Evaluation of Single Event Upset Mitigation Schemes for SRAM based FPGAs using the FLIPPER Fault Injection Platform *

M. Alderighi [1], F. Casini[2], S. D'Angelo[1], M. Mancini[1], S. Pastore[2], G.R. Sechi[1], R. Weigand[3]

[1] INAF, Istituto di Astrofisica Spaziale e Fisica Cosmica, Milano, ITALY
[2] Sanitas EG s.r.l., Milano, ITALY
[3]ESA/ESTEC, European Space Agency, Noordwijk, THE NETHERLANDS

Abstract

SRAM based reprogrammable FPGAs are sensitive to radiation-induced Single Event Upsets (SEU), not only in their user flip-flops and memory, but also in the configuration memory. Appropriate mitigation has to be applied if they are used in space, for example the XTMR scheme implemented by the Xilinx TMRTool and configuration scrubbing. The FLIPPER fault injection platform, described in this paper, allows testing the efficiency of the SEU mitigation scheme. FLIPPER emulates SEU-like faults by doing partial reconfiguration and then applies stimuli derived from HDL simulation (VHDL/Verilog test-bench), while comparing the outputs with the golden pattern, also derived from simulation. FLIPPER has its Device-Under-Test (DUT) FPGA on a mezzanine board, allowing an easy exchange of the DUT device. Results from a test campaign are presented using a design from space application and applying various levels of TMR mitigation.

1. Introduction

Field Programmable Gate Arrays based on SRAM (SRAM-FPGAs) have gained a primary role in several application areas due to their high density and unlimited on-field re-configuration capability. Nevertheless when used in high reliability applications and specifically space applications, the Single Event Effects (SEEs) have to be addressed [1] [2]. Single Event Upsets (SEU) are of particular concern, because in SRAM-FPGA they affect not only flip-flops and RAM blocks of the user design, but also the device configuration memory, they can therefore change the logical function of the circuit. Many mitigation techniques are available to designers for dealing with SEU, among them Triple Modular Redundancy (TMR) is a well known approach to design hardening [3] [4]. Xilinx provides a tool for automatic implementation of a TMR scheme (XTMR) suitable for their SRAM-FPGA devices, called TMRTool [9]. To prevent multiple error accumulation, TMR should be used in conjunction with a periodic configuration memory refresh, also referred to as scrubbing [10].

For complexity reasons (XTMR increases the FPGA resource usage by a factor > 3), mitigation may have to be applied only on part of the design, or different mitigation schemes are mixed. Prediction of the resulting overall SEU sensitivity is then difficult, and ground radiation testing is often an unavoidable step. Fault injection techniques like the hardware fault emulation based on partial re-configuration [5], or the laser fault injection [6] are usually cheaper to implement, achieve a higher upset rate, and they are more deterministic with respect

* The work is funded by the European Space Agency under contract 18559/04/LvH/gm.

1550-5774/07 $25.00 © 2007 IEEE
DOI 10.1109/DFT.2007.45

to the SEU locations; they can be applied to prepare or to complement radiation tests. The circuit robustness can also be assessed by static analysis, based on the topological analysis of the placed circuit [7].

In this paper, the FLIPPER fault injection platform [8] for evaluating the SEU effects in SRAM-FPGA and an analysis of a sample design using FLIPPER are presented. FLIPPER allows evaluating the SEU sensitivity of a design, for example, collecting a probability distribution of the number of randomly injected faults in configuration memory necessary to cause a functional fault. This information can be also useful for defining the scrubbing rate of the configuration memory.

The target device is a Xilinx XQ2VR6000 and the sample design is a CUC-CTM VHDL IP-core from the European Space Agency (ESA) that provides datation and alarm services for space applications. The original (plain) version of the design is analysed along with the XTMR protected design, the latter in two variants depending on the design output protection scheme: double or triple voted [9].

In Section 2 related and previous works are presented, while the FLIPPER system is described in Section 3. The sample design is sketched in Section 4, and the fault injection experiment and its results are given Section 5. Finally conclusions are drawn in section 6.

