Vaskova, A., Lopez-Ongil, C., Portela-Garcia, M., Garcia-Valderas, M. & Entrena, L. (2013, agosto). SEU Sensitivity Comparison for Different Reprogrammable Technologies With Minority Check Block. IEEE Transactions on Nuclear Science, 60(4), 2813-2818.

DOI: 10.1109/tns.2013.2245343

SEU Sensitivity Comparison for Different Reprogrammable Technologies with Minority Check Block

A. Vaskova, C. López-Ongil, M. Portela-García, M. García-Valderas, L. Entrena

***Abstract*.- In this work, a method is proposed for obtaining comparable measurements of the SEU sensitivity in reprogrammable devices that present different characteristics like internal architecture, technology, amount of available resources, etc. A specific minority checker is developed for reporting the presence of SEUs or MBUs which will help in this comparing task during dynamic tests.**

***Index Terms*—Cross section, SEU, MBU, Reprogrammable devices, Fault Injection Campaigns.**

I. Introduction[1]

THE use of reprogrammable devices (FPGAs and CPLDs[2]) in digital systems has been extended greatly in recent years. Not only automotive applications [1] but also space on-board equipment [2] are including more devices that can be reconfigured and that are adding a high degree of flexibility to the whole system, while performance and power consumption are still competitive.

Currently, digital systems are suffering faults from ionizing radiations and neutrons. For example, for satellites in LEO orbit (850 km, 90º) the expected number of proton particles per square cm, per day is around 1.58E09 as stated in [3]. While at sea level, the number of neutron particles per cm^2h^{-1} (>10MeV, NYC reference) is around 13 [4]. That means almost 19,000 particles will hit a $1cm^2$ system, every second in a LEO orbit and every 2 months at sea level[3].

On the other hand, radiation hardness of reprogrammable commercial devices is far from being high [5]-[7]. Considering *soft errors* caused by ionizing particles or terrestrial neutron radiation on memory elements (SEUs/SBUs/MBUs[4]), fault tolerance must be assured in the configuration memory (SRAM, Flash or EEPROM) and in the user memory (design *flip-flops* and ROM/RAM embedded modules).

Enhancing and measuring the dependability of these devices implies a bigger problem than for ASICs and COTS. There are many studies and proposals for reducing radiation sensitivity of these devices [6]-[8]. In general, redundancy and *scrubbing* are the best solutions for hardening reprogrammable devices working in safety critical applications.

Aerospace and automotive designers face also with a key problem. Selecting a reprogrammable device implies the difficulty of comparing the robustness of different manufacturers and device families available in the market. Traditional COTS and ASIC are qualified according to standards and their technology is labeled uniquely; the hardware is not to be changed. In the case of reprogrammable commercial devices, this label is not available, and designers must evaluate the circuit robustness in terms of configuration memory (technology) and the implemented design (user memory).

In this work, a method for comparing different reprogrammable devices with respect to their SEU sensitivity is proposed; taking into account these devices provide different configuration technologies, internal architectures for design implementation and amounts of available resources. This method will help system designers to obtain quantitative measurements in order to compare the sensitivity of different reconfigurable devices and, therefore, to choose the best solution for their application.

The paper is organized as follows. Section II summarizes the characteristics of different FPGA technologies and exposes the existing approaches used to measure the sensitivity under SEUs, in case of a programmable device. Section III explains in detail the method proposed to compare the sensitivity for different programmable devices. Section IV describes the developed experiments and analyses the obtained results. Finally, section V states the conclusions of this work.

II. Technology Qualification for Reprogrammable Logic Devices

A. Programmable Logic Devices

Basically, there are three types of FPGAs/CPLDs depending on the programming technology used: SRAM, Flash, or antifuse-based. Each one presents different characteristics and also, for a same type of programmable devices, vendors provide different device families with different architectures, performance, technology, reliability features, etc.

SRAM-based FPGAs present the highest densities providing a great capacity, allowing the implementation of

Manuscript received April 16, 2012.

Electronic Technology Department, Carlos III University of Madrid, Spain.(avaskova, celia, mportela, mgvalder, entrena@ing.uc3m.es)

[1] This work is partially supported in part by the Spanish Ministry of Science and Technology, code TEC2010-22095-C03-03.RENASER+ project

[2] *Field Programmable Gate Array (FPGA), Complex Programmable Logic Device (CPLD)*

[3] These data depend on solar activity while the effect of the particles depends on their energy.

