Failure rate analysis of radiation tolerant design techniques on SRAM-based FPGAs

Failure Rate Analysis of Radiation Tolerant Design Techniques on SRAM-based FPGAs

E. Vacca, S. Azimi, L. Sterpone

Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy

Abstract

Space applications using SRAM-based FPGA devices demand an accurate evaluation of high-energy radiation particle effects on the design functionality even when mitigation techniques are adopted. In this work, we evaluated the failure rate of different layout solutions of redundancy-based radiation tolerant design. Experimental results achieved thanks to fault injection campaigns and proton radiation tests on different radiation tolerant design implementations demonstrate that an isolation-based redundancy layout provides more than one order of magnitude radiation tolerance capabilities than a commercial solution.

1. Introduction

In the last decade, reconfigurable devices such as SRAM-based Field Programmable Gate Arrays (FPGA) become largely attractive to the aerospace industry, especially for their enhanced performance characteristics and on-site reconfiguration capabilities. However, these devices are particularly sensitive to space radiation-induced effects [1][2][3] since the hitting of a high-energy particle with the device's Configuration Random Access Memory (CRAM) can cause one or multiple Single Event Upset (SEU).

Common approaches to ensure a working design in FPGA in case of errors within the CRAM cells consist of hardening through the adoption of redundancy [4] [5] and performing periodic scrubbing of its content [6]. Nonetheless, high-level hardening techniques are often not sufficient to achieve good fault tolerance capabilities, since cross-domain failures are affecting the implemented circuit [7].

Commercial FPGA design tools aim to realize feasible, performant but not necessarily reliable solutions due to the increasing resource sharing of the circuit mapped on the FPGA according to performance and area optimization rules. This aspect turns on mapping several circuit modules of a design, even functionally independent or related to redundant modules to share the programmable routing or hard-wired interconnection resources and Configurable Logic Block (CLB) resources. Therefore, if a radiation-induced upset occurs in a memory cell that configures a resource used to implement multiple modules, there will be multiple potential errors that will propagate in the design. This increases the probability of a system failure, or, in the worst case of redundant design, provoking a cross-domain failure nullifying any redundancy or voting structure [6].

The main contribution of this work is to investigate the impact of reliability-oriented layout implementation rules, neglected during the application of high-level mitigation solutions.

To emphasize that it is not sufficient to adopt high-level fault-tolerant design techniques such as Triple Modular Redundancy (TMR), we developed a design flow to physically map circuits already hardened with architectural TMR. Furthermore, we evaluated and compared the failure rates of the proposed solutions by means of fault injection campaigns as well as proton radiation tests performed at the PSI facility.

The experimental results have shown that the efficiency of the hardening technique varies depending on the arrangement of resources on the device. The adoption of an isolation policy per TMR domain brings benefits to the failure rate for all energies and shows a clear improvement of an order of magnitude for high energies compared to a commercial solution without constraints. On the other hand, a high sharing solution has a failure rate trend equal to the original solution for low energies and a drastic deterioration at high energies.

2. The Developed Custom Layout Methodology

One of the most used techniques to deal with radiation-induced effects in SRAM-based FPGA is the TMR technique since it has a good trade-off between applicative simplicity and tolerance of the design. However, this technique is not infallible since it has intrinsic problems due to the creation of a Single Point of Failure, as in the voting stage [8], and to the presence of cross-domain failures which are mainly due to the layout implementation.

