On the evaluation of FPGA radiation benchmarks

Gaetan Bricas, Georgios Tsiligiannis, Antoine Touboul, Jérôme Boch, Maria Kastriotou, Carlo Cazzaniga, Christopher Frost, Luigi Dilillo, Lucas Matana Luza

Title: On the evaluation of FPGA radiation benchmarks

Author(s): G. Bricas, G. Tsiligiannis, A. Touboul, J. Boch, M. Kastriotou, C. Cazzaniga, C. D. Frost, L. Dilillo, and L. Matana Luza

DOI: 10.1016/j.microrel.2021.114276

Published: 11 October 2021

Document version: Post-print version (Final draft)

Please cite the original version:
G. Bricas et al., "On the evaluation of FPGA radiation benchmarks," Microelectronics Reliability, 2021, doi: 10.1016/j.microrel.2021.114276.

On the evaluation of FPGA radiation benchmarks

G. Bricas[a,b*], G. Tsiligiannis[a,b], A. Touboul[a,b], J. Boch[a,b], M. Kastriotou[c], C. Cazzaniga[c], C. D. Frost[c], L. Dilillo[a,d,e], L. Matana Luza[a,d]

[a] *University of Montpellier, 34095, Montpellier, France*
[b] *UMR-CNRS 5214, 34095, Montpellier, France*
[c] *ISIS Facility, STFC, Rutherford Appleton Laboratory, Didcot OX11 0QX, UK*
[d] *LIRMM UMR 5506, 34095, Montpellier, France*
[e] *Centre National de la Recherche Scientifique (CNRS), France*

Abstract

This paper presents a benchmarking methodology to analyse the failure mechanisms of FPGAs under radiation, using comparative results on the radiation sensitivity of parallel multipliers with different implementations. Atmospheric neutron beam test results of Artix7, Spartan7 and IGLOO2 FPGAs are presented and validated against fault injection campaigns.

1. Introduction

The reduction of satellite manufacturing costs and time to market induced by the New Space industry, coupled with the constant evolution of FPGA capacity, complexity and performance, has placed these components at the forefront of components for the industry and research community as a serious alternative against traditional radiation hardened ASICs. In addition, the increased use of FPGAs in areas with a high need for reliability against radiation such as nuclear power plants, ground transportation or avionics, has brought them to the spotlight of the radiation effects and reliability community.

What makes FPGAs so attractive also imposes specific considerations regarding their behaviour under radiation. Indeed, unlike ASICs and one-time programmable FPGAs, the configuration of reprogrammable FPGAs is stored in memory elements that are affected by different radiation effects such as Single Event Effects (SEEs) [1] or Total Ionizing Dose (TID) effects [2]. The intrinsic sensitivity of the FPGA resources, the architecture of the implemented circuit and the protection mechanisms used (e.g., memory scrubbing) have an intricate influence on the reliability of the system. This requires the development of new methodologies for tests, qualification and comparison of these components.

As access to radiation testing facilities is particularly expensive, such tests should allow to jointly meet several purposes: gathering a maximum of information on the sensitivity of the component, evaluating the predominant failure mechanisms, effectively comparing different components, providing meaningful data to assess the reliability of a final system while providing for designers, a guideline for the development of their system and the application of mitigation strategies.

Several approaches have been proposed in the past. The first methodology is based on the independent evaluation of the sensitivity of each of the component resources. In [3], the cross section of the configuration memory is evaluated by static tests by performing a readback of the configuration memory after an exposure to a radiation beam. In [4], monomorphic test structures such as shift registers are used to evaluate the sensitivity of flip-flops and LUTs. However, extrapolating these results to identify the reliability of a complete system is rather complex since the types of interactions between resources are highly dependent on the topology of the implemented circuit. Another approach consists in testing the FPGA under radiation by directly implementing the final design as described in [5]. Designs implemented on FPGAs are generally very complex. Logical error masking, especially for circuits where mitigation strategies have already been applied, implies a low error rate and therefore a low statistical significance for reliability assessment. Also, since the state space is generally large, not all circuit states can be sufficiently exposed during a single radiation test to be representative enough of the failures that may

* gaetan.bricas@ies.univ-montp2.fr
Tel: +33 6 80 41 02 87

occur during flight operation. Finally, the low visibility provided by these tests does not allow to identify properly the most sensitive areas of the design. The test must therefore be renewed after each modification. In [6], the ITC'99 benchmark is proposed as a universal reference for radiation testing to allow researchers and designers to compare on a common basis their results on different components and their mitigation strategies. However, these benchmarks are not perfectly adapted to radiation testing as it is limited in the diversity of used resources and circuit topologies. In [7] we have presented a set of benchmarking structures that combine the benefits of the aforementioned categories in order to provide with meaningful testing vehicles that are also in the core of many systems, able to reveal a large variety of radiation effects and the dependence between the internal FPGA elements.

