

On the design of tunable fault tolerant circuits on SRAM-based FPGAs for safety critical applications

L. Sterpone¹, M. Aguirre², J. Tombs², H. Guzmán-Miranda²

¹Politecnico di Torino

Dipartimento di Automatica e Informatica
Torino, Italy

{luca.sterpone@polito.it}

²Universidad de Sevilla

Departamento de Ingeniería Electrónica
Sevilla, Spain

{aguirre@gtex10.us.es, jon@gtex10.us.es, hipolito@gtex10.us.es }

ABSTRACT

Mission-critical applications such as space or avionics increasingly demand high fault tolerance capabilities of their electronic systems. Among the fault tolerance characteristics, the performance and costs of an electronic system remain the leader factors in the space and avionics market. In particular, when considering SRAM-based FPGAs, specific hardening techniques generally based on Triple Modular Redundancy need to be adopted in order to guarantee the desired fault tolerance degree. While effectively increasing the fault tolerance capability, these techniques introduce an important performance degradation and a dramatic area overhead, that results in higher design costs. In this paper, we propose an innovative design flow that allow the implementation of fault tolerance circuits in SRAM-based FPGA devices with different fault tolerance capability degrees. We introduce a new metric that allows a designer to precisely estimate and set the desired fault tolerance capabilities. Experimental analysis performed on a realistic industrial-type case study demonstrates the efficiency of our methodology.

1. Introduction

The electronic systems deployed in mission critical environments are increasingly demanding components, devices and systems that offer high performance characteristics combined with enhanced dependability. In particular, considering the electronic systems for space and avionics applications, a silicon device needs to satisfy strict dependability rules before they can be authorized for the usage in these fields.

The avionic, and especially the cosmic space environments, are prone to radiation particles that may hit the surface of the silicon electronic devices leading to dangerous effects in their desired behaviour [1].

Among the available technologies, Field Programmable Gate Arrays (FPGAs) based on Static-RAM configuration memory provide high performance, high resource count, and low cost, while allowing in-system reconfiguration, which are mandatory requirements for implementing successfully re-configurable systems. SRAM-based FPGA devices have become increasingly interesting for various kinds of applications ranging from the digital signal processing domain to the mission or safety-critical field. This has been mainly due to the

drastic decreasing of the cost and the development time needed to design and map complex applications.

When considering electronic systems deployed in aircraft and satellites, which are all safety critical domains, designers are investigating the benefits of adopting re-configurable architectures such as SRAM-based FPGAs. Nevertheless the advantages these devices offer, the dependability issues related to the Static-RAM cells technology demand the adoption of specific design hardening techniques that enable their usage in safety critical missions such as the space and avionics applications [2].

In this paper we propose a new design flow that allows designers to effectively implement fault tolerant circuits on SRAM-based FPGAs, tuning their protection capabilities against Single Event Effects (SEE) affecting the configuration memory. The proposed design flow is based on the combination of three tools adapted to the related problem: an analyzer tool (STAR-Discovery), a fault-injection platform (FT-UNSHADES-C) and an innovative development of reliability-oriented place and route algorithm (RoRA-LX). The analyzer tool, called *STAR-Discovery* performs the static analysis of the circuit FPGA implementation and for each module i of the implemented circuits, it generates a list Σ_i of SEUs that may produce failures and a list Π_i of configuration memory bits that effectively generate wrong system outputs. The analysis *STAR-Discovery* performs is applicable to any design implemented on a SRAM-based FPGA and is workload independent since they are based only on a circuit's model description. The fault-injection platform *FT-UNSHADES-C* has been developed according to inject SEUs within the FPGA's configuration memory cells. In details, it injects all the faults identified by *STAR-Discovery* and evaluates the dynamic evolution of the implemented circuit on the basis of the applied input stimuli. Thus, it refines the faults subset generating a list of faults that provoke failures when a given workload is applied to the system's inputs. Finally, a module i is harden against the SEUs effect by the *RoRA-LX* on the basis of a sensitivity metric computed according to the static and dynamic evaluations of the configuration memory bits. The *RoRA-LX* algorithm hardens the module i tuning the fault tolerance degree of the complete system.