For this reason, solutions such as the one presented in [9][10] refer to circuit implementation level changes, where the idea is to control placement and routing in

* Corresponding author: luca.sterpone@polito.it
Tel: +39 (011) 090 7009;

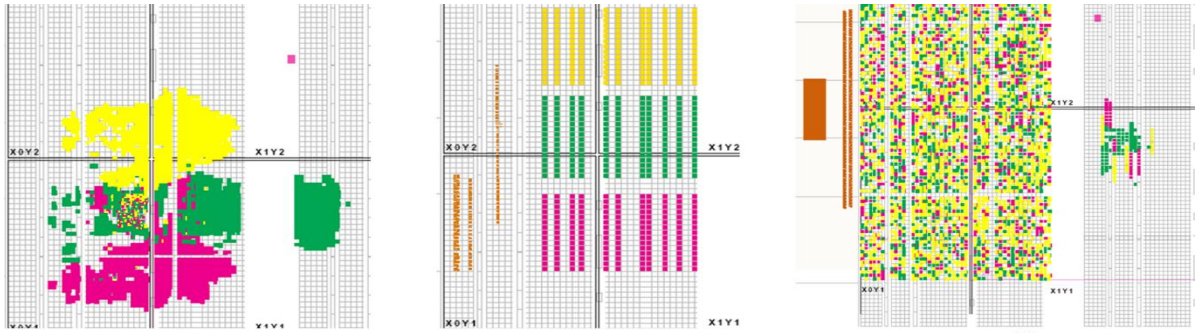


Fig. 1 Different layout solutions of TMR-hardened design: (a) original layout by Xilinx's Vivado (b) TMR-domain based isolated layout (c) High resource sharing layout. Each redundant module has been highlighted with a different color.

order to assure a layout solution that limits cross-domain failures and thus increases the reliability of the TMR system. The idea behind the relevance of the layout solution is that single CRAM cells can impact the behavior of multiple logic resources and affect resources related to more than one TMR domain.

In [11], a TMR-domain isolation-based design flow is presented which exploits the utilities provided by the Xilinx tool that require manual user intervention for placement customization while the routing is not considered.

The custom reliability-oriented place&route solution proposed in [12] shows that the efficiency of the TMR technique depends on the layout. However, it introduces a notable latency for providing the layout solution with respect to the commercial tool. In addition, it starts from a not-hardened design netlist and applies TMR to every function implemented. Our proposed approach is focusing on speeding up the layout customization process (even for non-TMR design) with an execution time that is limited to the actual time taken by the commercial tool to execute Tcl design-constraints commands.

The authors in [9] present the incremental placement and the striping techniques to resolve common-mode failure (CMF) caused by the routing of different clock domains on the same tile (CLB). The proposed incremental placement approach acts on swapping the SLICE contents with one from an adjacent tile, leading to isolation at the CLB level. Although, if the design is large or contains many independent modules, swapping may fail, takes a long time, or even stall by not finding the optimal solution. The striping solution consists of partitioning the FPGA area into columns and alternately assigning column resources to a different TMR domain. This solution has the advantage of being executed entirely within the commercial design flow. However, alternating resource allocation implies that connections between logical cells of same domain placed in different columns take place by sharing interconnection resources with the other domains that are placed in the columns in between. This leads to routing CMF in addition to routing congestion and unroutable design.

Our approach is a combination of these techniques, proposing a rectangular partitioning of the FPGA area that will contain logic from only one domain. In this way, we are able to resolve site CMF without the risk of routing congestion or routing CMF. Also, since each rectangle contains only one domain, there is no risk of routing CMFs at all. The developed tools are designed to have a fine granularity: the placement tool allows for modifying the layout acting on the single logic cell and primitive resource selection, and in the case of routing, a single net can be controlled and customized separately from the others.

2.1 Place & Route

The functional principle of the developed place and route is based on the reproduction of the Xilinx FPGA architecture. In the case of the placement tool, a Python environment has been built that emulates the FPGA structure by constructing an array of SLICEs having the same dimensions as the target device. The SLICE, in Xilinx nomenclature, is the elementary region of the FPGA in which the primitive logic resources are contained which is defined by a pair of coordinates (x, y).

Taking advantage of the fact that Xilinx provides a Tcl command to manipulate the layout by changing the location of a single cell indicating the name and the new target position, the tool aims to automate this process while requiring minimal user intervention.