[4] *Single Event Upset (SEU), Single Bit Upset (SBU) and Multiple Bit Upset (MBU)*

very complex circuits (up to several millions of logic cells), and a very high performance. These devices are reprogrammable and volatile, what usually involves the use of an external non-volatile memory to store the configuration data. These devices are sensitive to SEUs. An SEU in the configuration memory may cause critical faults such as an SEFI (Single Event Function Interrupt), requiring the reconfiguration of the FPGA to correct the fault.

Flash-based FPGAs present less capacity and performance than SRAM-based devices but the power consumption is lower and the memory configuration is fairly robust to SEUs. Flash gates are much less sensitive to soft errors than SRAM cells on a per bit basis [9]. These devices are reprogrammable and non-volatile.

Antifuse-based FPGAs are the most robust kind of the programmable logic devices but they are not reprogrammable.

CPLDs with a simpler internal architecture than FPGAs are aimed to prototype simpler circuits. These devices are reprogrammable and non-volatile. Their programming technology is based on floating-gate transistor, in an EEPROM structure.

The selection of the most suitable programmable logic device depends on the requirements of the given application. When the robustness with respect to SEUs is one of the requirements, designers must select the most robust device among those available in the market. SEU sensitivity must be compared for a selection of various devices in different manufacturers and/or families. When devices present different features in terms of architecture, technology or resources, obtaining comparable measurements is mandatory to state a precise conclusion.

B. Radiation Effects Sensitivity Testing

In order to measure the SEU sensitivity of a device technology, irradiation ground tests are necessary (proton, neutron, heavy-ion, alpha particles, etc.). The sensitivity of a circuit depends on two factors: technology sensitivity and functionality, since due to the workload some events can be masked or have no effect in the circuit behavior. An accepted method to quantify the SEU sensitivity of a circuit was proposed in [10]. It consists in calculating the SEU sensitivity as a combination of static and dynamic test results in terms of cross section, σ. For a given workload *W* the SEU sensitivity of a circuit can be quantified with the following equation [10]:

$$\sigma(w) = \sigma \cdot \tau$$

The technology factor is quantized by means of an irradiation ground test, where σ is the relation between the number of observed faults (N_F) and the fluence (Φ, number of particles per area unit), $\sigma = N_F/\Phi$. Being τ the error rate due to the workload, it can be measured by other fault injection methods.

In the case of a reprogrammable device, a SEU can affect the configuration memory and the user logic. These two effects imply very different consequences: a firm error and soft error respectively. For this kind of devices, the method to measure the SEU sensitivity implies some particularities. The static test is performed by reading the configuration memory to check errors without taking into account the effect in user logic. The number of the configuration memory bits is much higher than those related to user logic. However, some reprogrammable devices use different technologies for configuration memory and user logic, and an in-depth analysis requires studying both of them. Static test is not suitable for checking sensitivity in user memory, because test structures for this purpose (i.e. scan-path) could be modify when a fault affects the configuration memory.

On the other hand, the dynamic test consists in irradiating the reprogrammable device while it is working, and observing the circuit outputs. Misbehaviors are due to faults in user logic as well as in configuration memory. Thus, dynamic test is necessary to study the sensitivity of a programmable device in a complete way. This dynamic test should be done with a design allowing complete fault propagation to the device outputs. This would be the worst case, because any other design will mask more faults.

There are some works presented in the literature dealing with test of reprogrammable devices [11]-[14] . Generally, these works are presenting a comparison between two, or more, different devices from different manufacturers, with respect to a precise irradiation ground test (heavy ions in [11], heavy ion and Co_{60} in [12], protons in [13] and alpha particles and neutrons in [14]). Some of them only present results from static tests. The dynamic test is addressed in [13] and [14]. In [13], dynamic tests are performed by using the same circuit, without taking into account the size of each device; therefore, in the largest devices, some of the errors could remain undetected since part of the logic is unused. In [14], a report on soft error rate tests for three different FPGA vendors is presented. Multiple replicas of the same circuit are implemented in the bigger devices and a multiplexer is proposed, sequentially addressed from outside, as the block in charge of producing the unique output of the N-replicas. Although not specified, reading frequency is supposed to be N-times faster than input stimuli application, in order to check all the multiplier instances for every different input stimulus. It can be very difficult to achieve this condition for circuits working at high frequency. With the approach presented in this work, that constrain is solved.