The major contributions provided by the proposed design flow are the increase of performance in term of

running frequencies of the implemented circuit, and the reduction of the area overhead needed. We evaluated experimentally our methodology on a CORDIC Core microprocessor core representing a realistic industrial application.

The paper is organized as follows. Section 2 presents the background and an overview of already developed works concerning the design of tunable fault tolerant circuits when implemented on FPGA-based system. Section 3 presents an overview of the proposed design flow. Section 4 describes the developed sensitivity metric. Section 5 and Section 6 describe the details of the tool and the platform developed, while its experimental evaluation is provided in Section 7. Finally, conclusions and future works are drawn in Section 8.

2. Background and related works

The architecture of SRAM-based FPGAs is composed of a high-density array of SRAM memory cells, commonly called configuration memory, which content defines how the available Configurable Logic Blocks (CLBs) resources, memory modules and routing resources (wire segments and programmable switches) are programmed to implement any user application [3]. The manufacturing technology used to fabricate the SRAM memory cells is sensitive to radiations that may cause Single Event Upsets (SEUs) [4], which may correspond to the modification of the function performed by a Look-Up Table (LUT), a flip-flop or a routing segment.

The SEU upset rate is related to the kind of radiation environment the SRAM-based FPGA will be used in. At the ground level, neutrons and alpha particles are the most usual causes of SEUs, while in the space environment, there are protons and heavy ions. Investigation analysis performed using a Xilinx Virtex 1000 device (containing more than six-million bits), show a low neutron incidence at the ground level (3.6 SEUs in 1 million hours) [5], while during specific in-flight experiments on an average orbit, has evaluated a SEU upset rate ranging from 0.13 to 4.2 SEUs per hour [6]. Since SEUs may critically modify the application a SRAM-based FPGA implements, the adoption of protection mechanisms are mandatory. Although an existing solution is available using radiation-hardened (*rad-hard*) technologies making memory cells insensitive to latch-up and total dose effects, but devices remain still sensitive to the transient effect of SEUs. The cost of devices using such kind of technology is prohibitive for most applications. In the past years, several fault-tolerant techniques based on redundancy have been proposed in order to mitigate the effects of SEUs in not rad-hard SRAM-based FPGAs. Some example of these techniques, are Triple Modular Redundancy (TMR) [7] and Duplication With Comparison with Concurrent Error Detection (DWC-CED) [8]. However, they do not guarantee complete protection of faults inside SRAM-based FPGAs. For solving this problem, a specific

reliability-oriented place and route algorithm (RoRA) has been developed [10]. RoRA is an attractive solution for SRAM-based FPGAs because it introduces full hardware redundancy in the user's combinational and sequential logic, the routing signals and the I/O pads and it guarantees complete protection against SEUs affecting the FPGA's configuration memory [11]. However, when RoRA is applied to the entire design it introduces several drawbacks in terms of drastic increment of the area overhead and reduction of the user circuit running frequency.

Several approaches investigated the hardening capabilities of tunable fault tolerance techniques. In [12] an approach is proposed that uses selective triple modular redundancy (S-TMR) which extends the basic TMR technique by identify SEU sensitive gates given a circuit and then introducing TMR selectively at these gates. This approach is a valid alternative to the TMR, since the area of the S-TMR circuit is, in the worst case, equal to the area used by the full TMR circuit. Although, it is able to protect the logic component, this approach does not present any protection technique against multiple errors that can affect the FPGA's routing resource. Besides, this approach is applied to the gate level of the circuit without considering the FPGA configuration failure model. A different approach that considers the FPGA configuration failure model is presented in [13], where a BLTmr tool has been created to perform automatically selective TMR with a degree defined by the user. However, this approach presents the major drawback that it is only possible to apply four levels of TMR structures, without allowing the user to select a desired circuit module. Furthermore, these approaches present a common drawback that is the impossibility to investigate the dynamic evolution of the considered circuits, thus without considering the behaviour of the circuit's running application.