In this paper, we present new insights on the effectiveness of the radiation benchmarking structures presented in [7] and improvements to expand their capabilities. The capabilities of the benchmark are evaluated by atmospheric neutron beam testing of Xilinx's Spartan7, Artix7 (SRAM 28nm) and a Microsemi's IGLOO2 (Flash 65nm) FPGAs. Radiation test results are presented and analysed, along with a comparison with a fault injection campaign.

2. FPGA benchmarking methodology

2.1. Baseline concept

To meet all the requirements cited above, the benchmark must respect the following criteria:

- Scalability and portability across FPGAs of different technologies and manufacturers.
- Low error masking rate and reduced state space to increase failure visibility.
- A test setup that allows the identification of the origin of errors.
- Using different implementations or mitigation strategies to identify reliability-oriented design rules.
- Heterogeneity of the architectures implemented to more broadly represent the elements that are typically implemented on FPGAs.

The benchmark we propose in this paper is based on parallel multipliers. Achieving good implementation efficiency and performance is a critical challenge to hardware designers. As described in [8], this operator can be implemented in various ways, leading to different compromises in terms of resource utilization, power consumption and performance, and as a consequence impact the radiation performance of the system. Moreover, the same compromises are met in larger systems and thus, lessons learned from the benchmarking structures could be scaled to larger designs. In addition to the above-mentioned criteria, the multiplier has been chosen as a test structure for the following reasons:

- Multipliers are widely used functions in all computationally intensive applications (digital signal and image processing, cryptography, artificial intelligence, etc.). The utilization of logic functions close to those actually implemented improves the representativeness and facilitates the interpretation and reuse of the results.
- Multiplication functions can be implemented using different resources (DSP Blocks, Carry Logic, LUT, Flip Flop) and different arrangements between those basic elements. This allows, through the same logic function, to mobilize the most abundant resources of the FPGA and to use a great diversity in the circuit parameters (number of logic stages between the flip-flops, fan-in and fan-out of each element, LUT utilization, etc.)
- Multiplication is an operator that offers a great visibility of errors. With a judicious choice of input test vectors, the very nature of the operation allows to reveal most of the errors generated in the circuitry by monitoring only the output of the operator.

By using different implementations of the multiplier, a dual objective can thus be achieved: the comparison of the susceptibility of the operator on its different implementations can serve directly as a guideline for the designer while efficiently evaluates the sensitivity of the most abundant logic resources.

2.2. Multiplier implementation

A binary multiplier computes a set of partial products and then sums the shifted partial products together. Partial product computation being done by AND-gating the first operand with one bit of the second operand, the partial product reduction process contributes the most to the delay power and area of the multiplier. The differences in the implementation of this operator lie in the reduction of partial products. However, most of these methods, originally invented for ASICs, do not necessarily adapt to the particular architecture of FPGAs. Indeed, most FPGAs integrate in each configurable logic block, circuits dedicated to the fast propagation of carries allowing the generation of fast adders and counters. The methods must therefore adapt to this architecture as well as to the limitation imposed by the size of the LUTs and the routing capabilities of the component. Other methods have since been developed to improve the performance of multipliers implemented on FPGAs. In this paper we will mainly use four different implementations.

2.2.1. Benchmark 1: Carry save multiplier (CSM)

A CSM uses half adders (HA) and full adders (FA). The carries going out from each HA or FA are fed to the next partial product. Full adders with associated AND gate for partial product computation can be integrated into one LUT. However, this structure appears to be inefficiently implemented in FPGAs as the forward propagation of carries cannot take advantage of the horizontal propagation of carries imposed by Carry logic blocs.

2.2.2. Benchmark 2: IP Core - speed optimized (SO)

This architecture is extracted from Xilinx IP cores. It makes efficient use of the FPGA architecture by integrating within a single stage of LUTs and associated carry chain, the calculation and reduction of two consecutive partial products. The resulting partial sums are then added two by two until the final result is obtained as shown in Fig.1.