The tool proceeds in reverse order from the usual design flow. The post-implementation design, seen as a netlist of primitive resources already mapped to the target device, is extracted from the commercial tool. For each primitive resource used some properties are extracted by filtering out those of interest which as the SLICE to which it belongs and the parent module name. The cell name and its parent module name are provided as a hierarchical path. This is exploited by the tool to visualize and manipulate the layout with different granularity since the grouping of cells under a module is made through a string match that can be broken at different hierarchy levels. As a first step, the tool remaps the original design to its environment and performs a vulnerability analysis providing the current view of the layout and a list of SLICEs where

there's an overlap of cells belonging to different domains. Redundant modules in the hierarchy are automatically recognized by the tool, which proceeds to apply an isolation policy per domain. The tool determines the number of resources required per module and organizes them in a rectangular area. Isolation is achieved leaving at least 2 rows of *empty* SLICEs between different domains. Manipulation of non-redundant modules is left to the user who, by selecting the module from a list provided by the tool, must pass the extremes' coordinates of the confinement region. The new layout physical design constraints are delivered as a Tcl file to be sourced in the commercial tool design flow.

The routing tool was created to automate the nets customization process provided by the Xilinx tool. The process requires manual selection of each new path segment, and the choice of one programmable interconnection point (PIP) over another is left to the user, therefore it is not scalable to a high number of nets. However, by taking advantage of this utility, a sufficiently large number of PIPs were extracted and characterized in terms of directly connected PIPs together with their relative position. This was used as a starting point for developing a basic router capable of automatically customizing multiple nets.

The router is supported by a Python reconstruction of the Xilinx FPGA interconnect environment. Each Xilinx Switch Box is associated with a CLB and is uniquely identified by a pair of spatial coordinates (x, y). The Switch Box encloses a large number of PIPs. At present, only the PIPs we characterized were included in the Switch Box model adopted by the router.

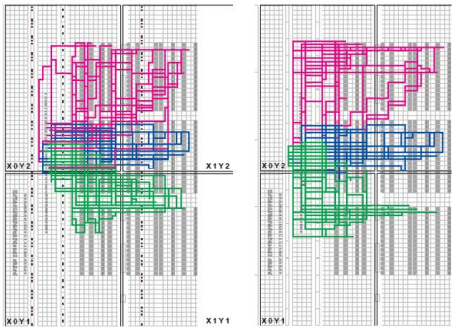
The router receives as input (i) the list of the nets to be customized, extracted from the post-place and route design within the Xilinx Vivado tool, and (ii) the constraints, in terms of path direction (i.e., horizontal or vertical) within a global routing channel (i.e., the semiplane horizontally delimited by x_{min} and x_{max}, and vertically delimited by y_{min} and y_{max}). For each net, our router selects a new interconnection segment that connects the source and destination cells while satisfying the constraints. Moreover, a net path can have multiple constraints, each tied to a separate global routing channel.

Our router generates in output each final routed path, in the form of Tcl commands, compatible with Xilinx Vivado. The power of this router lies in the fact that the customization process supports both multiple nets with the same constraints, and multiple nets with different constraints, for the same run.

2.2. Layout solutions

Thanks to the developed Place & Route tools, it is possible to implement two design hardening techniques. The first one is based on an isolation policy between redundant modules to minimize cross-domain failures. The second one is based on increasing the logic block area shared between the placed resources involved in the design. An example of the implementation solutions is illustrated in Fig. 1. Considering the original layout solution provided by Xilinx Vivado toolchain, shown in Fig. 1.a, the layout in Fig. 1.b implements an isolation policy between redundant modules, while Fig.1.c is the implementation of the high resource sharing solution. The imposed placement guidelines, applied to the isolated layout, organize resources so that each redundant core is contained within rectangular areas that have equivalent dimensions and have at least two logic block rows of unused resources between neighborhood regions. The TMR-majority voter has been placed in front of the central replica.

