Portela-Garcia, M., Garcia-Valderas, M., San Millan, E., Lopez-Ongil, C., Entrena, L., Martin-Ortega, A., Ramon de Mingo, J. & Rodriguez, S. (2011, junio). Sensitivity Evaluation Method for Aerospace Digital Systems With Collaborative Hardening. IEEE Transactions on Nuclear Science, 58(3), 1053-1058.

DOI: 10.1109/tns.2011.2109397

Sensitivity Evaluation Method for Aerospace Digital Systems with Collaborative Hardening

Marta Portela-García[1], Mario García-Valderas[1], Enrique San Millán[1], Celia López-Ongil[1], Luis Entrena[1]

Alberto Martin-Ortega[2], Jose Ramón de Mingo[2], Santiago Rodriguez[2]

Abstract— **Complexity of current digital systems and circuits involves new challenges in the field of hardening and measuring circuit's sensitivity under SEEs. In this work, a new solution for evaluating the SEU sensitivity of space systems based on using programmable logic devices is proposed. This solution is able to perform a deep analysis of fault effects in systems with hardware functionality distribution, taking into account the high complexity of the hardware nodes (complex programmable logic devices) and their collaborative hardening properties.**

***Index Terms*— Aerospace applications, collaborative hardening, radiation sensitivity, SEU**

I. Introduction

IONIZING radiation effects are a fundamental problem in electronic circuits which is aggravated with technology shrinking towards nanometric dimensions and increasing circuit complexity [1]. In the case of aerospace applications, analysis of circuit sensitivity together with the application of error mitigation techniques in critical areas, are mandatory tasks. There are many solutions already proposed for hardening digital systems and circuits, at first stages in design cycle [2][3] as well as at lower abstraction levels related to physical design stages [4][5]. Due to the insertion of redundant structures inside the circuit, any type of hardening technique (either at technological or functional level) always involves penalties in terms of area, cost, weight, power consumption and/or performance. Nevertheless, careful selection of areas to be hardened will reduce considerably this loss of competitiveness. In particular, sensitivity measurement in early stages of design cycle contributes to explore further possibilities in design space, to select the parts of the circuit to harden, and to enlarge reliability in a short period of time and with a low cost.

Furthermore, there are new challenges in the field of hardening, mainly due to the high complexity of current digital systems and circuits. In aerospace applications, on-board electronic systems usually include some functionality distribution among different devices. Reliability and, specially, dependability of this type of systems is a relatively recent problem [6]-[8]. Most of the proposed solutions are focused on applying software-based hardening techniques. However, electronic technology evolution has allowed an increasing presence of powerful functional units that are not executing software, and are highly accelerating the maintenance and the communication tasks inside and outside the spacecraft. Hardening this kind of systems is nowadays a problem to be solved as well as the measurement of the goodness of such solutions.

In this paper, we propose a new method for evaluating the sensitivity under SEEs (Single Event Effects) of aerospace digital systems composed of complex programmable devices, with hardware functionality distribution and collaborative hardening. Among the different kind of SEEs, this work deals with SEUs (Single Event Upsets) and SEFIs (Single Error Functional Interrupt), since when an SEU affects the configuration memory of the programmable device it can provoke a functional error. The approach consists in a hardware-implemented fault injection solution able to perform a deep analysis of fault effects in this kind of complex systems. This method provides a solution for the early evaluation of radiation sensitivity, both at the component and system level. In the experimental results, this method is applied to a distributed architecture of an on-board computer in OPTOS satellite, developed at INTA (National Institute for Aerospace Technique), Spain.

The paper is organized as follows. Section II presents collaborative hardening as a solution for very critical systems made of sensitive components. Section III details the sensitivity evaluation method proposed in this work. In section IV, experimental results are reported. Finally, section V states the conclusions of this work.