2.2.3. Benchmark 3: IP Core - area optimized (AO)

With this architecture, also extracted from Xilinx IP cores, the bits of the partial products are first reduced with a stage of optimally arranged LUTs as depicted in Fig.2. One part of LUTs (Type A) only compute the SUM bits of partial product over 1, 2 or 3 bits of the same weight ignoring the CARRY bit. The other part of LUTs (type B) compute the SUM and CARRY bits of 1 or 2 bits of the same weight while integrating the CARRY bit from the sum of the lower weight bits sharing the same input bits. The vectors extracted diagonally from this compression stage are then added 3 by 3 using ternary adders taking advantage of the carry chain architecture until the final result is obtained. To be noted that this architecture requires a 6-input LUT based fabric. Therefore, it cannot be implemented on part of low and medium end FPGAs that only integrate 4-inputs LUT such as the IGLOO2 FPGA tested in this paper.

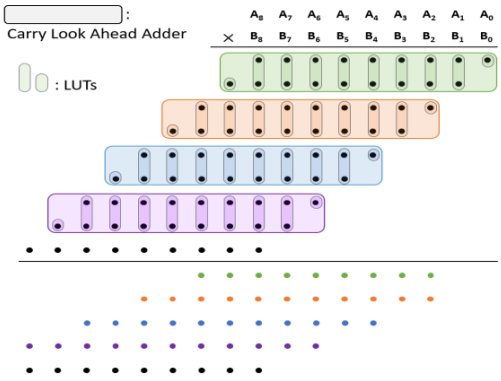

Fig. 1. Dot diagram example of **speed optimized** 9x9 multiplier partial product reduction

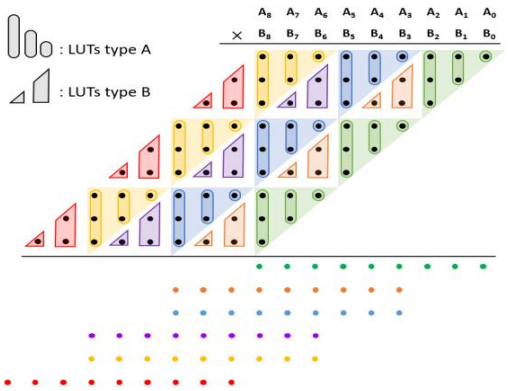

Fig. 2. Dot diagram example of **area optimized** 9x9 multiplier partial product reduction

2.2.4. Benchmark 4: DSP based multiplier

The last architecture uses the DSP blocks: hardwired block dedicated to arithmetic operations available in all modern FPGAs.

2.3. Experimental setup

These implementations have been tested with 16x16 multipliers using Finite Impulse Response (FIR) filters. This structure allows to test a large number of operators only by observing one output. The feedforward propagation of data and the absence of additional error masking suits well with the test's requirements. The filter output is continuously compared to a golden signal previously extracted from simulation and stored in a memory block protected by error correction codes. The FIR filter architecture thus allows to limit the proportion of resources allocated to internal test circuits, reducing the number of false errors detection due to checking circuit failures. The output of the comparator is monitored and sent to the host PC to record all the events.

The Soft Error Mitigation (SEM) IP provided by Xilinx [9] is integrated to the design to detect and correct SEUs in the configuration memory. The IP continuously reports the detected events to the host PC, which allows to calculate the effective bit cross-section of the configuration memory. The same IP has been used to perform the fault injection campaign on the same designs. The results of the fault injection campaign are compared with those of the irradiation campaign to confirm the consistency of the results and confirm the hypotheses made on the failure mechanisms. The Flash configuration cells of the IGLOO2 FPGAs are known to be immune to SEEs [10], and thus there is no need for memory scrubber. In order to validate the designs and concepts, radiation testing has been conducted at ChipIR [11,12,13] under a beam of atmospheric neutrons with a flux of 5.10^6.cm^{-2}.s^{-1}.