In particular, having placed the voter module in front of the central replica in the isolated layout and having not, initially, imposed any constraint on routing, led to a situation of sharing interconnection resources between domains. In Fig. 2.a are highlighted the output nets of the three redundant modules, flowing through the voting stage, as provided by the Xilinx tool. The green nets belonging to the bottom replica, in order to reach the voter, are routed upwards, invading the region where the central replica has been placed (blue nets). Similarly, the pink output nets of the top replica are routed downwards, again sharing the resources associated with the central replica. The developed router has been used to apply isolation at the net levels through the application of an L-routing path to the green and pink nets. The imposed constraints force the net to be routed only horizontally when the path is arranged inside the region of the parent redundant module plus an additional displacement to avoid congestion just out the module, and then only vertically until reaching the voter.



(a) (b)

Fig. 2. TMR-domain-based Isolation layout (a) before and (b) after net isolation. Colored nets represent the three top-level TMR hierarchical module output nets flowing through the voting stage.

3. Experimental results

The developed layout solutions have been evaluated with fault injection campaigns as well as proton irradiation tests. The hardware platform used to implement and test the design is the Xilinx Zynq-7020 AP-SoC which embeds an Artix-7 family FPGA [13] and dual-core ARM processors used to stimulate the DUT, collect the results and act as a golden reference during the tests. The benchmark circuit chosen is an advanced ALU working in Single Instruction Multiple Data mode extracted from a hardware implementation of the RISC-V ISA [14].
The block scheme of the implemented benchmark application is shown in Fig. 3, while the details of the benchmark circuits are reported in Table I.

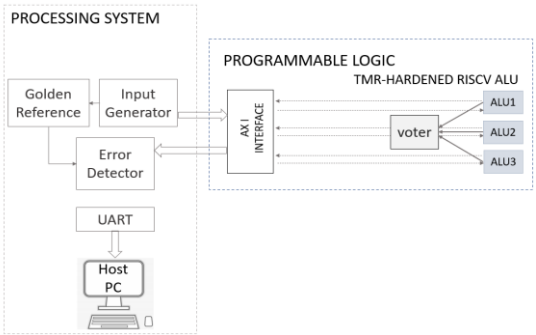


Fig. 3. Block scheme of the benchmark application.

As can be seen in Table I, the TMR technique is degrading the performance of the circuit since the addition of a voting stage increases the critical path. This is further accentuated by the usage of an AXI interface for data exchange between the ARM core present in the SoC and the TMR circuit implemented on the FPGA in order to have higher control of the behavior of the DUT during the test. Certainly adding a pipeline stage between the output of each module and the voting stage may be a way to optimize performance. Moreover, adopting fine-grained pipeline approach by inserting pipe registers within the individual redundant modules can also bring benefits. The adoption of the isolation policy for the TMR-domain, which involves leaving unused resources between the modules in question, further increases the critical path delay and thus there is a reduction in operating frequency and a consequent reduction in dynamic power contribution. The number of resources used, however, has a slight increase, probably due to the Synthesis guidelines adopted where buffers were required to cover the distances between modules.

Finally, the layout called High Resource Sharing is affected by a further decrease in frequency and increase in resources used due to the inherent nature of the layout, which for each redundant module, sees cells randomly distributed within a target region.

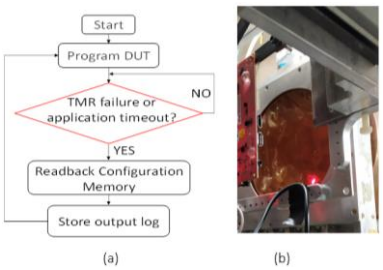


Fig. 4. The radiation test flow (a) and experiment set up (b).

Therefore, nets of cells belonging to the same computational block must cover the large distances hence a further increase in critical path delay and required resources.