II. Collaborative Hardening

Complex digital systems require specific hardening techniques in order to achieve a suitable level of dependability against SEUs, minimizing involved penalties. This is the case of systems with complex programmable logic devices (CPLDs). CPLDs present some very interesting features for their use in space applications. Typical solutions for sub-systems based on programmable logic are achieved by using

[1] Electronic Technology Department, Carlos III University of Madrid, Spain. (marta.portela, celia.lopez, mario.garcia, luis.entrena@uc3m.es)

[2] National Institute for Aerospace Technology (INTA). Madrid, Spain. alberto.martinortega@insa.es

expensive Rad-Hard devices or by triplicating each component and adding and external Rad-Hard voter. However, Rad-Hard technologies usually provide less performance and density, and/or can require higher power consumption, with respect to commercial technology.

A more effective solution consists on adding redundancy in the functions performed by each module, i.e. including redundant tasks in different hardware modules available in the system. With this hardening technique, the different modules collaborate with each other on making the critical tasks in the system dependable by means of adding redundancy, even though each module is not fault tolerant. Figure 1 shows an example of hardware distributed system with collaborative hardening. Each module is in charge of various tasks and the critical task (*task_n*) is performed by different modules at the same time.

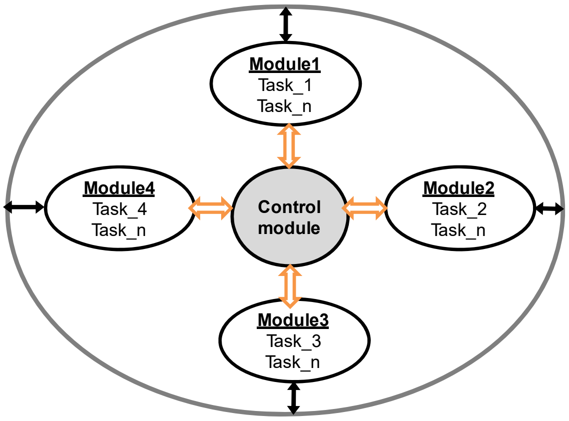

Figure 1. Collaborative hardening scheme in a hardware distributed system.

A possible fault can be detected by the control module, in charge of managing the different tasks in the distributed system, or by the modules themselves by means of the information that they exchange. This method profits from the available resources in the system and allows using modules that are sensitive to SEEs. Therefore, this is a low cost solution to make a fault tolerant system by using logic programmable devices.

III. Sensitivity Evaluation Method

Soft error sensitivity of a circuit depends on technological factors, functional features and the workload. In [9] a methodology to predict the SER of a circuit is proposed. This methodology consists of two main steps:

- Applying a static test, without running any application, in order to measure the soft error sensitivity of the circuit due to technological factors.
- Applying a dynamic test by checking the circuit behavior under faults while the circuit is running with a specific workload.

Using the methodology described in [9], the sensitivity of a circuit running a given application can be calculated as the product of the static cross section and the error rate obtained from a fault injection campaign for such workload.

Static test is usually performed by means of ground radiation testing. Results of static test can be used for different applications since they only depend on the technology used. Dynamic test is necessary for every application. It should be performed with evaluation techniques applicable at early design steps, in order to reduce the cost and time of a redesign, if it is necessary. Thus, functional hardening techniques can be applied during the design of the circuit and the effects of those solutions can be also evaluated at early design stages.

In this section, we describe the evaluation system proposed in this paper in order to perform the dynamic test under SEUs and SEFIs of a hardware distributed system based on programmable logic devices.

The sensitivity of a digital system depends on the sensitivity of each component and on their connections with the rest of the system. In case of a programmable device, SEU sensitivity during a dynamic test depends on the functionality prototyped on every device. Therefore, the dynamic fault rate must be calculated considering the fault rate in Sequential Logic as well as the fault rate in Configuration Memory, taking into account the interaction between the different components.