2. Related Work

Various systems have been reported in the literature for the evaluation of SEU fault tolerance of VLSI circuits for hi rel/space applications. Such systems check the response of a circuit in presence of faults by comparing the behavior of the fault free circuit and the faulty one for a given set of stimuli or test patterns.

Besides radiation testing in e.g. particle accelerator facilities, SEU injection can also be performed by simulation [10] [11] [12]. Both methods are relatively slow due to either a limited upset rate, or a reduced device operation speed respectively.

FPGA emulation based fault injection has proven effective to reduce fault evaluation time in this context, because it allows a high upset rate as well as a high speed operation of the design under test. FPGA fault injection can be used to evaluate the SEU sensitivity of an ASIC design, injection is then performed in flip-flops or memories of the target design only, by either reconfiguring the device in order to present the faulty behavior [14] [15] or modifying the original circuit adding extra hardware to modify the state of the circuit [16] [17]. These approaches do not consider the FPGA as the target technology.

In case SRAM-FPGA themselves are employed in avionic or space applications, SEU injection must be done also in the device configuration memory and not in flip-flops only. As the configuration memory is not modeled in the netlist of the target design, logic simulation can not be used. But SEU fault injection by partial reconfiguration can be exploited to evaluate the degree of protection of mitigation techniques, or to get a rough evaluation of circuit behaviors prior to radiation testing. A configuration bitstream SEU emulator for SRAM-FPGA was described in [18]. The goal of the simulator was to provide a map of the sensitive bits of the device configuration memory. A prototypal system aimed at studying the SEU sensitivity of the reconfiguration logic of Xilinx Virtex devices was described in [19]. A similar system was used for radiation test activity for the same devices reported in [20].

3. FLIPPER

FLIPPER is a system that was mainly conceived to inject bit-flip faults within the internal memory of FPGA's. It consists of a hardware platform and a software application running on a PC. The hardware platform is actually a flexible FPGA-based general purpose board that can

be employed in other SEE evaluation activities (namely, ground radiation test), and also in more general applications as a testbed equipment or a digital I/O board.

The main FLIPPER board (Figure 1a) manages the overall fault injection by means of a Xilinx Virtex 2 Pro (XC2VP20) device. This control board contains 128 MByte of either SDRAM or DDR memory and 16 MByte Flash, and communicates to a PC via a USB 2.0 port controlled by a dedicated microcontroller. It also has two 240 pin connectors plus one 60 pin connector for the communication of test data and control data to/from a Device Under Test (DUT). Two further P160 standard connectors allow to use standard expansion boards with the Flipper control board.

The piggy-back style DUT board (Figure 1b) contains the Device Under Test. The DUT device used in the experiment described in this paper was a Xilinx Virtex 2 FPGA (XQR2V6000), but any Xilinx V2, V2 Pro, V4, and next generation devices can be used (with some limitations) for fault injection test. Up to 416 signals are driven by the main board's FPGA, corresponding to a maximum of 416 triplets (1240 DUT I/Os) in case of XTMR-ed designs. Virtually any component can be used as a DUT device for functional test and radiation test, provided that some constraints are met as far as pin number and device accessibility are met. The DUT board includes a temperature control chip to monitor the device temperature during the overall experiment.

(a) (b)

Figure 1: Flipper hardware platform. (a) Main board. (b) DUT board

Fault injection in FPGA devices is based on frame modification and active partial re-configuration. Both single bit and multiple bit upset can be injected. The injection experiment is run by means of a software application that was specifically developed for the FLIPPER system. Figure 2 shows a snapshot of the GUI interface for such application.

By means of this application it is possible to set up the experiment options, and in particular: *i)* the target of injection, *ii)* the test mode (internal memory bit may be randomly addressed and faults may accumulate until a functional failure occurs, or the bits may be addressed sequentially, one at a time), *iii)* the fault type (either single bit flip or multiple bit flip), *iv)* the DUT clock rate, and *v)* the address range of memory bits that are involved in the current experiment.