3. Experimental results

By monitoring the output of the comparators and the messages sent by the SEM IP on the Xilinx FPGAs we were able to determine the exact timing of the errors in the data stream. We have distinguished three types of error signatures:

- **Single errors**, originating from SEU in the Flip Flops or Single Event Transient (SET) in the combinatorial logic captured by one of the end point Flip Flops
- **Non-persistent** upset, long frames of errors due to SEU in the configuration memory that stop after being corrected by the SEM IP
- **Persistent** upsets, coming from SEU in the configuration memory uncorrected by the SEM IP (causes discussed below)

Based on the number of events recorded, the cross section for each type of error has been calculated as shown in Fig.3. The analysis of these results must be put into perspective with the resource utilization as shown in Tab.1. Common observations can be made on all targets. Multipliers from benchmark 1 (CSM) are far more sensitive on all targets and for all type of error signature. Benchmark 3 (*AO*) also appears to be more sensitive than benchmark 2 (SO) (Fig. 4) while it is using slightly less logical resources and slices.

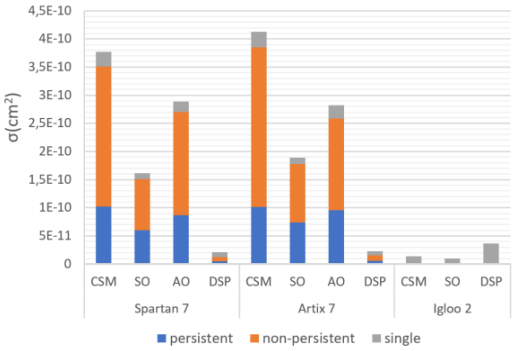

Fig. 3. Cross section for each type of error based on the events recorded during the test campaign

	Spartan7 and Artix7				
	Logical LUT	**Flip-Flop**	**CARRY chain**	**Shift-reg LUT**	**Logic block**
CSM	4734	1982	4	694	1400
SO	2607	3919	842	66	1061
AO	2355	4009	469	279	971

	Igloo2		
	Logical LUT	**Flip-Flop**	**1bit-CARRY**
CSM	6569	6297	0
SO	3183	4246	2669

Tab. 1. Resource utilization per FIR filter

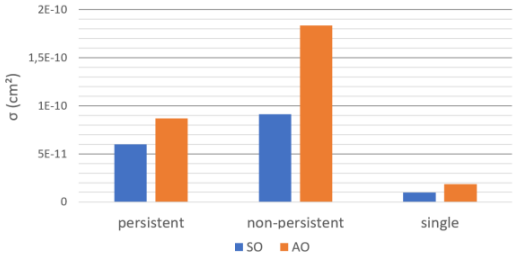

Fig. 4. Cross section comparison between benchmarking 2 and 3 sensitivity (Spartan7)

These results show the complexity of modelling the susceptibility of circuits implemented on SRAM-based FPGAs. As the functionality of the design itself can be modified (routing, configuration of LUTs and specialized blocks), the architecture of the circuit and the placement and routing parameters have a major impact on radiation sensitivity. This increased sensitivity with reduced number of used resources could be explained in this case by the replication of input data nets. Indeed, with the area-optimized version of the multiplier, some of the sum bits and carry bits of the partial products are computed by separated LUTs, the input signal bits are thus distributed to an increased number of LUTs increasing the number of used Programmable Interconnect Points (PIP). Furthermore, the implementation of ternary adders on Xilinx FPGAs cannot simply use the LUT/CarryLogic combination contained in a slice. Indeed, the output from the LUTs must be reused for the calculation of the higher order bit, forcing the use of extra-slice routing [14]. Since extra-slice routing resources are enabled by several PIPs, they add a susceptibility to CRAM bitflips that could counteract the reduction of used resources over binary adders.

On the other hand, the benchmark 4 (DSP) is far less sensitive on both SRAM devices while being more sensitive on the Flash device. This last result demonstrates that the reliability oriented designing guidelines must be adapted to each device, reinforcing the importance of a tailored benchmark for radiation qualification.

As for singular errors, for SRAM FPGAs, they are dominant for benchmark 4 (DSP) (40% of the total number of events) but are very minor for the fabric-based benchmarks (<7%). This relative domination can be explained by the fact that DSP blocks are dedicated blocks with limited flexibility and therefore require only a few configuration bits to define their functionality. This aspect also applies to the comparison between other filters: since CARRY LOGIC blocks are not configurable, their use to replace logic functions otherwise implemented on LUTs will reduce the susceptibility of the circuit to SEUs in the configuration memory. The number of single errors tends to show that the use of carry logic

blocks does not significantly increase the number of captured SETs.

