque, Otmane Ait Mohamed, and Yvon Savaria. 2019. Dependability modeling and optimization of triple modular redundancy partitioning for SRAM-based FPGAs. *Reliability Engineering & System Safety* 182 (Feb. 2019), 107–119. doi:10.1016/j.ress.2018.10.011

[7] F. Lima Kastensmidt, L. Sterpone, L. Carro, and M. Sonza Reorda. 2007. On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs. arXiv:0710.4688 [cs.AR] https://arxiv.org/abs/0710.4688

[8] Ian Kuon and Jonathan Rose. 2007. Measuring the Gap Between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26, 2 (2007), 203–215. doi:10.1109/TCAD.2006.884574

[9] Zhe Liu, Zukun Lu, Long Huang, Zhiwei Yao, Zhaojun Lu, and Jiliang Zhang. 2024. Recent advances on reliability of FPGAs in a radiation environment. *Microelectronics Journal* 148 (2024), 106176. doi:10.1016/j.mejo.2024.106176

[10] Davide Nicolini, Corrado De Sio, and Eleonora Vacca. 2024. Enhancing the Robustness of System on FPGA by Routing Isolation. In *Proceedings of the 21st ACM International Conference on Computing Frontiers* (Ischia, Italy) *(CF '24)*. Association for Computing Machinery, New York, NY, USA, 326–327. doi:10.1145/3649153.3653000

[11] Stephen M. Trimberger. 2015. Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology. *Proc. IEEE* 103, 3 (2015), 318–331. doi:10.1109/JPROC.2015.2392104

[12] Andrew E. Wilson, Michael Wirthlin, and Nathan G. Baker. 2023. Neutron Radiation Testing of RISC-V TMR Soft Processors on SRAM-Based FPGAs. *IEEE Transactions on Nuclear Science* 70, 4 (2023), 603–610. doi:10.1109/TNS.2023.3235582

[13] Jian Yu, Chang Cai, Bingxu Ning, Shuai Gao, Tianqi Liu, Liewei Xu, Mingjie Shen, and Jun Yu. 2021. Design and verification of multiple SEU mitigated circuits on SRAM-based FPGA system. *Microelectronics Reliability* 126 (2021), 114340. doi:10.1016/j.microrel.2021.114340