Within the application it is possible to import input and output values from a ModelSim simulation, at each clock edge. These data will be used as test vectors and reference (gold) values during the fault injection experiment. Test stimuli vectors may be up to 150 bit wide, and gold output vectors may be up to 120 bit wide. Test vectors and gold vectors are stored in the on board RAM, before the injection procedure begins, to allow a high speed functional test. The DUT is exercised with the whole set of test vectors anytime a bit flip is injected to verify the functional influence of such a fault on the device. Actual outputs from the DUT are compared on board with corresponding gold values to perform this functional verification.

When a mismatch is detected, a fault packet is sent to the PC including all the information relevant to the system behaviour. Once the test is run, it proceeds until the test stop condition (set by the software) is met.

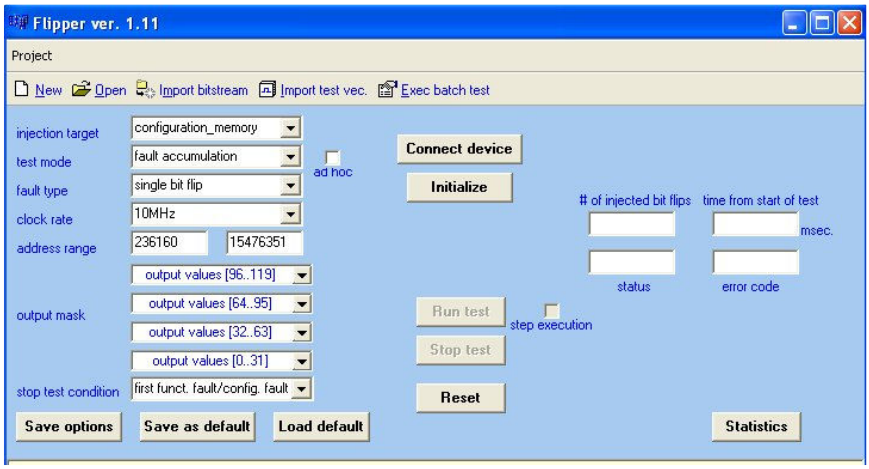


Figure 2: FLIPPER software application GUI

Compared to other solutions, the main assets of FLIPPER are:

- Piggy-back DUT board allowing an easy exchange of the DUT devices and hence keeping the test system up to date with the evolution of FPGA technology
- Memory based test pattern handling ensures a smooth flow to derive stimuli and expected outputs from a simulation testbench. The entire target FPGA can be used for the target design, it is neither necessary to duplicate the target design, nor to have stimuli generators implemented in hardware on the FPGA.

4. CUC-CTM

CUC-CTM stands for CCSDS Unsegmented Code and CCSDS Time Manager (CCSDS = Consultative Committee for Space Data Systems). The CUC-CTM is a synthesisable VHDL IP-CORE from ESA [21]. The CUC contains a frequency synthesiser and an elapsed time counter, and the CTM implements datation (time tagging) and alarm services. All services provided by the CTM are accessible via two AMBA APB slave interfaces, through which standard Spacecraft Time Source Packets according to the ESA Packet Telemetry Standard are available. There are several discrete output signals carrying the alarm or pulse-per-second (PPS), and input signals which allow time tagging of external events.

The netlist obtained by the XST synthesis tool from the CUC-CTM source code has been processed by the TMRTool from Xilinx. TMRTool offers different triplication options and output schemes. For this study the triplication has been applied to the entire design by using the XTMR "Standard" triplication. It means that all internal logic (combinatorial and sequential) is triplicated, and majority voters are inserted in all the state machine feedback paths. Two different output schemes have been implemented: triple and double minority voted output. They are quite similar, in both cases, the three redundant replicas (R1, R2, R3) of an output signal participate to the voting scheme, and the output signal that differs from the others is put in high impedance by a minority voter.

In the double voted output scheme only two out of three outputs of the replicas are brought to the device pad as shown in the right part of Figure 3.