From these results, we can already extract some guidelines for the implementation of arithmetic operators. For SRAM FPGAs, the flexibility of the resources used comes at the cost of an increase in the susceptibility to SEUs in the configuration memory which are predominant in this technology. The use of specialized blocks with a reduced flexibility is therefore recommended, DSP blocks are thus advised when available and the use of structures employing Carry Logic blocks will be favored over structures employing only LUTs.

On the contrary, for configuration immune FPGAs, irradiation tests are required to determine the relative sensitivity of the different resources. In the specific case of the IGLOO2 FPGA, fabric-based operators are less sensitive to radiations than DSP blocks but the penalty in power consumption may counterbalance this advantage.

On another note, the strong presence of persistent errors (~one-third of all failures) questions the correction capacity of the SEM IP under a high flux beam.

The analysis of the reports sent by the IP SEM provides an insight into these failure mechanisms and their prevalence. Based on these reports sent during the irradiation, it is possible to count the number of events detected in the configuration memory. For each event, the report indicates the position of the affected memory bit and whether the bitflip was corrected or not. Based on the number of detected bitflips, the cross section of the configuration memory for Spartan7 and Artix7 FPGAs can be calculated ($2.1 \cdot 10^{-15}$ cm²/bit and $2.2 \cdot 10^{-15}$ cm²/bit respectively) which is more than three times lower than the one measured at LANSCE ($7.0 \cdot 10^{-15}$ cm²/bit) according to [15]. This discrepancy can be explained by the difference between the neutron spectrum of both facilities as shown in [16].

By crossing the data extracted from SEM IP reports with the data from the comparator at the output of the filters, it clearly appears that each non-persistent error coincides with the detection and correction of a bitflip in the configuration memory. The delay between the occurrence of the bitflip and its correction can be calculated based on the number of clock cycles where the filter is reported as being faulty. An average detection time of 2.7ms has been measured which matches the one provided by the manufacturer (2.9ms) in [9].

Regarding persistent errors, the cross-referencing of data also reveals that each persistent error recorded is preceded by the failure of the SEM IP itself or by the detection of an uncorrectable error, forcing the mitigation system to leave its detection mode. These observations confirm the assumption made on the failure mechanisms involved in this experiment. According to the SEM IP documentation [9], when using the "*enhanced mode*" of the IP as we did for this experiment, the system is unable to correct the multiple non-adjacent errors in the same memory frame. The probability of two particles interacting with two bits of the same frame between two readbacks (4.6ms) being extremely low, the detected uncorrectable errors must originate from Multi Bit Upsets (MBU) or from particular configuration bits whose change of state would be considered as uncorrectable. This aspect is further studied through the fault injection campaign. From the number of uncorrectable errors and the number of failures of the SEM IP (1.6% of all event detected), we compute the effective cross section associated with the failure of the correction capacity σ_{SEM}= 2.8·10-10 cm². Based on this cross section and the one of a given filter, the proportion of persistent errors among the total number of configuration memory related errors can be estimated with Eq.1.

$$\frac{N_{persistent}}{N_{Total}} = \frac{\sigma_{SEM}}{\sigma_{SEM}+\sigma_{FIR}} \quad (1)$$

4. Fault Injection campaign

The SEM IP allows the emulation of SEUs in the configuration memory. Using its serial interface, bitflips can be injected anywhere in the configuration memory (except in Block RAM) by providing the address of the bit to be inverted. The manufacturer's EDA software provides the list of essential bits when generating the configuration file. The essential bits are, according to the manufacturer, the bits that potentially have an impact on the circuit. By restricting fault injections to the essential bits, we eliminate a major part of the bits that have no impact on the design. The fault injection campaign is performed on the designs used during the irradiation campaign, focusing on the Spartan7 FPGA. The architecture of the Spartan7 and Artix7 fabric being identical, the results of this campaign will remain valid for the Artix7 FPGA. For each design, we randomly inject approximately 150,000 errors among the essential bits according to the following procedure: each error is injected separately by forcing after each injection, the detection and correction of the error by the mitigation system. If an uncorrectable error or a failure of the SEM IP itself is detected, the FPGA is completely reconfigured. The accumulation of faults is thus prevented in order to test the criticality of each bit independently. For each injection, the output of the comparator is monitored to identify if the bit is critical (i.e. if its inversion causes a filter failure.). For each filter, the percentage of critical bits among those injected is measured, and the total number of critical bits is extrapolated from the total

number of essential bits. Using the cross-section per bit measured in part 3, the cross-section of each filter can be calculated by multiplying the number of critical bits to the cross-section per bit. These results are compared to those from the irradiation campaign in Figure 5.