Table I. The characteristics of the benchmark circuit implementations

TMR Designs	Performance [MHz]	Power [W]	Resource [%]
Original	50	1.397	13.18
Domain isolated	20	1.395	15.25
Resource sharing	15	1.394	23.10

3.1 Fault Injection experiment

Fault injection is a widely used technique to emulate radiation-induced effects. It consists of forcing a flip in one or more bits of the device CRAM. In this research work, the injection is performed by injecting a fault into the bitstream before it is loaded into memory. The bitstream was corrupted by employing the PyXEL [15] framework. Each injection campaign consists of generating 10k corrupted bitstreams, each one with a single inserted bit-flip. Table II shows the results of the injection campaigns conducted on the three layouts. The bare-metal testing routine executed by the processor core triggers an error every time there's a mismatch between the output provided by the DUT and the one produced by itself on the same input data. The collected data reveals that in the case of a single bitflip in the entire CRAM, the error rate is pretty low due to the fact that the design uses less than 25% of the overall FPGA resources.

Table II. Detected TMR failures over 10k 1-bit fault injections on Different layout configurations.

TMR Designs	TMR Failure Rate [#]	TMR Failure Rate [%]
Original	37	0,37
Domain isolated	24	0,24
Resource sharing	56	0,56

3.2 Proton irradiation results

We performed proton radiation experiments on all the implemented benchmark circuits, at the PSI facility in Villigen, Switzerland in order to measure the failure rate in really harsh environmental conditions.

As shown in Fig. 4.b the device was positioned perpendicular to the beam and each implemented

layout was tested considering different particle energies following the test flow reported in Fig. 4.a. The DUT was subjected to a range of energies from 16 to 200 MeV in order to calculate the SEU Cross-Section per Configuration Memory Bit (CRAM) which is reported in Fig. 5. For each energy, the cross-section points were calculated as the total sum of the SEUs that occurred in the CRAM divided by the total number of particles (computed using the flux provided by the facility) normalized with respect to the total unmasked bits of the entire CRAM. The curve saturates after 100 MeV which means that from that energy on, the number of SEUs accumulated does not depend on the energies carried by the particles.

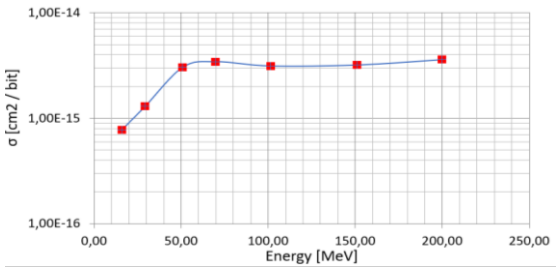


Fig. 5. SEU cross-section per unmasked CRAM bits for the XC7Z020 FPGA.

Due to the limited time available, not all layout solutions were tested for all beam energies. Therefore, the proposed comparison considers only those values of energies where data are available for all three layout proposals. In order to ensure a consistent comparison, the failure rate of each solution was normalized with respect to the beam fluence. Please notice that the results do not take into account the SEUs accumulated in the layout but only the number of failures detected with respect to the number of particles hitting the device during the effective execution time of the test routine. For this reason, the trend does not reflect that of the CRAM cross-section.

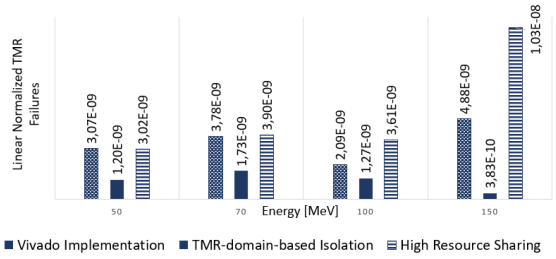


Fig. 6. Measured TMR Failure Rate with proton irradiation beam performed at PSI.

The obtained results are presented in Fig. 6, where the vertical axis shows the linear normalized TMR failure rate for the three implemented circuits considering 50, 70, 100, and 150 MeV proton energies. The achieved results show two relevant aspects. The former is that the TMR-domain-based isolation solution where logic and routing resources are isolated according to our developed approach outperforms the commercial TMR Xilinx Vivado implementation. The latter is that the Vivado and the High Resource Sharing designs have a similar failure rate for the range of energies below 100 MeV.