In summary, the existing comparative studies do not apply or proposed a general method, and the works that present a method imply some limitations. In this work, a general method to compare SEU sensitivities of different programmable devices is proposed.

III. PROPOSED METHOD

When comparing different reprogrammable devices with respect to their SEU sensitivity, two aspects has to be taken into account during dynamic tests:

- The differences in resources availability (due to different technologies and/or to different architectures) will be combined with the maximum occupation of devices tested.
- The differences in performance could be solved taking the worst case; so as operating frequency will be the

same in all devices under test.

In order to compare the SEU sensitivity of two devices (X and Y) from different manufacturers or with different architectures/technologies, the integration capacity must be taken into account. Suppose that available resources are less in X than in Y, and a design is prototyped in X occupying the whole device.

If the same design, CUT (Circuit under Test), is prototyped in device Y, a large amount of resources will be neither configured nor tested, during the dynamic irradiation ground test campaign. As shown in Figure 1, replicating the X-design in device Y seems a good solution, but some aspects must be considered:

- The number of device outputs is fixed (and finite). It will not be possible to observe the outputs of every replica during test campaign. A specific block will receive outputs of every replica and produce a unique output. This global output will be erroneous when any replica produces an erroneous output. In our proposal, this block is a *minority checker*.
- As every replica receives the same inputs, mapping tasks should be tightly driven to avoid redundant replicas elimination.

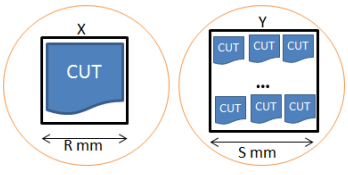

Figure 1. The two devices to compare in the test irradiation campaign

We need to prototype both devices with two designs which propagate every fault to the circuit outputs, and behave in the same way. In such a way, irradiation dynamic tests results are due only to technology sensitivity, because we are comparing the same circuits.

With our method, assuming SEU sensitivities of device X and of device Y are in the same order, an irradiation ground test campaign with the same fluence would produce N_F errors in device X and N'_F errors in device Y, but the cross section per bit should be equal.

In general, to compare the SEU sensitivity of two reprogrammable devices, an approximated measurement of the ratio between their sensitivities (S_X/S_Y) is:

$$\frac{S_X}{S_Y} = \frac{\sigma_X}{\frac{\sigma_Y}{NCUT}} = \frac{\sigma_X}{\sigma_Y} \cdot NCUT$$

Where σ is the cross section of the corresponding circuit and NCUT is the number of replicas implemented in the biggest device.

A. Circuit Under Test: StageMult

The reprogrammable device under test should prototype a design with a high observability, so that, an error in user logic can be observed in the outputs. Additionally, the design prototyped should occupy the maximum number of logic resources in order to obtain a complete measurement of the device robustness. The selection of the design to be prototyped and tested *CUT* is an important stage in the system engineering, as already stated in [12].

According to previous considerations, the test circuit proposed consists in a 10-bit multiplier. This is a proof-of-concept. Other designs are also interesting for this method.

The multiplier is dimensioned in order to maximize the use of the available resources in device X. Due to its pipelined structure this circuit provides a high observability. The probability of SEUs in the configuration memory is larger than in the registers. Therefore, the circuit under test has been designed with a higher amount of combinational logic in comparison with the memory elements (flip-flops). Registers are at the inputs, outputs and in the pipeline (one stage).

B. Minority Checker Block

As mentioned above, a specific block (*minority checker*) is designed to report the presence of errors in any replica implemented in device Y, Figure 2.

In device X the CUT is *StageMult*, while in device Y the CUT is *StageMultN*, containing N replicas of *StageMult*, plus a minority checker block. This block compares the outputs of the N replicas. Taking the *bit-j*, if there are some different bits, the block generates an equivalent *output-j* which represents this difference (the less repeated value). Minority detection is performed counting the number of '1s' and '0s' of every *j-bit*. Every single error is transmitted by this block. Also, multiple errors implying the minority of N bits are also propagated to the outputs, which are sent outside the device under test. Externally error checking is performed continuously during the irradiation ground test campaign to calculate the *Soft Error Rate*.