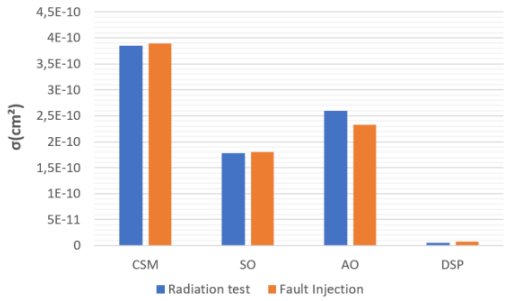

Fig. 5. Comparison of filters cross sections between the radiation tests (single error excluded) and the fault injection campaign (Spartan7)

As shown in this figure, the results from the two approaches are in very good agreement. This observation confirms the origin of the errors observed during the campaign. It also shows that the extraction of the effective cross section can be reused with a fault injection campaign to evaluate the susceptibility of another design without the need for further irradiation experiments. However, the fault injection approach does not assess the single errors due to SEUs in Flip-Flop or SETs in the combinatorial logic. This approach works particularly well in our case because the tested structures do not contain feedback loops or complex state space that could compromise the visibility of errors. With more complex architectures containing complex state machines, counters or microprocessors, this approach could turn out to be less efficient because the effect of the SEU-induced architectural modification might be different depending on the state of the system. It would then be necessary to establish a much more complex probabilistic model to estimate the susceptibility of the system. This would require that each state of the system and each combination of inputs be tested for each fault injected, which could significantly lengthen the duration of the fault injection campaign. To be noted that in this campaign, our system can inject and correct slightly more than one fault per second, a duration of about 36h per design was necessary to reach the 150,000 injected faults. This method is notably limited by the speed of the serial interfaces between the FPGA and the PC. By integrating the software part realized on the PC directly into the FPGA, we could greatly accelerate the injection rate. A particular attention must be brought to the number of injected faults necessary to reach a satisfactory precision. For circuits of low susceptibility, the number of injections must be more important. In our case, it was verified that the number of injections was sufficient to reach +/-5% of the real proportion of critical bits based on an exhaustive fault injection campaign (100% of the essential bits injected) on one of the least susceptible filters.

Concerning the uncorrectable errors, a number of bits were detected as uncorrectable after their injection or caused the failure of the SEM IP. As realized in part 3, the cross section corresponding to all these bits has been calculated: $\sigma_{SEM,FI} = 5 \cdot 10^{-11} cm^2$. This cross section represents only 17% of the cross section evaluated during the irradiation campaign. This result suggests that a large part of the uncorrectable errors detected during the irradiation campaign are due to MBUs or configuration bits that are not accessible for error injection by the SEM IP (internal device control registers and state elements). A fault injection was performed on the non-essential bits to verify that none of these bits could generate uncorrectable errors, which was confirmed.

5. Conclusion

The benchmarks proposed in this paper, based on diverse implementations of multipliers, has shown its ability to address the different requirements of radiation testing of FPGAs. The diversity of benchmarks regarding resource utilization and circuit topology allows to evaluate the sensitivity of the basic elements composing the FPGA fabric while providing useful guidelines for the reliability of computationally intensive designs. The test results provided a clear characterization of failure and recovery mechanisms on SRAM FPGAs equipped with an internal scrubbing system. The use of this scrubbing system also offered an efficient way to estimate the cross-section of the configuration memory bits that can be used without interruption during the operation of the circuit, unlike other methods based on external readback of the configuration memory. The extracted cross-section can be reused jointly with a fault injection campaign to estimate the radiation sensitivity of other designs without further radiation tests. The fault injection campaign based on the use of the SEM IP also allowed to confirm the experimental results and to explain more precisely the origin of observed failure mechanisms. Beyond this feature, the SEM IP appears to be a convenient and very efficient way to avoid the accumulation of errors in the configuration memory as 98.4% of the errors are corrected. Nevertheless, uncorrectable errors can compromise this mitigation system. Specific actions will have to be taken to manage this type of events (reconfiguration, external scrubbing, etc.). The comparative results of the different benchmarking structures and the different components have

highlighted some FPGA specific design rules encouraging the use of the least flexible logic blocks (DSP and Carry logic) for configuration sensitive FPGAs and to pay attention to the false attractiveness of using ternary adders in partial product reduction trees. Finally, the benchmark has shown that for Flash based FPGAs, even if the sensitivity to singular errors is of the same order of magnitude as SRAM based FPGAs, the immunity of the configuration memory to SEUs makes them much more tolerant to SEEs except for DSP based circuits.

