A Framework for Implementing Radiation--Tolerant Circuits on Reconfigurable FPGAs

Manuel G. Gericota, Luís F. Lemos, Gustavo R. Alves, José M. Ferreira

Abstract— **The outstanding versatility of SRAM-based FPGAs make them the preferred choice for implementing complex customizable circuits. To increase the amount of logic available, manufacturers are using nanometric technologies to boost logic density and reduce prices. However, the use of nanometric scales also makes FPGAs particularly vulnerable to radiation-induced faults, especially because of the increasing amount of configuration memory cells that are necessary to define their functionality.**

This paper describes a framework for implementing circuits immune to radiation-induced faults, based on a customized Triple Modular Redundancy (TMR) infrastructure and on a detection-and-fix controller. This controller is responsible for the detection of data incoherencies, location of the faulty module and restoration of the original configuration, without affecting the normal operation of the mission logic.

A short survey of the most recent data published concerning the impact of radiation-induced faults in FPGAs is presented to support the assumptions underlying our proposed framework. A detailed explanation of the controller functionality is also provided, followed by an experimental case study.

Index Terms— **Fault location, Fault diagnosis, Fault tolerance, Fault mitigation, Reconfigurable systems**

I. Introduction

THE introduction of Very Large Scale Integration (VLSI) technologies raised substantially the reliability of electronic systems, when compared with the previous use of discrete components. Hence, the use of fault tolerance techniques was confined only to specific applications requiring high levels of reliability or operating on harsh environments. Shrinking transistors' size leads to a greater integration and to a per unit power reduction, enabling chips to grow both in size and complexity. But new nanometer scales also brought negative aspects, such as a high sensitivity to radiation-induced faults, which affects values stored in memory cells. Therefore, this kind of faults has a particular impact on the reliability of SRAM-based Field Programmable Gate Arrays (FPGAs). The exponential growth in the number of memory cells needed for configuration makes these devices especially vulnerable to radiation-induced faults, such as Single Event Upsets (SEU) and Multi-Bit Upsets (MBU) [1-4]. Although these faults do not physically damage the chip, their effects are permanent, since the functionality of the circuits mapped into the device is permanently altered.

Although anti-fuse technology FPGAs are less prone to SEUs due to the absence of configuration memory cells, SRAM-based FPGAs have been the preferred choice in space missions, like the MARS 2003 Lander and Rover vehicles. The reason was because their processing performance is 10 to 100 times higher than the performance attained by anti-fuse technology FPGAs, and also due to their reconfigurable features. These features enable resource multiplexing, updating of algorithms during long space missions (avoiding mission obsolescence), and correction of design flaws while in orbit [5].

In non-reconfigurable technologies, such as ASICs, protection against SEUs is restricted to flip-flops, because logic paths between them are typically hard-wired. Nevertheless, Single Event Transients (SETs) — a charge transient induced in a wire by the incidence of an heavy ion — may be propagated to flip-flop inputs, where they have a high probability to be registered, causing soft-errors in the user data. Besides, if an SET strikes a clock line, double-clocking may occur, leading to an extemporaneous update that may affect part of or all the flip-flops driven by that line (depending on the charge value and on line attenuation). Further protection may only be achieved through full module redundancy. This is also a preferred choice to improve the reliability of highly critical applications based on FPGAs [4-7]. Due to their inherent configurability, FPGAs are especially suitable for the implementation of modular redundancy, since it does not require any architectural innovation and it is function-independent. However, and because these devices rely on memory cells to define logic paths, they are also susceptible to SEUs. Again, in this case, the only effective protection is full module redundancy [6].

In a discrete implementation of a Triple Modular Redundancy (TMR) system, if a defect affects the functionality of a single module, reliability decreases, but the system will continue to work correctly. However, a second failure in one of the remaining modules will lead to a system failure. Ideally, when a module fails, it should be replaced to restore the initial redundancy, but this action may not be

This work is supported by an FCT program under contract POSC/EEA-ESE/55680/2004.

Manuel G. Gericota Luís F. Lemos and Gustavo R. Alves are with the Department of Electrical Engineering, School of Engineering - Polytechnic Institute of Porto, Rua Dr. Antonio Bernardino de Almeida, 4200-072 Porto, Portugal (e-mail: {mgg, lfl, galves}@isep.ipp.pt).