A. *Fault rate in sequential logic*

In order to evaluate the fault rate in sequential logic, an intensive fault injection campaign is required. We proposed to carry out this experiment by using an Autonomous Emulation System [14]. This method is an FPGA-based fault injection technique that consists in implementing the complete fault injection system in hardware (circuit under test and fault injection tasks). This technique provides very high fault injection rates (around millions of faults/s). This capability is necessary for obtaining a significant measure of SEU sensitivity in current circuits and systems, where the number of possible faults is around hundreds of millions.

In general, fault effect classification adopted is Failure (F) when obtained outputs are different to the expected ones, Latent (L) if the fault produces a different internal state and Silent (S) when there are no differences between the faulty and the golden circuit. However, the obtained fault rate with this fault classification is very pessimistic since, actually, an output error does not necessarily involve a failure in a circuit. In order to perform a further analysis, the Autonomous Emulation System presented in [14] has been modified by adding the following capabilities:

- Different weights can be assigned to outputs, depending on their mission.
- New fault effects classification has been arranged, defining Detected (D) faults where there are output differences but these do not provoke system failures.
- A complete analysis of outputs behavior after fault injection has been introduced in the evaluation tool. This analysis consists in observing the evolution of the fault effect along with the workload, beyond the initial fault classification.

This way, we can distinguish different fault effects with different criticality level. For example, with this extended analysis we can classify faults as FS when the fault produces a Failure but later the effects disappear completely (S), or as FF when the injected fault provokes additional failures during the workload execution, and then we can conclude that this case is even more critical than the fault classified as FS. Another important case is when the fault is classified as D. In this case the fault does not produce any misbehavior but it is important to know if it can provoke one later. Therefore, we have to distinguish among the categories DD, DF, DS, etc.

B. *Fault rate in configuration memory*

A hardware emulation system, intended to evaluate the effect of SEUs in the configuration memory has been developed. It is in charge of modifying the bitstream file for the programmable device by inserting bit-flips and observing the produced effects in the circuit behavior. The scheme of the proposed emulation system is shown in Figure 2. This new hardware emulation system consists of three components: the circuit under test prototyped in a programmable logic device, an FPGA board to implement all the injection tasks and a host PC to manage the complete process.

1) *Host PC*

A software tool has been developed in order to generate the fault list, to insert faults in the configuration memory (bit-flip) and to configure the programmable device (circuit under test) with the faulty bitstream. The fault list has been generated by selecting faults randomly with a uniform probability distribution among all the possible ones.

2) *Fault injection tasks*

Fault injection tasks are implemented by means of an FPGA board. It contains a module in charge of generating the necessary inputs for the target circuit, a block to generate the golden results and a checker module to compare golden and faulty results that allows the detection of failures in the circuit under test.

3) *Circuit under test*

The circuit under test is one of the programmable logic devices that comprise the hardware distributed system. It can be evaluated alone or connected to other system modules to study the effect of applying collaborative hardening techniques.

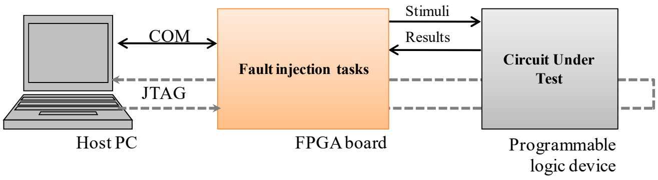

Figure 2 Hardware emulation system used to evaluate SEU effects in the configuration memory of a programmable logic device.

IV. Experimental Results

A. *Case Study: Optos Spacecraft*

The National Institute for Aerospace Technique (INTA) has developed the OPTOS pico-satellite, designed to serve as a testing platform for new technologies in space. In spite of its small size, OPTOS consists of many different subsystems and experiments [11]. In order to prove the feasibility of the proposed evaluation method we have chosen as a case study the distributed architecture of the OPTOS' On-Board Data Handling (OBDH). The OBDH is the subsystem in charge of managing and storing data in the satellite. The OPTOS' OBDH is based on the use of re-programmable devices (FPGAs and CPLDs).