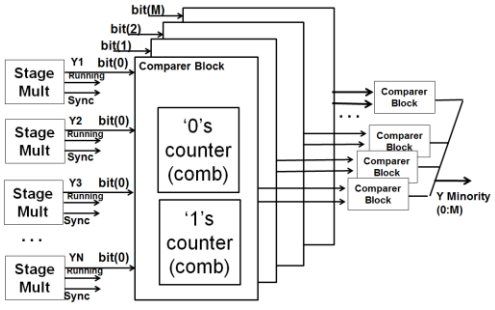

Figure 2. StageMultN: N replication of StageMult, with minority checker

IV. Experimental Results

A. Device technologies to compare

In this work, three devices of CoolRunnerII™ from Xilinx® and one device of Igloo® from Actel® have been chosen to be compared. These devices are under consideration for their use in OPTOS *cubesat* developed by INTA [15].

In the following, the main characteristics of these device technologies are summarized. Table 1 presents main characteristics of devices from Xilinx and Actel.

1) Cool Runner II from Xilinx

The CPLD CoolRunner-II (CRII) devices from Xilinx are not designed for functioning in high radiation environments, but their special features make them greatly suitable for space use. Their ultra-low power consumption (28.8 µW) together with small factor packages make CRII a valuable weapon to deal with self-powering batteries systems and multiple redundant sensors. CRII CPLDs use the standard 180-nm CMOS technology to create the CPLD architecture and to provide very low power consumption [16].

This technology implements two configuration memories that enable *On-the-fly* reconfiguration and, therefore, small reconfiguration times. A non-volatile memory (Flash) stores configuration that can be downloaded onto volatile configuration memory (SRAM) every time the device is powered up or restarted. Devices of 32, 256 and 512 macrocells have been chosen to be compared.

2) Igloo from Actel

The Actel® Igloo® series of low power FPGA is highly suitable for space purposes. Igloo is a higher density FPGA with extra pin capability and low power modes, which makes it a valuable target for small satellite platforms with low power budgets and high computational requirements. Igloo devices are reprogrammable and full-featured Flash technology. Flash-based FPGAs require lower power consumption than SRAM-based FPGAs and besides Igloo series provide a low power static mode, called *Flash*Freeze* mode, which retains all SRAM and register information consuming 160 µW and can still return to normal operation in 1µs. A 130-nm LVCMOS manufacturing process with seven layers of metal is used. Standard CMOS design techniques are applied to implement logic and control functions [17].

	CRII CPLD, Xilinx			**Igloo FPGA Actel**
Device	**XC2C32 QFG32**	**XC2C256 7TQ144**	**XC2C512 7PQG208**	**M1AGL1000 FGG484**
CMOS (nm)	180	180	180	130
Length x Width (mm)	5x5	20X20	28x28	23x23
# of Macrocells	32	256	512	24,576
# bits in *StageMult*	4	10	4	10
# of instances of *StageMult*	1	1	8	25

Table 1. Characteristics of compared devices from Xilinx and Actel

B. Design implementation

First, in order to evaluate the effectiveness of the *minority checker* block, two devices from the same manufacturer are compared: CRII-32 and CRII-512. A 4-bit *StageMult* multiplier was prototyped on a CRII-32 with a total amount of 85% of area occupation. On the other hand, when prototyping this multiplier on a CRII-512, only a 6% of area was occupied, Table 2. Therefore, a total number of 8 units of 4-bit *StageMult* and a *minority checker* block could be prototyped on a CRII-512 resulting in an 89% of area occupation, Table 3. In that case, minority checker requires the 26% of those combinational resources.

Device	**Macrocells** Used / Total	**Product Terms** Used / Total	**Registers** Used / Total
CRII-XC2C32 *SM*(4-bit)	27/32 (85%)	86/112 (77%)	24/32 (75%)
CRII XC2C512 *SM* (4-bit)	27/512 (6%)	86/1,792 (5%)	24/512 (5%)

Table 2. Results of area occupation when implementing a 4-bit StageMult in different devices from Xilinx.