José M. Ferreira is with the Department of Electrical and Computer Engineering, Faculty of Engineering of the University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal (e-mail: jmf@fe.up.pt).

possible immediately. In certain cases, like in space applications, it may even be impossible. In the case of FPGA--based systems, a significant improvement in reliability may be achieved without a significant rise in costs — in the event of a module failure, the initial redundancy may be restored by reconfiguration of the affected module. No physical replacement is therefore necessary.

This paper presents a set of rules for a new framework for implementing circuits immune to radiation-induced faults in FPGAs, based on its confinement, detection, location and mitigation. The proposed framework is built around a customised TMR implementation, associated to a fault detection-and-fix controller. This controller is responsible for:

(i) detecting data incoherencies;

(ii) locating the faulty redundant module; and

(iii) restoring the original module configuration, fixing it without affecting the normal operation of the functional logic.

This mechanism was implemented on a XC2V1500, a device that belongs to the Virtex-II FPGA family from Xilinx. Our proposed approach enables the confinement and detection of faulty modules, and the determination of when reconfiguration must be applied to restore proper system operation, before cumulative errors, induced over time, lead to its failure. A short survey of recent literature concerning the impact of radiation-induced faults on FPGAs and on FPGA based TMR implementations, is reviewed to support the options assumed during the implementation phase. A discussion concerning implementation issues, mainly related to design options and architectural features of the FPGA, which may prevent an efficient implementation of the proposed framework, are also presented. The work herein presented is part of a broader project, addressing the design of FPGA based self-healing circuits. Practical implementation aspects are also pointed out, and current and future research lines are presented in the concluding section.

II. PREVIOUS OBSERVED RADIATION EFFECTS ON FPGAS

The results of several radiation campaigns in SRAM-based FPGAs, carried out with the objective of understanding the effects of radiation-induced faults, were reported by several authors [2, 3, 7]. These authors observed that, in general, radiation changes the correct functionality of the circuits, an effect defined as a Single Event Functional Interrupt (SEFI). A classification of SEFIs according to the affected resources and their effects was proposed in [1-2].

Several fault injection approaches, proposed as alternatives to (expensive) radiation campaigns, may also be found in the literature. In these papers the effects of SEUs are emulated as bit-flips in the bitstream of the configuration memory of the FPGA, either through changes in the original configuration bitstream or at run-time, through dynamic reconfiguration [8, 9]. The greatest advantage of these methods is the higher controllability of the experiments, in contrast to the unpredictability of radiation injection, which enables a better diagnostic of the effects of each SEU. A combination of both techniques, not only to increase the controllability of the experiments, but also to verify the accuracy of the emulation fault injection techniques used, may be found in [4, 5, 10, 11].

Lately, several hardening techniques have been proposed to avoid SEU effects on the functional behavior of circuits. Correcting techniques based on dynamic reconfiguration, known as scrubbing, like those presented on [12-14], periodically read back the configuration memory to detect bit-flips caused by SEUs. If a bit-flip is detected, the affected frame is reconfigured and the system is reset. However, the same authors recognized some limitations to these techniques: a fault-free read-back of the configuration bitstream does not always guarantee that a SEU did not occur. As an example, SEUs or SETs affecting flip-flop states occur without upsetting the bitstream, but may severely disturb or even halt function operation. Another drawback is fault detection latency. Reading back the whole configuration memory may take from several milliseconds to a few hundred milliseconds, depending on the size of the FPGA and on the interface used to perform the read-back operation. By then, the fault may already have caused the irreversible malfunctioning of the whole system. In some cases, it may even be impossible to recover from this situation.

Alternative techniques based on hardware redundancy were proposed without the aim of identifying and correcting the fault, but just to mask its existence. Through extensive TMR testing, several authors have shown that SEU-induced failures can be properly controlled for the Virtex family of FPGA devices [6, 7, 11, 15]. Fault tolerance is achieved using extra components to instantaneously mask the effect of a faulty component, meaning that no fault propagation will occur. Still, as no fault detection occurs, the faulty module is not replaced and therefore the initial redundancy (and reliability) is not restored. Consequently, over time, cumulative faults will increase the probability of a general system failure.