The elements within the system establish communication with each other through a CAN protocol implemented on a diffuse Optical Wireless Link (OWLS). The design concept is particularly applicable to the development of small satellites where requirements of low consumption, low cost, re-use and flexibility must be achieved while maintaining a high degree of reliability at system level.

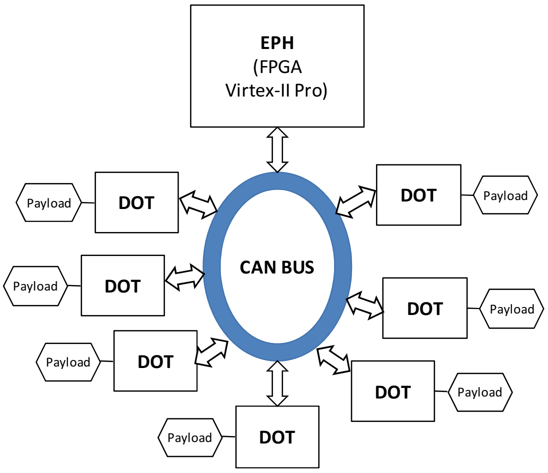

Figure 3. Hardware scheme of the OPTOS' OBDH (On-Board Data Handling)

The design contains the following units:

- EPH (Enhanced Processing Hardware). It is the system core. It consists in a soft core processor implemented on an FPGA, and it is mainly associated to the management of the ground communications subsystem.
- DOT (Distributed OBC Terminal) Units. They are small programmable devices in charge of implementing simple functions related to autonomously commanding and managing different satellite subsystems and units.

In the case of the EPH, a Xilinx's VIRTEX II Q-pro FPGA has been chosen. It guarantees up to 200 krad and latch-up free operation. Nevertheless, it is very sensitive to particle impact (protons and heavy ions) of energies above 1 MeV.

Regarding DOTs, they are implemented by using Complex Programmable Logic Devices (CPLDs). In particular, CoolRunner-II^{TM} (CR-II) CPLDs from Xilinx was the technology selected. The reliability requirements impose SEU sensitivity must be limited to less than 1 functional error (SEFI) every 30 days. The proton tests carried out on the CoolRunner-II^{TM} [13] indicate that the error rate in the configuration memory due to radiation is under this limit. The way to prevent the effect of accumulated errors is restarting the device to reconfigure it, which in the case of DOTs means switching the device off and on. Each DOT will be operating less than a maximum time (around 30 minutes). Therefore, frequent reconfiguration is possible. A DOT which is not doing any operation related to its associated payload can be switched off.

One of the critical tasks in the satellite is to keep the real time updated (Real Time Distribution module). In order to implement this task with fault tolerance, a specific protocol has been developed based on collaborative hardening:

1. The EPH shall be responsible for receiving the real time when it is running under communication with the earth-station mode and for introducing it in the CAN bus.

2. Every active unit shall store this time as the satellite's valid real time.

3. Every active unit shall try to introduce the time command (including real time) into the bus every second. The unit with the highest priority will manage to transmit first.

4. The other units will abort their own time command transmission and shall check all the received time information and compare it with their own time.

Each unit is susceptible to fail as shown in the dynamics test results in [13]. However, the reliability is assured through a collaborative scheme. Every awaked unit is simultaneously running these critical procedures and whenever an error is detected, the unit is reconfigured erasing any previous upset.

This kind of complex systems must be quite difficult to test without a high level test procedure as the one shown hereafter.

B. Device Technology

CoolRunner-II^{TM} devices present ultra low power consumption and wide configurability. With respect to fault tolerance, they present three important sensitive areas:

- Non-volatile configuration memory (Flash) which keeps the configuration to be programmed on the device.

- Volatile configuration memory (SRAM). Whenever the device is powered up, the non-volatile memory is transferred onto volatile one.

- Sequential Logic is SEU sensitive too, but its small size reduces the error probability compared to configuration memory.