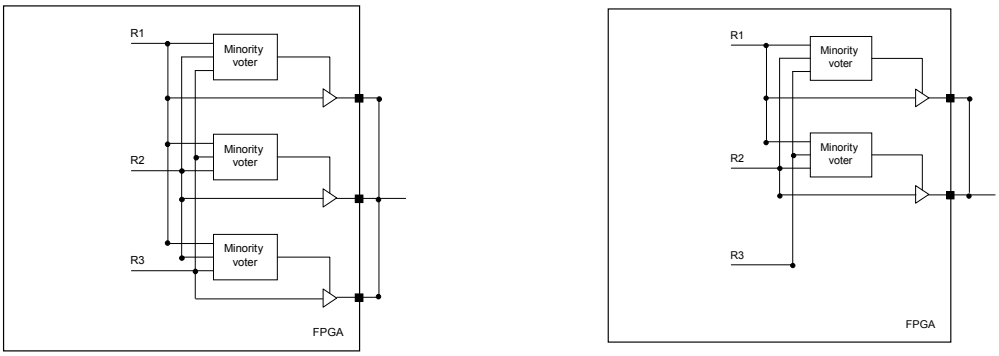


Figure 3: Triple and Double Voted Output Schemes

This allows saving one third of the output pads. Indeed, the drawback concerning the lower level of protection has to be assessed. The device resource usage is reported in Table 1. It has to be noticed that the double voted scheme, for the CUC-CTM, saves 73 out of 569 output pads.

	Plain	XTMR triple voted output	XTMR double voted output
FF	785 out of 67584	2361 out of 67584	2361 out of 67584
LUT	1789 out of 67584	7167 out of 67584	7094 out of 67584
IOB	212 out of 824	569 out of 824	496 out of 824
GCLK	1 out of 16	3 out of 16	3 out of 16

Table 1: XQR2V6000 resource usage

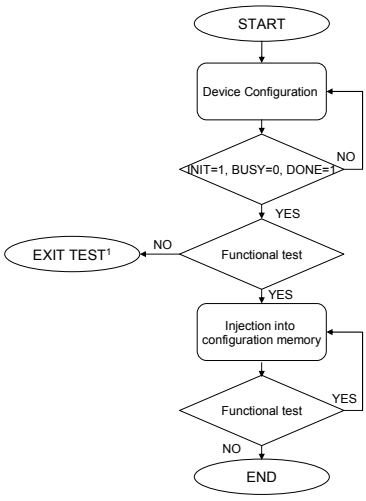
The VHDL test-bench provided by ESA together with the CUC-CTM source code, has been converted into stimuli for the fault injection campaigns. The test-bench is structured into several procedures, each of them verifies a specific function: after the reset verification, the access to both AMBA APB interface is tried, then the frequency synthesiser and time counter are configured. The verification of high level functions is performed afterwards. Among these functions there are time sample, time correlation, alarm and periodic pulse generation, interrupt, and time packet read out.

5. Experiment

A flow diagram of the experimental run is shown in Figure 4. A fault injection campaign is usually composed by many runs through the entire Figure 4 to obtain statistically relevant data. For each of the two XTMR design variants and for the plain design a fault injection campaign has been performed. In both cases the XQR2V6000 [22] is initially configured and, after checking the configuration signals (INIT, BUSY and DONE), the whole set of stimuli is applied for verifying the experimental set up and the correct design behaviour. FLIPPER then injects an SEU by active partial re-configuration into a randomly chosen configuration memory location and applies the stimuli. This procedure iterates, successively accumulating bit flips (SEU) in the configuration memory, until a functional fault is detected, i.e. the values on one or more output signals deviate from the expected outputs (golden pattern). The fault signature is logged to the workstation, the user can analyse how many and which bits of the configuration bitstream have been upset, and which outputs of the design have failed at which cycle of the stimuli.