We believe that this is due to the common-mode errors happening in resources not directly hardened by the TMR solution such as majority voter's outputs. On the other hand, as the energy increases, there's a significant increase in the failure rate for the High Resource Sharing circuit that is doubling the Vivado design. The main reason is that when a particle's energy increases, particle strikes are affecting more CRAM cells, leading to the phenomenon of Multiple Bit Upsets (MBU), where several neighborhood CRAM bits are corrupted simultaneously. The corrupted data, in the High-Resource-Sharing case, is configuring resources associated with more than one TMR replica so leading to cross-domain faults and to the overall failure of the TMR system.

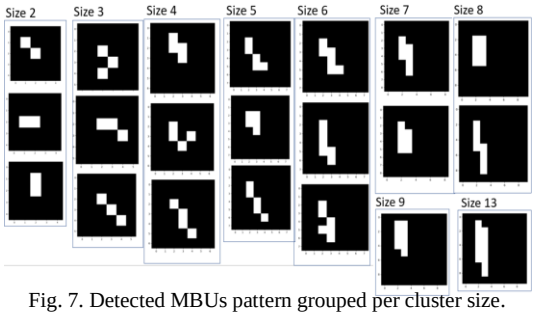


Fig. 7. Detected MBUs pattern grouped per cluster size.

Evidence for the presence of MBUs was derived by comparing readbacks of the CRAM, acquired at periodic 3-second intervals during the radiation test. The readbacks associated with the same beam run and test run are characterized by an increase in the total number of SEUs, between one readback and the next. Among all the detected SEUs in the readbacks, those whose Euclidean distance is less or equal to $\sqrt{2}$ are classified as MBUs [16]. Clusters of MBUs of different sizes have been identified and an increase in the size of the cluster related to the increase in particle energy has been noticed. However, for all the energies tested, a huge number of small-size clusters have been detected. In particular, the clusters of size 2 were found to be the most frequent and with greater multiplicity. In fact, considering the individual design data, about 2,000 MBUs distributed over 900 readbacks were detected. Of these, about 1,300 MBUs of size 2 were identified, followed by those of sizes 3 and 4, with an average presence of between 250 and 300, scattered over all readbacks in the

design. Fig. 7 shows some of the patterns found, grouped by cluster size.

3.3 *Fault Injection with accumulation*

The results obtained from the radiation test were then verified through additional fault injection campaigns with accumulation in order to reproduce the same behaviour of the DUT under the beam. In this injection mode, random injections were continued one bit at a time until a TMR failure was observed. The comparison of the three layout solutions is done in terms of the number of SEUs accumulated up to the time of the occurrence of a TMR failure. In this perspective, the higher the number of SEUs required for the design to fail, the more robust the solution. Fig. 8 shows the injection results of over 1,000 injections per layout solution. The results demonstrate that most of the failures occur for a number of SEUs smaller than 40, independently of the layout solution. In particular, about 60% of the events are concentrated under the maximum threshold of 10 upsets tolerated.

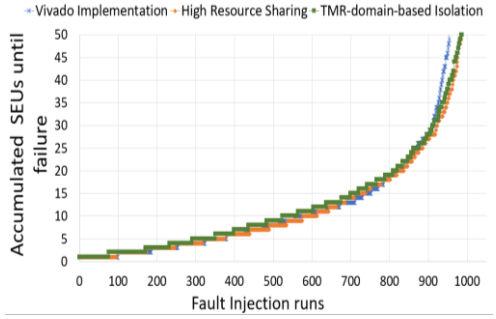


Fig.8 Accumulated SEU failure-rate for the three layout solutions with respect to 1,000 fault injection campaigns with accumulation.