Proton irradiation tests were performed on this technology in order to calculate the Soft Error Rate [13]. Static and dynamic tests were performed in order to check sensitiveness of internal memories. The application run in the CoolRunner-II was a pipelined multiplier with two 10-bit operands and 20-bit result. The prototyped multiplier (5-stage pipelined) used almost all the available resources inside the tested device. The tests results are summarized below:

- No parts failed from TID up to 22 kRad.
- No Single Event Latch-up was detected for proton energies up to 63MeV and fluencies around 10^{10} protons/cm^2.
- Cross -section did not increase for proton energies greater than 30 MeV (saturation cross-section), while proton energy threshold is less than 10MeV.
- No SEU was observed inside Flash memory.
- Regarding volatile memory sensitivity, SRAM cells are quite sensitive to protons with energies greater than 15 MeV. SEU and SEFI rate predictions were got from CREME96 code with cross-section data (as a function of proton energy) and orbital parameters (680 km orbit and 98º, typical Low Earth Orbit). An MTBF (Mean Time Between Failure) of 11 days is expected in the worst case. In dynamic tests with the pipelined multiplier, the SEFI rate is 25% failure/SEU. Therefore, the measured MTBFF (Mean Time Between Functional Failure) will be 44 days, since only a fraction of the SEUs will produce a functional failure.

In general, shutting down the device, when its functionality is not needed, avoids error accumulation in memory elements. If CoolRunner-II devices can work in short time slots, errors in SRAM memory can be eliminated in every reboot.

C. Evaluation Results on OPTOS' OBDH with the Proposed Method.

Sensitivity evaluation on OTPOS' OBDH system has been divided in two steps. First, effects of faults in CoolRunner-II^{TM} (CR-II) configuration memory have been evaluated. Secondly, fault injection campaigns have been performed in order to analyze the robustness of sequential logic.

1) Sensitivity evaluation on configuration memory

Table 1 shows the results obtained when faults in CR-II configuration memory are injected. Three different applications have been used. Each fault injection campaign consists in injecting a uniform sample out of 296,402 possible memory locations.

Firstly, a fault injection experiment has been performed using the pipelined multiplier that was used in the dynamic irradiation tests [13]. This circuit involves the use of more than 90% of the available resources in the CR-II device. In this experiment, 79% of the possible faults have been evaluated. This fault injection campaign took several days. The functional failure rate obtained from fault injection is around 18% failure/bit-flip. Applying the methodology described in [9], the MTBFF is calculated like the product of the error rate and the static cross section published in [13]. Therefore, the obtained MTBFF is 32 days versus the 44 days obtained from the irradiation tests. In general, results obtained regarding functional failures with this fault injection method provide more information than dynamic irradiation test results, where only 100 faults were achieved. The results for the pipelined multiplier can be considered like a worst case, since the pipeline structure prevents functional masking errors and the CR-II device is practically fully occupied.

The other evaluated applications are the RTDM (Real Time Distribution Module) and the CAN Manager module that are real satellite's tasks. These applications have been chosen to measure the soft error sensitivity under a workload since they are actual critical tasks that are going to be implemented in the circuit during normal operation. The workload executed by RTDM consists of more than fifty-one million clock cycles and for the CAN Manager module the workload consists of more than fifty-four thousand clock cycles.

The result data greatly depend on the area occupied in the programmable logic (only 12% of device is used for RTDM and 4% for the CAN Manager). The results show the low error rate that these modules present. This is due to the intrinsic fault tolerance that characterizes them and the collaborative hardening techniques used.