Device	**Macrocells** Used/ Total	***Total***		***Minority Checker***	
		Product Terms Used / Total	**Registers** Used / Total	**Product Terms** Used / Total	**Registers** Used / Total
CRII XC2C512 *SM*N (4-bit) N=8 units	453/512 (89%)	1,326/ 1,792 (74%)	188/512 (37%)	465/ 1,792 (26%)	0/512 (0%)

Table 3. Area occupation of StageMultN &Minority Checker in the Xilinx largest *CoolRunner-II* device

Secondly, considering the possibility of comparing Xilinx technology with Actel technology, selected devices from these manufacturers must implement equivalent CUTs, although internal architectures were fairly different. This is the general case: the aim is to compare two devices from different manufacturers and technologies.

From Xilinx, a *CoolRunner* CR-II-256 device was selected, while from Actel a *Igloo* M1AGL1000 was chosen. CRII-256 device is smaller than Igloo device. A 10-bit *StageMult (SM)* multiplier was prototyped on the CR-II-256 device with a 77% of area occupation, while this same multiplier took the 3% of area occupation in the M1AGL1000 device, Table 4.

Device	**Macrocells** Used / Total	**Product Terms** Used / Total	**Registers** Used / Total
CRII-XC2C256 *SM*(10-bit)	196/256 (77%)	805 /896 (90%)	109 /256 (42%)
Igloo M1AGL 1000 *SM*(10-bit)	667/24,576 (3%)	585 / 24,576 (2%)	82 / 24,576 (0.3%)

Table 4. Results of area occupation when implementing a 10-bit *StageMult* in devices from Xilinx and Actel.

Device	**Macrocells** Used/ Total	***Total***		***Minority Checker***	
		Product Terms Used / Total	**Registers** Used / Total	**Product Terms** Used / Total	**Registers** Used / Total
Igloo M1AGL 1000 *SM*N(10-bit) N=25 units	22,410 / 24,576 (91%)	20,488 / 24,576 (83%)	2,300 / 24,576 (9%)	4,180/ 24,576 (17%)	0/24,576 (0%)

Table 5. Area occupation of StageMultN &Minority Checker in the Igloo device

As shown in Table 4, the area occupation for a 10-bit multiplier *StageMult* implementation in CRII-XC2C256 and M1AGL1000 devices is very different. A total amount of 90% combinational logic resources are used while the 42% of memory elements are employed in CRII-XC2C256. This matches with the primary goal of looking for SEU effects in

the configuration memory as well as in user memory, and for their propagation to the circuit outputs. If the same circuit is prototyped in an Igloo FPGA, less than the 3% of available combinational resources are required. Taking N = 25, the Igloo FPGA is filled up to 91%, Table 5. In that case, minority checker requires the 17% of those combinational resources. This block is totally combinational, any SEU affecting this block would be due to a fault in the configuration memory. Therefore, the block functionality would change provoking erroneous outputs. Besides, fault masking in this block is not probable.

C. Fault Injection Method: Autonomous Emulation

A SEU sensitivity analysis has been performed with four intensive fault injection campaigns through *hardware-based emulation*. Two designs have been tested twice: *StageMult* and *StageMultN,* first in Xilinx devices and, secondly, in Xilinx and Actel devices. The objective of these experiments is to study the obtained fault classification in order to use these data for analyzing further results from the irradiation ground test campaigns. Performed experiments are also aimed to check the effectiveness of the *minority checker* block with respect to error propagation in different circuits, when many instances of the main CUT are prototyped.

A complete set of single bit-flips is considered, every memory element receives a SEU in every clock cycle of the workload[5]. The fault injection campaign has been performed with Autonomous Emulation platform [18], a fault injection technique based on hardware emulation.

Autonomous Emulation has been proved as an efficient tool for sensitivity evaluation of single circuits. It is focused on the analysis of circuits' behaviour with and without faults injected in their critical elements. The comparison between both behaviours allows the classification of the fault effect. The tool considers circuit's outputs and internal memory elements for fault classification. So, when the effect of a fault is propagated to the outputs, fault classification is set to Failure or Detected, depending on the type of output. Furthermore, this tool is also able to determine if a fault effect has completely disappeared within the circuit (Silent fault) or if it is still remaining in some memory elements of the circuit (Latent fault).