The consideration of the results achieved during radiation campaigns concerning MBUs, due to single charged particles, is also important, since they may potentially affect multiple redundant modules and produce incorrect values. The effects produced by MBUs are intrinsically related to the architecture of the configuration memory. In Virtex families, configuration memory is divided into one bit wide vertical frames that span from the top to the bottom of the array. Each column of Configurable Logic Blocks (CLBs) comprises multiple frames, which combine internal CLB configuration and state information, with column routing and interconnection information. In [5] it is reported that MBUs in Virtex devices occurred all in the same configuration frame, while in the Virtex-II family, the percentage of MBUs that occurred in the same configuration frame decreases to 88%. However, no MBUs spanned the configuration data of separated resource columns [4]. No correlation was observed between MBUs and module granularity sizes, which indicates that even at very fine granularities, if the modules are placed far enough so as not to share routing networks, TMR is still a good option. These results also reveal important information about the placement of the configurable memory cells inside the FPGA. This information is important to understand the fault induction mechanism due to radiation.

In summary, the association between dynamic reconfiguration and TMR seems to be the most effective way to mitigate the effects of radiation (although extra care is required during the mapping of the circuits into the FPGA).

The experimental results and conclusions reviewed above were taken into account when developing our proposed framework for the design and implementation of radiation immune FPGA-based circuits.

III. FRAMEWORK RULES

Effective protection of an FPGA-based circuit against radiation requires a TMR design. In addition, it has to incorporate an autonomous mitigation mechanism to avoid circuit failures due to the cumulative effects of SEUs.

In a classic TMR implementation [16], the correct circuit output values are settled by voting elements that accept the outputs from three redundant sources and deliver the majority vote at their outputs. To ensure a consistent reliability index, voters have also to be replicated, in a scheme known as T-TMR [16]. T-TMR implementations mask any single fault emerging during circuit operation. Multiple faults may also be masked, providing that **i)** they affect only one of the redundant modules or voters, **ii)** if affecting different modules, they involve different signals and bitwise comparison is used. In these cases, faults are confined to the module or voter where they emerged, and are not visible from its outside.

To fully prevent functional problems caused by configuration upsets, each signal should enter the FPGA in triplicate, using three input pins [15]. Otherwise, if a single input was connected to all three redundant modules, then a failure at the single input would cause the error to propagate through all the redundant modules, and thus the error would not be masked.

This same principle applies to clock signals. Each of the triplicate circuit modules should receive its own clock. Otherwise, spurious signals induced by SETs on a single clock line may lead to an extemporaneous update of all the three-module registers and to the asynchronous output of possibly incorrect values.

Output signals should also leave the FPGA in triplicate, with minority voters monitoring each output [15]. The three signals converge to a same node outside. When one output is different from the others, the correspondent pin is driven to high impedance.

To avoid the effect of MBUs on the different modules [4], the three redundant functional modules should be placed in different columns of the FPGA. This means that the FPGA should be divided into four vertical areas: three for the functional circuit modules and a fourth area for placing the detection-and-fix controller. The interconnections between a module and its own Input/Output Blocks (IOBs) should not cross other modules'area. The overall implementation scheme proposed is illustrated in figure 1.

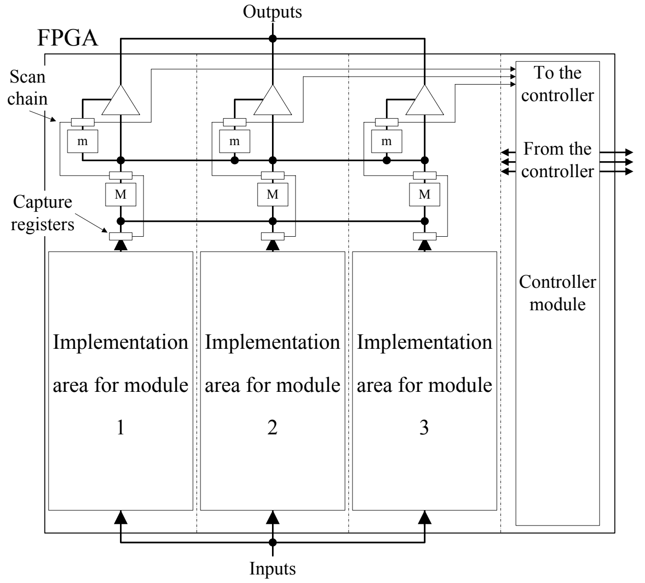

Fig. 1. Proposed framework overview.