The methodology presented in this paper is completely innovative if compared with previously developed approaches since it is able to analyze and harden a circuit mapped on SRAM-based FPGA tuning its level of fault tolerance, considering both the static analysis and the dynamic evolution of the circuit.

3. The proposed design flow

The proposed flow pursues the dynamic identification of the critical configuration bits as it is represented in figure 1.

Starting from a description of the design and the workload stored as a set of input vectors. The first step is to create a placed and routed design that should be integrated as a pre-routed macro. This design is treated as a re-allocable macro that can be integrated into the FPGA array as a tile. As the FPGA vendors maintain the same internal configuration structure in all the devices belonging to a family, this method allows the

implementation of a design in any device of the same structure.

The second step is the analysis of the bitstream file using the *STAR-discovery* tool. This tool will identify the critical configuration bits that may potentially affect to the design behavior. As the bitstream has a complex organization of each part of the FPGA the size of the database is typically very large, depending on the design, a completely safe design needs larger amount of resources and its performance becomes worse due to the increasing of the internal delays.

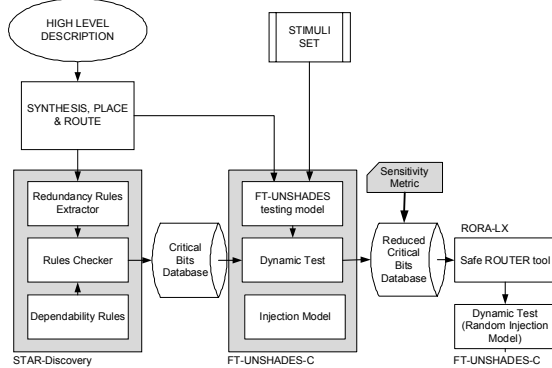


Figure 1. The proposed design flow for implementing tunable fault tolerant circuits.

The third step is a dynamic test for refining the database. This stage is performed by the *FT-UNSHADES-C* tool. The main idea is that, for a particular workload, some configuration bits (CBs) can not affect to the design behavior, because of the actual workload the related resources are not used in any of the clock cycles. These CBs should directly be extracted from the database. Second, CBs are injected at different percentages of the total workload. Some of them should not affect, depending of the time that their related resources are used. Following this method, a classification of the CBs has been produced depending on its incidence in the design behavior.

Fourth step consists of the re-placement and re-routing of the system avoiding the critical CBs. We have defined a reliability level depending on incidence level in the particular workload. *RoRA-LX* is a new adaption of the *RoRA* tool that allows to take into account only the CBs of the database that are classified using the percentage of the workload time. For this reason we have defined a metric that classifies each bit depending on its criticality in the performed test.

Finally a new injection campaign is made for measuring the improvement of the new placed and routed design.

4. The sensitivity metric

In order to measure the fault tolerance degree of the implemented circuit we defined a sensitivity metric that allow to calculate the SEU incidence on each circuit's module considering the number of critical configuration memory bits and the workload executed by the circuit.

When the configuration memory bits are considered the metric of a circuit's module j is computed as the number of critical CBs identified by the STAR discovery tool ($CB_{CRITICAL}$) divided by the total number of configuration memory bit controlling the considered circuit's module ($CB_{MODULES}$). Thus, we defined the Static Sensitivity as $Sensitivity_{Static}(j) = CB_{CRITICAL}(j) / CB_{MODULES}(j)$.

When the executed workload is addressed, we considered that a SEU in a CB acts as a permanent fault until the end of the workload.

Thanks to this analysis, we defined the dynamic sensitivity metric for each circuit module j as $Sensitivity_{Dynamic}(j) = \sum FT_{CB_i}(j)$ (with i ranging from 1 to the total number of critical bits $CB_{CRITICAL}$ identified by the STAR-discovery tool) / $CB_{MODULES}(j)$, where the FT_{CB_i} are the configuration memory bits identified by the STAR-discovery tool. In details, FT_{CB_i} are the configuration memory bits that may affects the behavior of the implemented circuit. Please note that $CB_{MODULES}(j)$ is not an estimation of the circuit's area but only of the configuration memory bits used, thus it is related to the density of the implemented circuit.