The fault injection system has been prototyped on an XC5VLX110T Xilinx Evaluation Platform. The use of an FPGA for hardware emulation is not relevant for the experimental results.

In *Autonomous emulation*, the injected faults are classified according to the effect produced on the circuit behavior. The categories considered are the following. When the effect of a fault is propagated to the outputs, fault classification is set to ***Failure***. When the fault effect has completely disappeared within the circuit, fault is classified as ***Silent fault*** (no error). Finally, when the fault effect remains in some memory elements of the circuit, it is classified as ***Latent.***

[5] In this case, a complete workload has been applied: $2^{10}x2^{10}$ test vectors for the 10-bit multiplier and 2^4x2^4 for the 4-bit multiplier.

First, single fault injection campaigns were performed. Secondly multiple errors caused in different bits of the multiplier were analyzed.

D. Fault Injection Campaigns. Single Event Upsets

In 1 unit of 10-bit *StageMult* a single fault has been injected in every circuit flip-flop (109 flip-flops) and in every clock cycle of the workload (1,572,954 clock cycles). The total number of injected faults is **171,451,986**. In Table 6 fault classification for each case is reported. As a global result, 31.15% of faults are silent and 0 % is latent. Failure rate is 68.85%.

In 25 units of 10-bit *StageMult* a single fault has been injected in every circuit flip-flop (2,300 flip-flops) and in every clock cycle of the workload (1,572,954 clock cycles). The total number of injected faults is **3,617,794,200**. Results are equivalent to those obtained for one replica of *StageMult*, as expected. This fact allows us to conclude that errors affecting the circuit will arise at the circuit outputs with high probability.

	StageMult	*StageMult*
	1 unit of 10- bit	*1 unit of 4-bit*
	#Faults %	#Faults %
Silent	53,406,088 **31.15%**	3,182 **28.51%**
Failure	118,045,811 **68.85%**	7,964 **71.36%**
Latent	87 **0.00%**	14 **0.13%**
Total	**171,451,986** **100.00%**	**11,160** **100.00%**

Table 6. Fault classification for single fault injection campaign.

The column 3 of Table 6 shows the obtained fault classification for 4-bit *StageMult*. A single fault has been injected in 24 flip-flops and in every clock cycle of the workload (465 clock cycles). The total number of injected faults is **11,160**. Failure rate is 71.36%.

Furthermore, in 8 units of 4-bit StageMult, a single fault has been injected in 188 flip-flops and in every clock cycle of the workload (465 clock cycles). The total number of injected faults is **87,420**. Results are equivalent to those obtained for one replica of *StageMult*, as expected.

As already reported in the literature, different technologies present different SEU sensitivities. For example, flash-based FPGAs are much robust than SRAM-based FPGAs, [9]. Thanks to the method proposed in this paper, these differences will arise in a dynamic irradiation test campaign.

E. Fault Injection Campaigns. Multiple Event Upsets

A subset of double faults has been injected in devices with 10-bit multipliers, Table 7. These double faults have been injected in flip-flop pairs in every clock cycle of the workload. The flip-flop pairs have been chosen according to circuit netlist adjacencies.

The total amount of faults is reported in line 6. The occurrence of double faults provokes the reduction of silent faults. In both circuits, the effect of these faults is similar. The difference is due to the fact that in case of *StageMultN* faults

can affect different replicas, what is equivalent to a single fault in *StageMult*.

	StageMult		*StageMultN*	
	#Faults	%	#Faults	%
Silent	42,989,215	**25.07%**	1,106,132,184	**30.57%**
Failure	128,462,687	**74.93%**	2,511,659,966	**69.43%**
Latent	84	**0.00%**	2,050	**0.00%**
Total	**171,451,986**	**100.00%**	**3,617,794,200**	**100.00%**

Table 7. Fault classification for double fault injection campaign.

V. CONCLUSIONS

In this work a method for obtaining comparable results in fault injection campaigns on reprogrammable devices is proposed. The method implies the prototyping of a fully observable circuit in the different devices to be tested, including the required circuit replicas for complete area occupation (for the larger devices) and a specific *minority checker* to propagate any error to the outputs of the device.