TABLE 1. FAULT INJECTIONS RESULTS ON CR-IITM CONFIGURATION MEMORY FOR A MODULE OF A SINGLE DOT

Block	#Injected faults	Failure	Silent
Multiplier	235,078	41,387(17.6%)	193,691 (82.4%)
RTDM	30,585	1,293 (4.2%)	29,292 (95.78%)
CAN Manager	235,439	1,173 (0.5%)	234,266(99.5%)

2) *Sensitivity evaluation on sequential logic*

In order to analyze sequential logic sensitivity, fault injection campaigns have been performed on two internal modules of single DOTs, RTDM and CAN Manager Module. A complete analysis of fault effects is obtained by means of Autonomous Emulation. Three billion and one million faults have been injected and evaluated in RTDM and CAN Manager respectively. The fault effects depend on the circuit activity and therefore, it is necessary to evaluate the fault sensitivity of the circuit when it is executing the typical workload.

In Table 2, the obtained fault classification for each experiment is shown. In this report, typical analysis for user memory elements is given; percentages of faults provoking *failures*, *latent* or *silent* faults provide a general idea of the module robustness. On the columns on the right, the enhanced fault classification is detailed.

Enhanced fault classification consists in keeping the observation of the fault effect in the circuit beyond the first classification. Thus, the fault injection system stores the initial classification of a fault, the next one and the final fault classification. The different categories are named with three characters that represent the successive classifications. In case the fault effect disappears during the workload execution in the initial or secondary classification, one or two characters can be sufficient to name the corresponding category (S, FS, DS). The enhanced classification for each fault stops when the fault becomes silent or when the execution of the workload ends. The proposed enhanced classification could be extended if necessary by storing a higher number of intermediate classifications. The implementation of a fault injection system with enhanced classification requires dedicated hardware to store the necessary information. It increases the length of a fault injection campaign with respect to a typical one since an enhanced classification does not stop the fault observation when a fault is classified as failure. Fault injection with typical fault analysis provides rates of around millions of faults per second, while with enhanced analysis the rate is around thousands of faults per second.

All the considered categories finish with silent effects since during the system execution the modules are continuously receiving correct data from other modules in the system (real time in the RTDM case, and CAN messages for the CAN Manager module). This is because of the collaborative hardening technique applied in the OPTOS satellite. While fault disappearing (S) and single misbehaviors in outputs (FS, DS, FLS) are easily manageable, recurrent failures or loss of synchronism require some extra mitigation techniques (FFS, DDS, DFS). These added mitigation techniques are mainly related to invisible counters which are non-rewritable from the system, or to flag signals in charge of synchronizing different modules inside or outside DOTs. Passive hardware redundancy (TMR) provides a high degree of robustness in very critical cases but it involves a high area overhead.

In the RTDM, the critical outputs are those that indicate when a certain time period (ms or sec) has elapsed. 73.3% of the faults provoke failures in these critical outputs (FS, DFS and FFS). On the one hand, faults classified like DFS and FFS produce a difference in the output activation resulting in two failures. Nevertheless, the time value is recovered during the next activation of the output and therefore, these faults do not affect the real time value. On the other hand, faults classified as FS provoke a missing activation of one of the critical outputs or an additional one. Therefore, these faults (3.3% out of the injected faults) have to be masked or prevented, since they corrupt the stored real time.

Regarding the CAN Manager module, 71% of the faults produce a failure, i.e. a wrong value in the outputs. However, not all the outputs are critical. Analyzing in more detail the faults that produce a failure in the circuit, we can grade the severity of the fault effect. From the total number of failures just 15% of the faults produce critical failures. These critical failures are due to misbehavior of the circuit in charge of restarting the communication. Therefore, these critical failures can be prevented by hardening the flip-flops used to implement this functionality. It requires hardening only three flip-flops (15% of the total number of the flip-flops used for implementing the CAN Manager module).

Global analysis could be performed on the whole system (with several DOTs) to check fault masking effects that will confirm previous categorization applied on the outputs of different modules. Only a fault injection campaign through hardware emulation is able to evaluate such a complex system, dealing with thousands of millions faults.

V. CONCLUSION

In this paper, a new method for evaluating the radiation sensitivity of aerospace digital systems composed of complex programmable devices, with distributed hardware functionality and collaborative hardening is presented. The evaluation method analyses in depth the fault effects in both the sequential logic as well as in the memory configuration of the programmable device.