References

[1] P. S. Ostler and al., 'SRAM FPGA Reliability Analysis for Harsh Radiation Environments', IEEE Transactions on Nuclear Science, vol. 56, no. 6, pp. 3519–3526, Dec. 2009, doi: 10.1109/TNS.2009.2033381.

[2] J. J. Wang and al., 'Total ionizing dose effects on flash-based field programmable gate array', IEEE Trans. Nucl. Sci., vol. 51, no. 6, pp. 3759–3766, Dec. 2004, doi: 10.1109/TNS.2004.839255.

[3] H. M. Quinn and al., 'A Test Methodology for Determining Space Readiness of Xilinx SRAM-Based FPGA Devices and Designs', IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 10, pp. 3380–3395, Oct. 2009, doi: 10.1109/TIM.2009.2025469.

[4] J.-J. Wang, D. Dsilva, N. Rezzak, S. Varela, and S. Cui, 'Single Event Effects Testing on the SERDES, Fabric Flip-Flops and PLL in a Radiation-Hardened Flash-Based FPGA-RT4G150', in 2016 IEEE Radiation Effects Data Workshop (REDW), Jul. 2016, pp. 1–6, doi: 10.1109/NSREC.2016.7891741.

[5] E. Gousiou, G. F. Penacoba, J. C. Cubillos, and E. Gousiou, 'Radiation tests on the complete system of the instrumentation of the LHC cryogenics at the CERN Neutrinos to Gran Sasso (CNGS) test facility.', Topical Workshop on Electronics for Particle Physics, Paris, 21 -25 Sep 2009, p. 4.

[6] H. Quinn et al., 'Using Benchmarks for Radiation Testing of Microprocessors and FPGAs', IEEE Trans. Nucl. Sci., vol. 62, no. 6, pp. 2547–2554, Dec. 2015, doi: 10.1109/TNS.2015.2498313.

[7] G. Bricas et al., 'Novel FPGA Radiation Benchmarking Structure', presented at RADECS, Vannes, France, Sep. 19–20, 2020

[8] M. Kumm, S. Abbas, and P. Zipf, 'An Efficient Softcore Multiplier Architecture for Xilinx FPGAs', in 2015 IEEE 22nd Symposium on Computer Arithmetic, Lyon, France, Jun. 2015, pp. 18–25, doi: 10.1109/ARITH.2015.17.

[9] Xilinx document,'*Soft Error Mitigation Controller v4.1*'. Accessed: Feb. 15, 2021. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/sem/v4_1/pg036_sem.pdf

[10] Microsemi, '*SmartFusion2 and IGLOO2 Neutron Single Event Effects (SEE)*'. Accessed: Feb. 15, 2021, [Online]. Available: https: //www.microsemi.com/document-portal/doc_download /135249-tr0020-smartfusion2-and-igloo2-neutron-single-event-effects-see-test-report

[11] C. Cazzaiga, M. Bagatin, S. Gerardin, A. Costantino, and C. D. Frost, 'First tests of a new facility for device-level, board-level and system-level neutron irradiation of microelectronics', IEEE Transactions on Emerging Topics in Computing, p. 1 1, 2018, doi: 10.1109/TETC.2018.2879027.

[12] Dr Georgios Tsiligiannis et al; (2020): SEE response evaluation of robotic systems for the nuclear decommissionning, STFC ISIS Neutron and Muon Source, https://doi.org/10.5286/ISIS.E.RB2010438

[13] Dr Luigi Dilillo et al; (2020): Evaluation of a fault-tolerant RISC-V, STFC ISIS Neutron and Muon Source, https://doi.org/10.5286/ISIS.E.RB2010053

[14] J. M. Simkins and B. D. Philofsky, 'Structures and methods for implementing ternary adders/substractors in programmable logic devices,' U.S. Patent 7 274 211, Sep. 25, 2007.

[15] Xilinx document, '*Device Reliability Report*', Accessed: Feb. 15, 2021, [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug116.pdf

[16] C. Cazzaniga and C. D. Frost, 'Progress of the scientific commissioning of a fast neutron beamline for chip irradiation,' J. Phys. Conf. Ser., vol. 1021, no.1, p. 012037, May 2018.