In the plain version, 1000 injection runs were accomplished, for a total of about 3×10^5 SEU injected. For the triple voted output (TVO) and double voted output (DVO) versions the

injection runs were 2500 and 1000 corresponding to 2×10^6 and 1×10^6 SEU injected respectively. From the result file produced by FLIPPER a few graphs have been extracted, showing the probability of observing an output error as a function of accumulated SEUs. The distribution and the cumulative distribution of the number of injections to the first functional fault, for the plain and both design variants, are shown in Figure 5 and Figure 6 respectively. The class width in the figures is 25. The frequency is computed as the ratio between the numbers of accumulated SEUs to the output error in the class and the total number of injected SEUs. The cumulative distribution is the integral of the distribution (adding the actual frequency in a class to the frequencies of the previous ones); it indicates the probability of a functional failure to occur after a given number of configuration bit upsets. The mean value of the distribution of the number of injections to the first functional fault is 337 for the plain design, 1330 for the triple voted output version, and 1152 for the double voted output. The corresponding highest output error probabilities (8%-2,27%-2,87%) are after 25, 1000, and 775 injections respectively (Figure 5).



[1] A malfunction has occurred, and the test cannot proceed.

Figure 4: Experimental run flow diagram

A comparison of the three cumulative frequency distribution diagrams is given in Figure 6. It clearly shows the two design variants behave similarly (mach better than plain) and the triple voted output scheme gives only a little improvement. This is quite satisfactory as one third of the output pins could be saved in this design. Zooming the Figure 6 into the interval [0,50] results in the curves shown in Figure 7, where again the double voted version closely follows the triple voted one. As shown by the figure, the distribution details the error probability for a few injected SEUs.

Figure 7 provides the indication that up to 8 accumulated SEUs, no failure has been observed. This suggests the conclusion that even a scrubbing rate allowing up to 8 upsets may be enough to prevent malfunction.

Assuming a bit-error rate of 8000 bit-errors/day for this device (geostationary orbit, worst case solar activity [23]), this corresponds to a scrubbing rate of 1/1000 day or 1.44 min. However in a practical case, the statistical nature of the processes and the certainly limited coverage of test stimuli derived from a simulation test-bench will require a margin to be taken.

