A Study of the Robustness Against SEUs of Digital Circuits Implemented with FPGA DSPs

Felipe Serrano, Juan Antonio Clemente and Hortensia Mecha
Computer Architecture Department, Universidad Complutense de Madrid (UCM), Madrid, Spain

***Abstract*—In this paper we present an experimental validation of the reliability increase of digital circuits implemented in Xilinx™FPGAs when they are implemented using the DSPs (Digital Signal Processors) that are available in the reconfigurable device. For this purpose, we have used a fault-injection platform developed by our research group, NESSY [1]. The presented experiments demonstrate that the probability of occurrence of a SEU effect is similar both in the circuits implemented with and without using embedded DSPs. However, the former are more efficient in terms of area usage, which leads to a decrease in the probability of a SEU occurrence.**

I. Introduction

FPGAs feature an interesting trade-off between performance and flexibility, which makes them suitable for aviation, nuclear and spatial applications. However, in these hazardous environments, devices are usually exposed to a high amount of radiation [2]. This can lead to the appearence of Single Event Upsets (SEUs), which consist in a change in the content of a memory cell. In SRAM-based FPGAs, SEUs affect specially to their configuration memory and can thereby provoke serious errors in the normal operation of the system.

For this reason, the development and adoption of techniques that neutralize or mitigate the effect of SEUs in embedded has become an issue of a great importance during the last few years. A widespread technique to protect circuits against SEUs is the Triple Modular Redundancy (TMR) [3]. It consists on replicating the circuit three times and processing the three obtained outputs by *voting* them (i.e., selecting the most frequently occurred result) in order to generate a single output. Thus, if one of the instances of the circuit delivers an incorrect output as a consequence of a SEU, the voter can mask this error and the faulty instance can be corrected by means of partial reconfiguration. This approach is very effective; however it has an extremely high cost in terms of area consumption, since the original circuits must be replicated 3 times. Hence another interesting alternative is a technique known as ”scrubbing”, which consists on reconfiguring the whole circuit periodically [4]. Nevertheless, this solution generates some performance overhead since the reconfiguration process takes in the order of milliseconds to be completed. This overhead can be unaffordable in some cases.

This paper presents a different solution to increase the robustness of digital circuits implemented on FPGAs: *implementing them by using embedded DSPs existing in the target FPGA, provided that they are available in the device*. Thus, this paper presents an experimental study that compares the robustness of real digital circuits against SEUs when they are implemented with and without using the DSPs that are embedded in a Xilinx™Virtex 5 FPGA. The experimental results demonstrate that the versions of the circuits that use DSPs are more robust against SEUs than their counterparts implemented without using them.

In order to obtain these results, we have used an option that exists in the synthesis and implementation processes of the Xilinx™ISE 12.1 development tool, which activates/desactivates the use of DSPs for the final implementation of the circuits. In this study we have tested several versions of a complex circuit: a Feed Forward Equalizer (FFE) filter.

The robustness of the different implementations has been evaluated with the fault-injection platform NESSY [1]. It emulates a SEU by modifying a bit in the region of the configuration memory that implements the circuit under test. NESSY has been implemented using a Xilinx™XUPV5-LX110T development board, which features a Virtex 5 FPGA. However, the presented results are extensible to any other Xilinx™Virtex device.

Other works existing in the literature [5], [6], [7], [8], [9], [10] propose different fault injection platforms on circuits implemented on FPGAs and present experimental results obtained by performing exhaustive fault injection campaigns. However, to the best of our knowledge, *none of these works propose a comparative study as the one that we propose in this article*.

II. Related Work

During the last few years, important efforts have been made in order to know in depth the impact of SEUs on circuits implemented on SRAM-FPGAs. A popular technique consists on exposing the configuration memory of the FPGA to protons, neutrons [11] or ionized particles, such as argon o carbon [12]. However, these tests are usually very expensive (in fact, their typical cost is in the order of hundreds of thousands of dollars), they are very time consuming and in many occasions it is impossible to control with absolute precision the amount and location of the errors that have been found, which is crucial in such kind of experiments. For this reason, alternative approaches, such as *SEU emulation*, have become a popular alternative. These techniques emulate the occurrence of SEUs by injecting *bitflips* (i.e., changing the value of a cell memory from 0 to 1 or vice-versa) in the configuration memory of the FPGA through the partial reconfiguration of the device.