When one or more faults appear in one of the modules or voters, the T-TMR implementation confines the fault and masks its existence, avoiding its propagation to the rest of the circuit. However, the cumulative effects of several faults induced over time may suppress the effectiveness of the confinement and masking mechanism, allowing fault propagation. With the aim of detecting the emergence of faults a detection-and-fix controller is implemented in the fourth area defined on the FPGA logic space. A detailed overview of this controller structure is shown in figure 2.

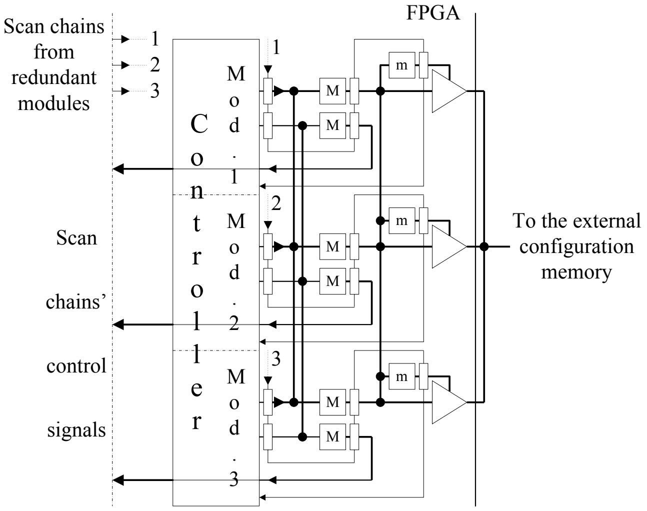
FPGA
Scan chains from redundant modules
1
2
3
Controller
Mod. 1
Mod. 2
Mod. 3
1
2
3
M
m
Scan chains' control signals
To the external configuration memory

Fig. 2. Overview of the detection-and-fix controller structure.

The detection-and-fix controller is responsible for detecting data incoherencies, locating the faulty module and restoring the original configuration. This is done transparently, through partial reconfiguration of the affected functional module, without human intervention, since physical component replacement is not needed. As a result, a higher level of maintainability is achieved without implying the inoperability of the circuit.

IV. Fault Detection, Location and Mitigation

This last point implies not only the existence of redundancy but also of a mechanism able to detect the emergence of an induced fault. It is very hard to detect a fault in a T-TMR implementation using traditional online test strategies, since the redundancy of the circuit masks its effect. In our approach, the detection of the faulty modules is done via three scan chains that regularly capture the values at the outputs of the modules and voters.

A Boundary-Scan (BS)-like register [17] is used to implement the scan chain, composed of simpler cells comprising only a capture / shift stage, as shown in figure 3. The absence of the latch stage means that no delay is introduced in the signal's path by the scan chain. To avoid capturing undefined values, the scan chain is updated synchronously with the system clock (assuring that modules or voters outputs will be in a steady state when they are captured).

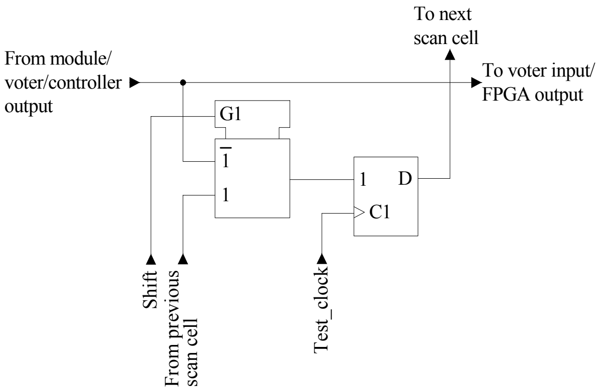

Fig. 3. Observe-only BS cell (comprising only the capture / shift stage).