The dynamic sensitivity effectively estimates the sensitivity of the SEU effects within the configuration memory of SRAM-based FPGAs taking in account the workload used for the analyzed circuits.

5. The STAR-discovery/RoRA-LX tool chain

The methodology we proposed in this paper in order to develop tunable fault tolerance circuits on SRAM-based FPGAs includes two algorithms: the *Static Analyzer Discovery* (STAR-discovery) tool and an enhanced version of the reliability-oriented place and route algorithm *RoRA* (RoRA-LX). Some preliminary works describing these tools have been presented in [9] and [10]. In this paper we present new versions of both the algorithm.

5.1 The Static Analyzer tool: STAR-Discovery

STAR-Discovery is a new technique predicting the possible impact of SEUs in SRAM-based FPGA systems without resorting to simulation neither fault injection techniques.

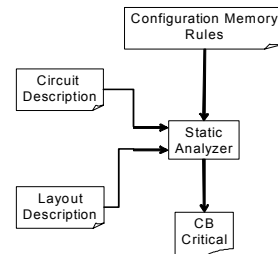


Figure 2. The flow of the proposed STAR-Discovery tool.

The technique is based on a topological inspection of the design implemented using SRAM-based FPGAs. By coupling information about the modification SEUs may

induce in the resources of the used FPGA device with a set of dependability rules, the technique is able to identify all the possible SEUs which modify the circuit topology in such a way that, when a suitable stimuli is applied over the circuit's inputs, the circuit produces erroneous results.

Figure 2 depicts the architecture of STAR-LX, and shows the following elements:

- *Static Analyzer*: it is the tool that checks whether the placed and routed circuit is sensitive to soft errors affecting the configuration memory of the SRAM-based FPGA implementing the circuit.
- *Circuit Description*: it is a file containing the structural description of the circuit, which consists of logic functions (either combinational or sequential) and connections between them.
- *Layout Description*: it is a file containing the description of where each resource in the Circuit Description is placed and routed on the FPGA's area.
- *Configuration Memory Rules*: it is a database of constraints that defines the architecture and the organization of the configuration memory bits, correlating the FPGA's resources with the configuration memory cells.
- *CB critical*: it is a file that lists all the configuration memory bits that violates the configuration memory rules.

Given the Circuit and Layout Descriptions, the STAR-Discovery tool verifies whether configuration memory bit may affect the functionalities of the implemented design. In case it violates the configuration memory rules, it is stored in the CB critical file.

5.2 The reliability-oriented place and route RoRA-LX

The RoRA-LX place and router is an enhanced version of the RoRA algorithm that has been introduced in [10] to solve the problem of crossing error domains of the Triple Modular Redundancy techniques.

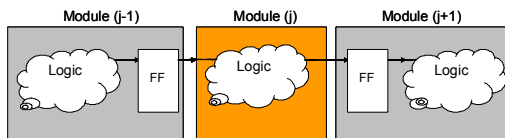


Figure 3. A circuit scenario. The module (j) is considered to be hardened by the RoRA-LX algorithm.

Basically, the RoRA algorithm routes each connection between two logic components through the shortest path it can find. The path is composed of routing Programmable Interconnection Points (PIPs). During path selection, RoRA labels dynamically the routing PIPs, in such a way that it avoids the routing of two connections that may be subject to Short effects or Multiple Opens effects as clearly illustrated in [10].

The RoRA-LX algorithm consists of the last phase of the proposed methodology. It may be applied selectively to a single module of a circuit implemented on SRAM-based FPGAs. Basically, the RoRA-LX algorithm introduces hardening redundancy into the considered circuit module. Thus hypothesizing to harden only a circuit module (j) comprises between the modules (j-1) and (j+1) respectively, as illustrated in figure 3, the RoRA-LX algorithm performs the following phases:

1. It triplicates the logic and interconnections of the module (j) and it places and routes each resources in such a way to avoid any possible domain crossing errors between the distinct replicas.
2. It triplicates the last FF of the circuit module (j-1) in such a way that each replica has its own FF as input.
3. It triplicates the first FF of the module (j+1).
4. It inserts a double voter architecture that consists of three majority voters and a minority voter inserted between the FFs of the module (j+1) as illustrated in figure 4.