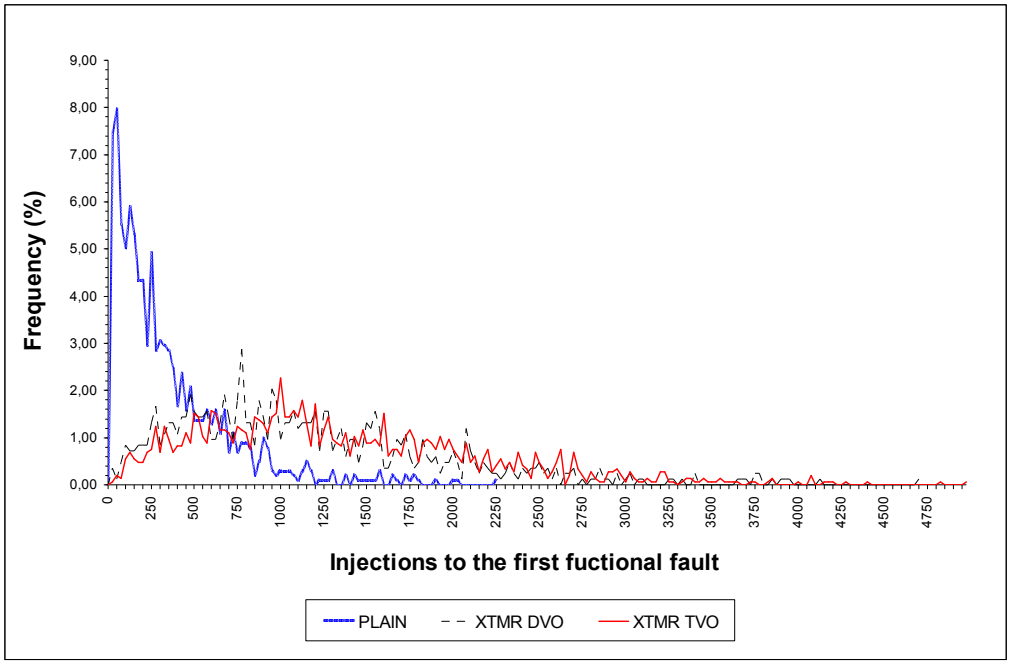


Figure 5: Distribution of the number of injections to the first functional fault

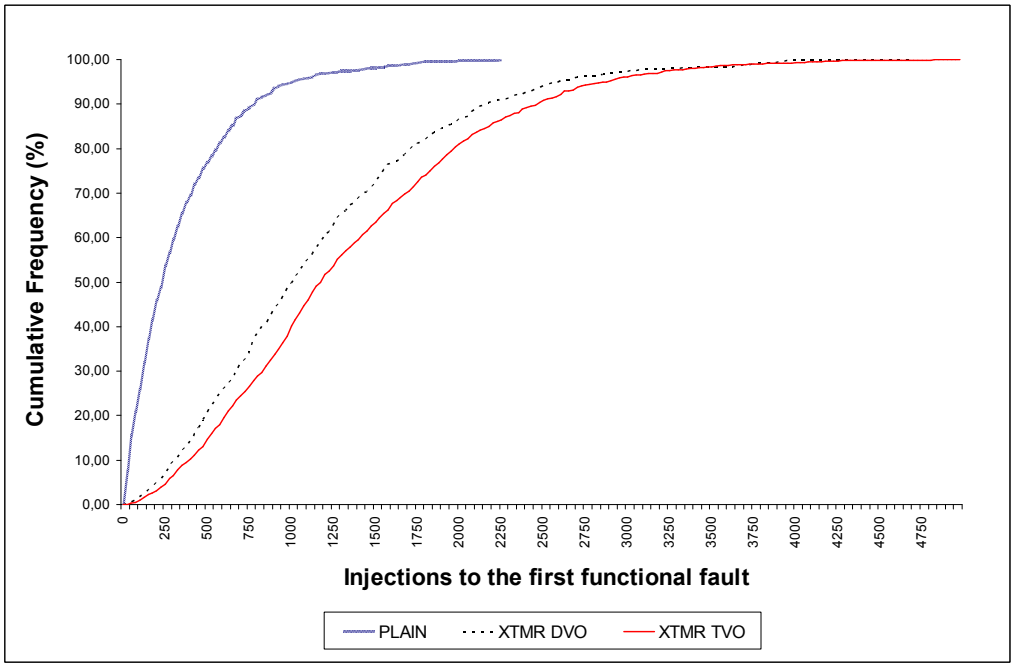


Figure 6: Cumulative distribution of the number of injections to the first functional fault