The scan chain control signals are generated by the detection-and-fix controller. This controller regularly updates the scan chains and shifts its contents, comparing the output values. Our framework uses three parallel scan chains, each covering a different module. This approach makes it easier for the controller to accurately diagnose which of the three module areas was affected by a fault, and to trigger its reconfiguration. More than one scan chain in parallel also has the additional advantage of decreasing fault detection latency, since the shifting time is divided by the number of parallel chains (enabling more frequent capture operations).

The sequence of tasks carried out by the detection-and-fix controller is represented in the flowchart shown in figure 4. This sequence is continuously repeated in search of emerging faults (either in the controller or in the user modules). The serial bitstreams captured through the scan chains are shifted to the internal controller where they are compared, bit-by-bit.

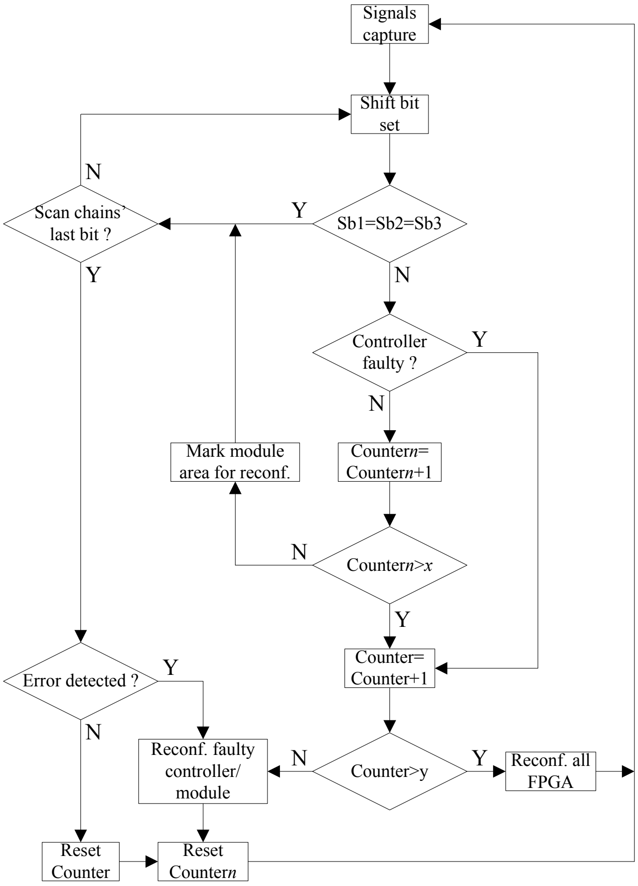

Fig. 4. Detection-and-fix controller flowchart.

If an incoherency is detected, the module or voter where it was found is probably faulty. Obviously, the controller and the scan chains may also be affected by SEUs. To ensure their correct operation, the controller is equally implemented using a T-TMR design and its modules and voter output signals are also covered by the scan chains, creating a self-verifiable circuit. The option of concentrating the controller in only one area, despite being implemented in T-TMR, was taken to reduce complexity and the number of occupied CLB columns. However, since it occupies fewer slices than those available in each column, a convenient separation between modules was implemented.

The first bits of the scan chain belong to the outputs of the controller. If an incoherency is detected in those first bits, the controller will be fully reconfigured at once. This procedure guarantees that the controller is working properly. While not being a critical component (concerning the functionality of the system), a fault-free controller is mandatory to avoid the accumulation of errors and the subsequent system failure.

If an incoherency is detected on an output of one of the modules or voters, the area where it is implemented will be reconfigured after the last bit of the scan chains has been shifted. If several incoherencies are detected in the same module, the module is reconfigured after a parameterizable number of errors, even before reaching the last bit of the scan chains. A new capture operation is then performed and the verification process restarted.

Of course, if an upset affects the values shifted through the scan chain, this will falsify fault diagnosis and consequently trigger an extemporaneous reconfiguration of one the modules. This operation, although unnecessary, will not affect the operation of the system.