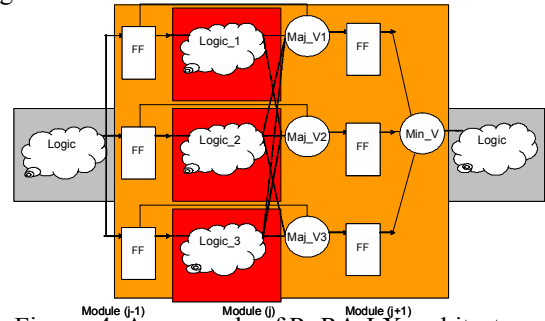


Figure 4. An example of RoRA-LX architecture hardening of the module (j)

The reader should note that all the routing and logic resources of the module (j) and of the several FFs and voters inserted, are placed and routed according to the reliability-oriented rules of RoRA-LX thus in such a way to avoid crossing error domains that can defeat the TMR architecture.

6. The FT-UNSHADES-C platform

FT-UNSHADES system, deeply described in [13], is an FPGA based platform originally dedicated to the study of the reliability of netlists by means of partial reconfiguration techniques. It is based on a computer suite of tools and a dedicated hardware platform based on a Xilinx FPGA of Virtex II family. In the original form of *FT-UNSHADES* system, faults (SEUs) are injected as bit-flips into one or more user registers of the design. Packets of configuration bits containing the actual value of a particular register are uploaded from the hardware platform, processed, and downloaded with new values. *FT-UNSHADES* software is fed with the *bit allocation file* obtained from the Xilinx standard design flow. In this file the physical allocation inside the FPGA of each design register is reported and associated to its logical name, resulting from the high level synthesis process.

The target register is selected from the complete set of user registers and memories that compound the design, before the injection. Other injection tools based on FPGAs perform the injection in a blind way, because the injection process is done register by register and the identification of each fault is made when the injection campaign is finished.

6.1 The fault injection model

In *FT-UNSHADES* the complete the design under study, called Module Under Test (MUT) is instanced twice and only one is attacked, the other instance is considered as reference for comparison, as described in figure 5. On board SRAM memories store the stimuli to produce the workload. Figure shows the injection model. It can be seen the two instances plus the system clock controller useful to decide when the bit-flip is going to be injected. The event detection allows knowing when the fault has been propagated to a primary output, to classify it.

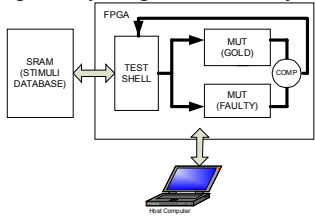


Figure 5. The architecture of the developed FT-UNSHADES-C platform.

Each injection is made following these steps:

1. Time of injection is selected. One or more registers are selected to change its actual value when time is reached.
2. The system is programmed to stop at the injection time.
3. The system reaches the injection time and stops the clock. The bit-flips are induced into the selected registers. The system is resumed until an event is detected at the outputs.
4. Fault is classified.

Note that there are no restrictions on time, locations and how one or several faults are injected.

6.2 FT-UNSHADES-C over configuration

FT-UNSHADES-C is the platform oriented to produce the bit-flips of the configuration memory bits. The faulty design is treated as a pre-routed macro, instead of a post synthesis macro. Note that the gold instance can be a post synthesis macro.

From *STAR-Discovery* tool, the configuration bits are identified as register location within the standard *bit allocation file*, and significant names are given to identify, inside the design hierarchy, which configuration bit is attacked at a time and its functionality.

FT-UNSHADES produces results over the selected CBs while the design is executed. In order to define the category of each CB, we test the total database for the complete workload. Experimental results show that in the dynamic test, some CBs are not sensitive during the complete workload, or what is the same, some CBs

continue being critical for this particular workload. It is possible to determine the most critical part affecting the design.