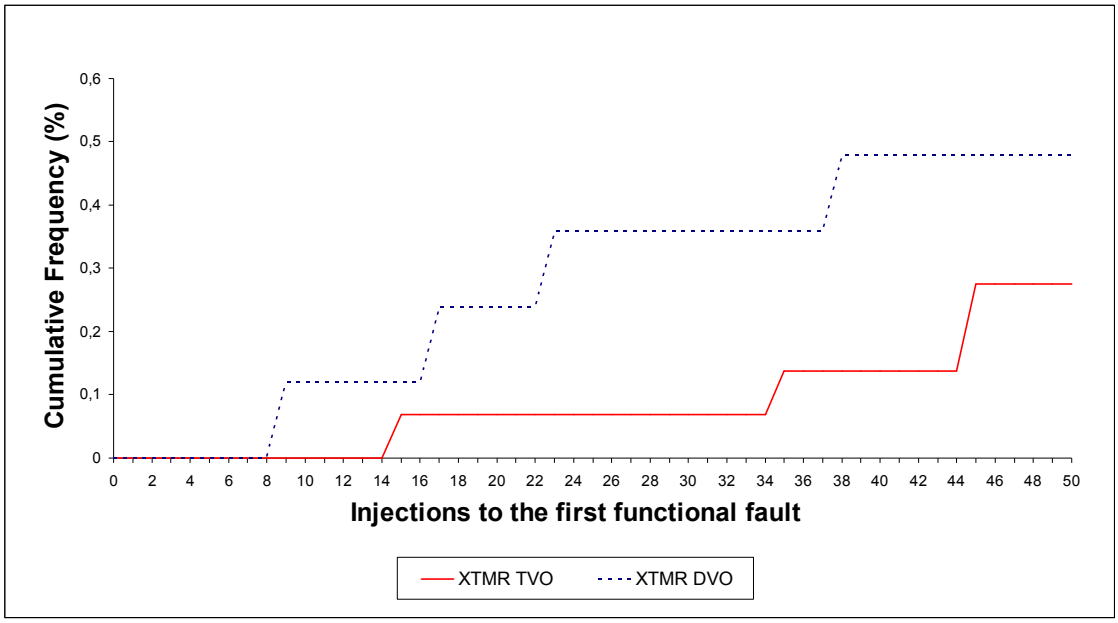


Figure 7: Cumulative frequency distribution (histograms class width = 1)

The experiment (3 mitigation schemes, 4000 runs and $3{,}3\times10^6$ injected faults) took a global execution time of 17 hours. This corresponds to an injection time of 18 ms/fault. Part of this time (50 μsec) is for partial reconfiguration, while the rest comprises the functional test execution (26000 testbench cycles), the generation of fault data packets and communication to the host. Compared to other reconfiguration based systems [17], FLIPPER shows a higher injection efficiency.

6. Conclusions

FLIPPER allows to quantitatively compare the level of protection offered by different mitigation techniques. In particular, results can help defining the scrubbing rate for a given circuit/design. Fault injection into a sample design from space application shows that the XTMR scheme considerably improves SEU fault tolerance. Also, the level of protection provided by a double minority voted scheme for the outputs of the design is only slightly lower than that offered by the triple voted scheme. In spite of that, one third of the output pins is saved.

7. References

[1] P.E. Dodd and L.W. Massengill, "Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics", *IEEE Trans. on Nucl. Sci*., vol. 50, n. 3, pp 583-602, June 2003.

[2] M. Caffrey, P. Graham, E. Johnson and M. Wirthlin,, "Single-Event Upsets in SRAM FPGAs", *in Proc. of the Military and Aerospace Applications of Programmable Devices Int'l Conference (MAPLD),* September 2002.

[3] C.C. Yui, G.M. Swift, C. Carmichael, R. Koga and J.S. George, "SEU Mitigation Testing of Xilinx Virtex II FPGAs", Radiation Effects Data Workshop Record, 2003.

[4] S. Rezgui, G.M. Swift, K. Somervill, J. George, C. Carmichael, and G. Allen, "Complex Upset Mitigation Applied to a Re-Configurable Embedded Processor", *IEEE Trans. on Nucl. Sci*., vol. 52, n. 6, pp 2468-2474, Dec. 2005.

[5] G.M. Swift, S. Rezgui, J. George, C. Carmichael, M. Napier, J. Maksymowicz, M. Moore, A. Lesea, R. Koga, T.F. and Wrobel, "Dynamic Testing of Xilinx Virtex-II Field Programmable Gate Array (FPGA) Input/Output Blocks (IOBs)", *IEEE Trans. on Nucl. Sci*., vol. 51, n. 6, pp 3469-3474, Dec. 2004.

[6] V. Pouget, A. Douin, D. Lewis, P. Fouillat, G. Foucard, P. Peronnard, V. Maingot, J.B. Ferron, L. Anghel, R. Leveugle and R. Velazco, "Tools and Methodology Development for Pulsed Laser Fault Injection in SRAM-Based FPGAs", *8th Latin American Test Workshop (LATW 2007)*, [Cusco (Peru), 11- 14 Mars 2007].

[7] L. Sterpone and M. Violante, "A new analytical approach to estimate the effects of SEUs in TMR architectures implemented through SRAM-based FPGA's", *IEEE Trans. Nucl. Sci*., vol. 52, no. 6, pp. 2217–2223, Dec. 2005.

[8] FLIPPER product sheet, http://microelectronics.esa.int/techno/Flipper_ProductSheet.pdf.

[9] TMR Tool User Guide, UG156 (v1.0), 30 Sept. 2004, *Xilinx*.