Two aspects are evident from the collected data. On one hand, the design with the isolation policy has higher tolerability than the other two solutions, presenting for each range of accumulated SEUs a detachment from the other two. On the other hand, what is inferred is that the failure rate for low values of SEUs is significant. This certainly confirms one of the most critical aspects of the TMR technique, which is the creation of a single-point-of-failure in the design due to the voter module and drives the need to study and develop techniques that take this into account and mitigate its effects.

4. Conclusions and future works

This work presents a new design methodology for SRAM-based FPGAs on custom layout isolation capable to reduce the circuit failure rate and outperforming commercial solutions. Fault injections and proton irradiation tests were conducted on different P&R solutions of a TMR-hardened design to show how reliability changes as a function of the layout adopted. The proton test result proved that the effectiveness of a hardening technique can be highly affected by the layout choice and that, by adopting policies of isolation by domain, it is possible to achieve a reduction of the failure rate of more than one order of magnitude with respect to commercial approaches. In future works, we are investigating the design of the different voting structures and the impact of MBU on the measured failure rate.

References

[1] C. De Sio, S. Azimi, L. Sterpone, B. Du,"Analyzing Radiation-Induced Transient Errors on SRAM-Based FPGAs by Propagation of Broadening Effect," in *IEEE Access*, vol. 7, pp. 140182-140189, 2019.

[2] S. Azimi, L. Sterpone, B. du, L. Boragno, "On the analysis of radiation-induced Single Event Transients on SRAM-based FPGAs", Microelectronics Reliability, Vol.88–90, pp. 936-940, 2018.

[3] C. De Sio, S. Azimi, L. Sterpone, "On the analysis of radiation-induced failures in the AXI interconnect module", Microelectronics Reliability, vol. 114, pp. 1-5, 2020.

[4] R. Yarzada et al., "A Brief Survey of Fault-Tolerant Techniques for Field Programmable Gate Arrays," IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0823-0828, 2022.

[5] S. Azimi and L. Sterpone, "Digital Design Techniques for Dependable High Performance Computing," 2020 IEEE International Test Conference (ITC), 2020, pp. 1-10.

[6] M. Berg et al., "Effectiveness of Internal Versus External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis," in IEEE Transactions on Nuclear Science, vol. 55, no. 4, pp. 2259-2266, Aug. 2008.

[7] L. Sterpone and L. Boragno, "Analysis of radiation-induced cross-domain errors in TMR architectures on SRAM-based FPGAs," IEEE 23rd International Symposium on On-Line Testing and Robust System Design, pp. 174-179, 2017.

[8] D. G. Mahmoud *et al*., "Fault secure FPGA-based TMR voter," 7th Mediterranean Conference on Embedded Computing (MECO), pp. 1-4, 2018.

[9] M. Cannon et al., "Improving the Effectiveness of TMR Designs on FPGAs with SEU-Aware Incremental Placement," IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 141-148, 2018.

[10] M. J. Cannon et al., "Improving the Reliability of TMR With Non triplicated I/O on SRAM FPGAs," IEEE Transactions on Nuclear Science, vol. 67, no. 1, pp. 312-320, Jan. 2020

[11] A. Portaluri, C. De Sio, S. Azimi, and L. Sterpone, "A New Domains-based Isolation Design Flow for Reconfigurable SoCs," 2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design (IOLTS), 2021, pp. 1-7.

[12] L. Sterpone and M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs," IEEE Trans. on Computers, vol. 55, no. 6, pp. 732-744, June 2006.

[13] Xilinx, "7 Series FPGAs Data Sheet: Overview", Xilinx Product Specification, September 8, 2020.

[14] P. D. Schiavone, et al., "The RI5CY: User Manual Revision 4.0", 2019.

[15] L. Bozzoli, et al., "PyXEL: An Integrated Environment for the Analysis of Fault Effects in SRAM-Based FPGA Routing," International Symposium on Rapid System Prototyping (RSP), pp. 70-75, 2018.

[16] B. Du et al., "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, July 2019.