At the end of the experiment, a new analysis is done to verify that the design obtained from *RORA-LX* tool is now more dependable than previous version. In this new analysis, the *FT-UNSHADES-C* has been used without any restriction in order to emulate the radiation environment, using a completely random selection of the CBs.

7. Experimental evaluations

The new methodology has been tested using a highly sensitive design. We have used a pure straight forward pipelined structure of a CORDIC processor. This design has been used with different workloads to show that it influences the configuration database obtained from the dynamic test. An implementation has been made on a Xilinx XC2V3000 device. The characteristics of the implemented circuits have been reported in Table 1 where we have indicated the number of resources including the Flip-Flops (FFs), the 4-input LUTs and the I/O pins, for the unhardened CORDIC core processor and for the RoRA-LX hardened version. Please note that, the CORDIC processor uses about the 25% of the XC2V3000 available resources, also the CORDIC has been placed and routed in a constrained area on the DUT FPGA in such a way to stress the RoRA-LX capabilities.

Resources	FFs [#]	4-input LUTs [#]	I/O pins [#]
CORDIC core plain	3,315	3,246	225
CORDIC core RoRA-LX	3,524	3,750	225

Table 1. Characteristics of the implemented CORDIC core processor.

CORDIC modules	CB Critical	CB Module	$Sensitivity_{Static}$
S5	3,636	6,210	0.59
S6-S1	5,151	8,322	0.62
S6-S2	37,206	52,460	0.71
S6-S3	5,136	10,224	0.50
S6-S4	5,176	10,256	0.50
S6-S5	5,127	10,180	0.50
S6-S6	5,418	7,403	0.73
S6-S7	4,999	11,301	0.44
S6-S8	5,363	8,010	0.67
S7	3,032	5,640	0.53

Table 2. The STAR-Discovery results reported

We executed the static analysis using the STAR-Discovery tool and then the dynamic evaluation of the identified CBs. The results of this static analysis are reported in Table 2 where are reported the critical CB, the total number of bit used for each module and the computed $Sensitivity_{Static}$, while the dynamic analysis

results performed by FT-UNSHADES-C have been reported in the Table 3.

According to the Static and Sensitivity metric, since the most sensitive modules are S6-S6 and S6-S7, we choose to apply the RoRA-LX algorithm to these CORDIC's modules. We then performed the static analysis and the dynamic evaluation of the tuned and hardened circuit. The results of the last analysis are reported in Table 4. The reported results show that the number of critical CBs of the module S6-S6 and S6-S7 have been reduced to zero. Viceversa, it is observed an increasing of only 3 bits in the module S6-S5 and 1 related S6-S8. This is due to the hardened structure inserted by the RoRA-LX algorithm. The RoRA-LX algorithm correctly harden the modules S6-S6 and the module S6-S7, however the architecture has still two points of failure related to the input and output logic between the hardened module. However, the RoRA-LX reduces drastically the number of critical CBs thus increasing the fault tolerance capability of the circuits. The obtained results demonstrated also that the proposed module is able to tuning the fault tolerance capability of the circuit according to the selected hardened modules.

Thus considering the area overhead introduced by RoRA-LX, we have an increasing of 15% of the circuit area while having a reduction of the SEUs sensitiveness of more than the 50% considering the results reported in the table 4.

CORDIC modules	ΣT_{pi}	$Sensitivity_{Dynamic}$
S5	838	0.23
S6-S1	960	0.18
S6-S2	2,587	0.07
S6-S3	0	0
S6-S4	0	0
S6-S5	0	0
S6-S6	2,604	0.48
S6-S7	2,019	0.40
S6-S8	1,210	0.23
S7	1	0

Table 3. The computed Dynamic Sensitivity on the plain version of the CORDIC core.

CORDIC modules	ΣT_{pi}	$Sensitivity_{Dynamic}$
S5	838	0.23
S6-S1	960	0.18
S6-S2	2,587	0.07
S6-S3	0	0
S6-S4	0	0
S6-S5	3	0
S6-S6 RoRA-LX	0	0
S6-S7 RoRA-LX	0	0
S6-S8	1,212	0.23
S7	1	0

Table 4 The computed Dynamic Sensitivity for the hardened circuit.