In order to inject and analyze faults in the configuration memory a new hardware emulation system has been developed. Regarding faults in sequential logic, Autonomous Emulation has been used and a new fault classification technique has been proposed and implemented in order to consider the different cases that can occur in a digital system made of sensitive components. With this approach a detailed analysis of the fault effects during the complete workload execution can be performed.

Only fault injection through hardware emulation is able to evaluate such a complex system, dealing with thousands of millions faults. The proposed method, along with irradiation experiments for technology characterization, allows the insertion of further mitigation techniques in early design stages. A significant reduction in design time and final cost can be achieved for this type of systems.

REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS). 2005-2009 Editions

[2] D. K. Pradhan "Fault-Tolerant Computer System Design" Prentice Hall, 1996

[3] C. López-Ongil, L Entrena, M. Garcia-Valderas, M. Portela-Garcia "Automatic Tools for Design Hardening" in "Radiation Effects on Embedded Systems" Springer (ISBN: 978-1-4020-5645-1) 2007

[4] M. Nicolaidis "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies" IEEE 17th VLSI Test Symposium, pp. 86-94, april 1999

[5] M. Nicolaidis "Design for Soft Error Mitigation" IEEE Transactions on Device and Materials Reliability, Vol. 5, No. 3, September 2005

[6] K. Kyamakya, K. Jobmann, M. Meincke "Security and Survivability of Distributed Systems: An overview" Proceedings of IEEE Milcom, Los Angeles CA. Mayo 2000. Pp 449-454.

[7] E. A. Strunk, J.C. Knight, M.Anthony Aiello. "Distributed Reconfigurable Avionics Architectures" 23rd Digital Avionics Systems Conference, Salt Lake City. October 2004.

[8] López, J.; Royo, P.; Barrado, C.; Pastor, E. "Modular Avionics for Seamless Reconfigurable UAS Missions" 27th Digital Avionics Systems Conference: Integrated modular avionics is the modern approach. [NJ: IEEE], 2008

[9] S. Rezgui, R. Velazco, R. Ecoffet, S. Rodriguez, J. R. Mingo "Estimating error rates in processor-based architectures" IEEE Transactions on Nuclear Science, Vol. 48, Issue 5, pp. 1680-1687, October, 2001

[10] Xilinx. White Paper 215 "CoolRunner CPLD Radiation Sensitivity" October 2004. 1-800-255-7778.

[11] I. Lora et al. "INTA PicoSatellite OPTOS: Mission, Subsystems, and Payload". 4th Annual CubeSat Developers' Workshop. 2007.

[12] "Optical Wireless for Intra-Spacecraft Communications (OWLS)". European Space Agency (ESA) Technology Research

[13] M. García-Valderas, M. Portela-García, C. López-Ongil, L. Entrena, A. Martin-Ortega, J. R. de Mingo, M. Álvarez, S. Esteve, S. Rodríguez. "The Effects of Proton Irradiation on CoolRunner-II CPLD Technology" Radiation Effects on Components and Systems Workshop (RADECS'08).Finland

[14] C. López-Ongil, M. García-Valderas, M. Portela-García, L. Entrena, "Autonomous Fault Emulation: A New FPGA-based Acceleration System for Hardness Evaluation", IEEE Transactions on Nuclear Science, Vol. 54, Issue 1, Part 2, pp. 252-261, Feb. 2007

TABLE 2. EXAMPLE OF FAULT RATE ANALYSIS FOR SINGLE NODES IN OPTOS' OBDH

Block	# Faults	#FFs	Typical analysis			Enhanced analysis						
			F	S	L	FS	FFS	FLS	S	DS	DDS	DFS
RTDM	3.123.279.483	61	80.3%	19,7%	0,0%	3,3%	20%	0%	0%	3,3%	20%	50%
CAN Manager	1.080.040	20	70.6%	28,8%	0,6%	34%	26%	11%	29%	0%	0%	0%