[10] Daniel González Gutiérrez, "Single Event Upsets Simulation Tool Functional Description", ESA ESTEC, July 2004 http://microelectronics.esa.int/asic/SST-FunctionalDescription1-3.pdf

[11] H. Cha, E. M. Rudnick, J. Patel, R. K. Iyer, G.S.Choi, "A gate-level simulation environment for alpha-particle-induced transient faults", *IEEE Trans. Comput*., vol. 45, pp. 1248–1256, Nov. 1996.

[12] M. Rebaudengo, M. Sonza Reorda, M. Violante, B. Nicolescu, R. Velazco, " Coping With SEUs/SETs in Microprocessors by Means of Low-Cost Solutions: A Comparison Study", *IEEE Trans. on Nucl. Sci*., Vol. 49, No. 3, June 2002

[13] C. Carmichael, "Correcting Single-Event Upsets Through Virtex Partial Configuration", XAPP216 (v1.0) June 1, 2000, *Xilinx*.

[14] L. Antoni, R. Leveugle, B. Fehér, "Using Run-Time Reconfiguration for Fault Injection in Hardware Prototypes", in *Proc. of the 2000 Int'l Symposium on Defect and Fault Tolerance in VLSI Systems*, [Monte Yamanashi, October 25-27, 2000], pp. 405-413.

[15] M. Aguirre, J.N. Tombs, F. Muñoz, V. Baena, A. Torralba, A. Fernández-León, F. Tortosa, and D. González-Gutiérrez, "An FPGA based hardware emulator for the insertion and analysis of Single Event Upsets in VLSI Designs", in *Proc. Radiation Effects on Components and Systems Conf. (RADECS),* Madrid, Spain, Sept. 2004.

[16] P. Civera, L. Macchiarulo, M. Rebaudengo, M. Sonza Reorda, and M. Violante, "Exploiting Circuit Emulation for Fast Hardness Evaluation", in *IEEE Trans. on Nucl. Sci*., vol. 48, n. 6, pp 2210-2216, Dec. 2001

[17] C. López-Ongil, M. García-Valderas, M. Portela-García, and L. Entrena, "Autonomous Fault Emulation: A New FPGA-Based Acceleration System for Hardness Evaluation", *IEEE Trans. on Nucl. Sci*., vol. 54, n. 1, pp 252-261, Feb. 2007.

[18] E. Johnson, M. Caffrey, P. Graham and M. Rollins, "Accelerator Validation of an FPGA SEU Simulator", *IEEE Trans. on Nucl. Sci*., vol. 50, n. 6, pp 2147-2157, Dec. 2003.

[19] M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, A. Marmo, S. Pastore, and G.R. Sechi, "A Tool for Injecting SEU-like Faults into the Configuration Control Mechanism of Xilinx Virtex FPGAs", in *Proc. of the 2003 Int'l Symposium on Defect and Fault Tolerance in VLSI Systems*, [Cambridge, November 3 - 5, 2003, USA], pp. 71-78.

[20] M. Alderighi, A. Candelori, F. Casini, S. D'Angelo, M. Mancini, A. Paccagnella, S. Pastore and G.R. Sechi, "SEU Sensitivity of Virtex Configuration Logic", in *IEEE Trans. on Nucl. Sci*., vol. 52, n. 6, pp 2462-2467, Dec. 2005.

[21] CCSDS Unsegmented Code (CUC) & CCSDS Time Manager (CTM) Synthesizable VHDL Cores Data Sheet, Dec. 2003*, ESA ESTEC*. http://microelectronics.esa.int/core/corepage.html

[22] Virtex-II Platform FPGA User Guide, UG002 (v2.0) 23 March 2005, Xilinx.

[23] C. Carmichael, B. Bridgford, G. Swift, and M. Napier; "A Triple Module Redundancy Scheme for SEU Mitigation of Static Latch-Based FPGAs", *Military Applications of Programmable Logic Devices (MAPLD) Conference 2004,* available at http://www.klabs.org/mapld04/presentations/session_l/5_l189_carmichael_s.ppt.