8. Conclusions

In this paper, we have proposed an innovative design flow that allows the implementation of tunable fault tolerance circuits on SRAM-based FPGA devices. We described a new metric that allows the estimation of the fault tolerance degree of the analyzed circuit considering both the static characteristics of the FPGA architecture and the dynamic sensitivity of the applied workload. We have presented experimental results on a real-world industrial-type case study demonstrating the feasibility of the developed methodology. Future activities will be focused on performing more experimental analysis based on fault injection on more complex circuits, analyzing the performance and to perform accelerated radiation ground tests.

9. References

- [1] M. Nikolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies", IEEE 17th VLSI Test Symposium, April 1999, pp. 86 – 94.
- [2] J. R. Srour, J. M. McGarrity, "Radiation effects on microelectronics in space", proceedings of the IEEE, Vol. 76, Issue 11, Nov. 1988, Pages. 1443 – 1469.
- [3] J. Rose, A. El Gamal, A. Sangiovanni-Vincetelli, "Architecture of Field-Programmable Gate Arrays"; IEEE proceedings, vol. 81, no. 7, July 1993, pp. 1013 – 1029.
- [4] E. Normand, "Single Event Upset at Ground Level", IEEE Transactions on Nuclear Science, vol. 43, No. 6, Dec. 1996.
- [5] M. Ohlsson, P. Dyreklev, K. Johansson, P. Afke, "Neutron single event upsets in SRAM-based FPGAs", Radiation Effects and Data Workshop, 1998, pp. 177 – 180.
- [6] M. Wirthlin, E. Johnson, N. Rollins, M. Caffrey, and P. Graham, "The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets", Proceedings of the 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 133 – 142, 2003.
- [7] C. Carmichael, "Triple Module Redundancy Design Techniques for Virtex FPGAs", Xilinx Application Notes XAPP197, 2001.
- [8] F. Lima Kastensmidt, G. Neuberger, R. Hentschke, L. Carro, R. Reis, "Designing Fault-Tolerant Techniques for SRAM-based FPGAs", IEEE D& T of Computers, Nov-Dec 2004, pp. 552 – 562.
- [9] L. Sterpone, M. Violante, "A New Analytical Approach to Estimate the Effects of SEUs in TMR Architectures Implemented Through SRAM-based FPGAs", IEEE Transactions on Nuclear Science, Vol. 52, No. 6, December 2005, pp. 2217 – 2223.
- [10] L. Sterpone, M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs", IEEE Transactions on Computers, Vol. 55, Issue 6, June 2006, pp. 732 – 744.
- [11] M. Sonza Reorda, L. Sterpone, M. Violante, F. Lima Kastensmidt, L. Carro, "Evaluating different solutions to design fault tolerant systems with SRAM-based FPGAs", JETTA: Journal of Electronic Testing: Theory and Applications, Kluwer Academic Publisher, Vol. 23, No. 1, February, 2007, pp. 47 – 54.
- [12] P. K. Samudrala, J. Ramos, S. Katkooi, "Selective Triple Modular Redundancy (STMR) Based Single-Event Upset (SEU) Tolerant Synthesis for FPGAs", IEEE Transactions on Nuclear Science, Vol. 51, Issue 5, October 2004.
- [13] B. Pratt, M. Caffrey, P. Graham, K. Morgan, M. Wirthlin, "Improving FPGA Design Robustness with Partial TMR", IEEE 44th Annual International Reliability Physics Symposium, San Jose, 2006.
- [14] M.A. Aguirre, J. N. Tombs, F. Muñoz, V. Baena, H. Guzmán, J. Nápoles, A. Torralba, A. Fernández-León, F. Tortosa-López, D. Merodio. "Selective Protection Analysis Using a SEU Emulator: Testing Protocol and Case Study Over the Leon2 Processor", IEEE Trans. On Nuclear Science. Volume 54, Issue 4, Aug 2007 pp. 951 - 956