As stated in section III, prototyping the same design in programmable devices with different sizes, will imply many undetected SEUs. Therefore, a complete occupation enhances the irradiation ground test campaigns. On the other hand, maintaining the experiment setup for all devices under test will enlighten the campaign. In this sense, authors present a minority checker, for larger devices with several design replicas, which produces the same outputs as a single design, in smaller devices.

Experimental results on fault injection campaigns have been developed. Similar fault classifications are achieved when prototyping equivalent circuits in different devices, thanks to a *minority detector* at the outputs of every replica. These data will be useful for analyzing further results from the irradiation ground test campaigns.

This method will be used for qualifying programmable devices for space applications, Igloo® and CoolRunner-II™ in OPTOS satellite, and for applications dealing with collaborative hardening and distributed functionality, (RENASER+ Spanish research project[6]).

ACKNOWLEDGEMENTS

Authors would like to thank Alberto Martín-Ortega, Santiago Rodríguez, José Ramón de Mingo and Héctor Guerrero, from National Institute of Aerospace Techniques (INTA), Madrid, Spain, for their offer to participate in the qualification tests of reprogrammable devices for aerospace applications.

REFERENCES

[1] "TT CAN IP Module", Automotive Electronics. Product Information. Bosch. March 2012. (Bosch product info ttcan ip pdf)
[2] "A System-on-a-Chip for Small Satellite Data Processing and Control" ("ChipSat"), from ESA Web page http://microelectronics.esa.int/soc/
[3] W.K.Stuckey et al. "Solar Ultaviolet and Space Radiation Effects on Inflatable Materials" Space and Missile Sytems Center. 2000(8565-9)
[4] JESD89A: Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices Oct 2006 http://www.jedec.org
[5] "Single Event Effects in FPGAs" Actel report.
[6] M. Sonza Reorda et al. "Multiple errors produced by single upsets in FPGA configuration memory: a possible solution" European Test Symposium 2005. pp 136-141.
[7] Pratt et al. "Improving FPGA Design Robustness with Partial TMR". Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International. pp. 226-232
[8] S. Srinivasan et al. "Improving Soft-error Tolerance of FPGA Configuration Bits" International Conference on Computer Aided Design, 2004. pp.107-110.
[9] S. GerarDin, et al. "Scaling Trends of Neutron Effects in MLC NAND Flash Memories", IEEE International Reliability Physics Symposium (IRPS), pp. 400-406, May 2010.
[10] S. Rezgui, R. Velazco, R. Ecoffet, S. Rodriguez, J. R. Mingo "Estimating error rates in processor-based architectures" IEEE T. on Nuclear Science, Vol. 48, Issue 5, pp. 1680-1687, October, 2001.
[11] G.M. Swift, R. Roosta. "Tradeoffs in Flight-Design Upset Mitigation in State-of-the-Art FPGAs: Hardened by Design vs Design Level Hardening". On-line source (hdl:2014/41057) NASA Technical Reports Server. MAPLD 2004.
[12] M. Berg. "Signle Event Effects in Field Programmable Gate Array (FPGA) Devices: Update 2012" NEPP Electronic Technology Workshop, June 2012
[13] D. M. Hiemstra, G. Battiston, P. Gill, "Single Event Upset Characterization of the Virtex-5 Field Programmable Gate Array Using Proton Irradiation", IEEE Radiation Effects Data Workshop (REDW), pp. 4, 2010.
[14] iROC Technologies "Radiation Results of SER Test of Actel, Xilinx and Altera FPGA instances". GRE 2 ACTEL SERTEST DEC 03. Eng. TR008. April 2004.
[15] I. Lora et al. "INTA PicoSatellite OPTOS: Mission, Subsystems, and Payload". 4th Annual CubeSat Developers' Session III, Workshop. 2007.
[16] www.xilinx.com
[17] www.actel.com
[18] C. López-Ongil et al."Autonomous Fault Emulation: A New FPGA-based Acceleration System for Hardeness Evaluation". IEEE T. on Nuclear Science. Vol54. Issue 1, part 2. pp. 252-262, Feb. 2007

Código de campo cambiado

Código de campo cambiado