A more complicated situation takes place if the structural configuration of the scan chains is affected by a fault. In this case, several neighboring bits will be disturbed, falsely indicating that a general failure in one or more modules occurred. Additionally, it won't be possible to locate the place where the faults emerged. Therefore, after the detection of a parameterizable number of errors, either in the controller or in the modules, the controller undertakes a full dynamic reconfiguration of the FPGA and completely restores the scan chains.

The exact location of the faulty module or voter, enables the controller to activate the partial reconfiguration and restoration of the faulty module. An external memory stores the original partial configuration files concerning the four defined areas. Due to the volatility of the FPGA configuration memory, this external memory was already necessary to hold the FPGA configuration bitstream (to be uploaded during system power up).

The inclusion of a fault detection mechanism improves the performance of the recovery procedure. In this case, scrubbing takes place only when necessary and on a well defined target. Bearing in mind the intervals between the occurrence of SEUs, even in space applications [12], this solution enables considerable power savings when compared with periodic "blind" full reconfiguration.

SEUs that do not upset the bitstream, like those affecting flip-flop states that cannot be removed by reconfiguration, will also be detected. As mentioned before, this means that scrubbing by itself can not ensure fault-free operation, and that TMR is always needed to avoid fault propagation. However, due to the transient nature of upsets, the soft error will be recovered by the circuit when the affected flip-flop is again updated.

V. Case study

To evaluate the effectiveness of our approach, a twenty four-bit counter was implemented in a XC2V1500-based prototyping board, according to the rules defined in our proposed framework. The detection-and-fix controller used a total of 254 slices, distributed across two of the 40 available CLB columns, representing an area overhead of 5%. Notice that this overhead is constant and independent of the size or the complexity of the circuits implemented on the FPGA. The remaining 38 columns were divided in three areas of 12 columns each, leaving a total of 2304 slices available for the implementation of each user module. The extra two columns (remainder of the division of 38 by 3) were placed among the three areas of 12 columns, to reinforce protection against column-spanning MBUs.

Each module area enables the implementation of circuits far more complex than the one used to validate the proposed solution. The incorporation of the scan chain implied an overhead of 3 slices per module output, necessary to capture the module output and the outputs of the corresponding majority and minority voter. The overhead is therefore dependent on the number of outputs of the user circuit and not on its complexity. In case of fault detection, the detection-and--fix controller initiates the partial reconfiguration of the affected area, by resolving the location address of the file to be configured. Our prototyping board uses SystemAce [18] from Xilinx to keep trace of the partial configuration files and to configure the FPGA. However, different kinds of interfaces may be used to provide the partial reconfiguration files, including remote sources. The partial reconfiguration files were generated using the Foundation tools from Xilinx.

The dynamic reconfiguration of part or of the whole FPGA does not affect the operation of the functions whose functionality is not changed, even if they are active and if its placement area is covered by the reconfiguration procedure. In other words, the mitigation procedure is completely transparent.

The maximum speed of operation achieved by the detection-and-fix controller was 200MHz. Since capture operations must be synchronous with the operation of the user's circuit, this frequency also defines the maximum operating speed of the circuit.

Several tests based on localized fault injection through partial reconfiguration proved the effectiveness of the proposed concept. In addition, a random fault injection procedure is under way to better simulate real working conditions. An FPGA configuration frame is picked out randomly and a bit flip is also randomly performed in one of its configuration bits. The FPGA is then partially reconfigured using the altered bitstream.

VI. Conclusion

This paper presented a framework for the confinement, detection and mitigation of radiation-induced faults in FPGAs, built around a customised TMR implementation. Several issues addressing the effectiveness of TMR to cope with radiation-induced faults were reviewed and discussed. Based on a compilation of experimental data reported by several authors, it was shown that T-TMR plus scrubbing is the most effective approach to mitigate radiation-induced faults in FPGAs and to extend the reliability of the implemented circuits. According to the same experimental data, several techniques were listed to improve the effectiveness of T-TMR implementations and a set of rules was proposed to get the most from a T-TMR implementation in terms of radiation-induced fault protection. A practical case-study enabled the quantification of the overhead of our proposed solution and the assessment of its effectiveness. Further work is being done to better evaluate the behavior of the circuit.

REFERENCES

[1] L. Sterpone, M. Violante, "Analysis of the Robustness of the TMR Architecture in SRAM-Based FPGAs", *IEEE Trans. on Nuclear Science*, Vol. 52, No. 5, pp. 1545-1549, October 2005.

[2] M. Ceschia et al., "Identification and Classification of Single-Event Upsets in the Configuration Memory of SRAM-Based FPGAs", *IEEE Transactions on Nuclear Science*, Vol. 50, No. 6, pp. 2088-2094, December 2003.

[3] M. Bellato et al., "Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA", *Proc. of the Design, Automation and Test in Europe Conf.*, pp. 584-589, 2004.

[4] H. Quinn, P. Graham, J. Krone, M. Caffrey, S. Rezgui, C. Carmichael, "Radiation-Induced Multi-Bit Upsets in Xilinx SRAM-Based FPGAs", *Proc. Military and Aerospace Appl. of Prog. Logic Devices Conf.*, 2005.

[5] M. French, P. Graham, M. Wirthlin, Li Wang, G. Larchev, "Radiation Mitigation and Power Optimization Design Tools for Reconfigurable Hardware in Orbit", *Proc. of the Earth-Sun System Technology Conference*, 2005.

[6] C. Carmichael, E. Fuller, P. Blain, M. Caffrey, "SEU Mitigation Techniques for Virtex FPGAs in Space Applications", *Proc. Military and Aerospace Applications of Prog. Logic Devices Conf.*, 1999.

[7] E. Fuller, M. Caffrey, C. Carmichael, A. Salazar, J. Fabula, "Radiation Testing Update, SEU Mitigation, and Availability Analysis of the Virtex FPGA for Space Reconfigurable Computing", *Proc. Military and Aerospace Appl. of Prog. Logic Devices Conf.*, 2000.

[8] F. Lima, C. Carmichael, J. Fabula, R. Padovani, R. Reis, "A Fault Injection Analysis of Virtex FPGA TMR Design Methodology", *Proc. 6th European Conf. on Radiation and its Effects on Components and Systems*, pp. 275-282, 2005.

[9] M. Rebaudengo, M. S. Reorda, M. Violante, "Simulation-based analysis of SEU effects on SRAM-based FPGAs", *Proc. of the 12th Intl. Conf. on Field-Prog. Logic and Applications*, pp. 607-615, 2002.

[10] M. Wirthlin, E. Johnson, N. Rollins, M. Caffrey, P. Graham, "The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets", *Proc. 11th IEEE Symp. on Field-Prog. Custom Computing Machines*, pp. 133-142, 2003.

[11] G. M. Swift et al., "Dynamic testing of Xilinx Virtex-II field programmable gate array (FPGA) input/output blocks (IOBs)", *IEEE Trans. on Nuclear Science*, Vol. 51, No. 6, pp. 3469-3474, December 2004.

[12] M. Gokhale, P. Graham, E. Johnson, N. Rollins, M. Wirthlin, "Dynamic reconfiguration for management of radiation-induced faults in FPGAs", *Proc. 18th Intl. Parallel and Distributed Processing Symp.*, pp. 145-150, 2004.

[13] M. Abramovici, C. Stroud, C. Hamilton, S. Wijesuriya, V. Verma, "Using Roving STARs for On-Line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications", *Proc. of the Intl. Test Conference*, pp. 973-982, 1999.

[14] M. G. Gericota, G. Alves, M. L. Silva, J. M. Ferreira, "Active Replication: Towards a Truly SRAM-based FPGA On-Line Concurrent Testing", *Proc. of the 8th IEEE Intl. On-Line Testing Workshop*, pp. 165-169, 2002.

[15] Triple Module Redundancy Design Techniques for Virtex FPGAs, *XAPP 197 Application Note*, Xilinx, 2001.

[16] P. K. Lala, *Self-Checking and Fault-Tolerant Digital Design*. San Francisco, CA: Morgan Kaufman Publishers, 2001.

[17] IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std 1149.1), *IEEE Std. Board*, June 2001.

[18] System ACE MPM Solution, *Product Specification*, Xilinx, 2003.