A number of fault injection platforms for digital circuits implemented on FPGAs have been presented in the literature.

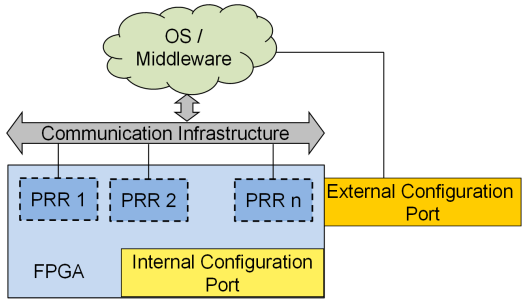

Fig. 1. Hardware multitasking system onto which the digital circuits tested with NESSY will be deployed

Firstly, FLIPPER [5], [6] is a platform that emulates SEUs by modifying the contents of the configuration memory of a Xilinx™ Virtex-II Pro FPGA. It has been validated by exposing it to radiation using a proton accelerator [6]. However, the limitations of this platform are: 1) It is not easily portable to others FPGAs and 2) it introduces modifications on the circuit under test, hence it does not allow to test the very same circuit that will finally be implemented in the FPGA for a real mission. If a fault injection platform has this defect, it is commonly known as an intrusive platform [9].

Secondly, FT-UNSHADES [7] was initially developed to analize the effect of SEUs in VLSI designs. In [8] the authors present an extended platform, which can emulate SEUs in the configuration memory of a Xilinx™ Virtex-II Pro FPGA. The main feature of this platform is its high performance. However, its scope is very limited for testing circuits on reconfigurable systems, since *it only emulates SEUs in the regions of the configuration memory that modify the content of the secuential elements of the circuit under test* (flip-flops, registers, BRAM memory blocks), skipping the rest of the configuration bits. In addition, this platform is instrusive.

Finally, the CAD group of the Politecnico di Torino University [9], [10] have presented another two fault injections platforms for Xilinx™ Virtex-II Pro FPGAs. On the one hand, similarly as FT-UNSHADES, the platform presented in [9] achieves a high performance. However, it is an intrusive platform. On the other hand, in [10] the same authors present an enhanced version of [9] that guarantees its non-intrusiveness. However, this feature leads to a huge performance loss. Thus, the average fault injection time of [10] increases by about three orders of magnitude with respect to [9].

In spite that these platforms share the same objetives and the methodology as NESSY, is very important point that, to the best of our knowledge, *neither in these works nor in the state-of-the-art it has been presented a comparative study of the robustness of circuits implemented in FPGAs as the one that we present in this article*.

III. The SEU-Emulation Platform NESSY

NESSY is a SEU emulation platform that, contrarily to other state-of-the-art approaches, features a high performance and non-intrusiveness at the same time. These properties are achieved having in mind the applications are be executed in the hardware multitasking system that is depicted in Figure 1.

The target FPGA is divided in a set of *Partial Reconfigurable Regions* (PRRs). These PRRs are conected to a *Communication Infrastructure* that can be implemented by means of a bus or a *Network-on-Chip* (NoC) [13]. Each task to be executed in this system is placed and routed in one of these PRRs and it communicates with the remainder of the system using the aforementioned communication infrastructure. The I/O of the tasks is implemented along with the hardware of the task mapped in the corresponding PRR. Each PRR is configured by means of partial reconfiguration, either by using a configuration port (either internal or external).

NESSY generates the partial and global bitstreams of the system by invoking the Xilinx™ PlanAhead and EDK 12.1 development tools. It is also important to point out that this tool emulates SEUs *without introducing any modification in the placement and routing of the circuit under test except the injected bitflip*. Hence it is a non-intrusive platform.

IV. Interpretation of the Virtex 5 Architecture by NESSY

In the previous version of NESSY [1], this platform injected bitflips *only* in the bits that contained configuration information of the *Configurable Logic Blocks* (CLBs) of the FPGA, thereby skipping the configuration bits regarding the remaining logic resources. In order to perform the study presented in this article, NESSY has been enhanced in order to emulate SEUs in the DSPs embedded in the FPGA too.

In a Virtex-5 FPGA, the logical resources are organized as a bidimensional matrix of units of resources, which we have named *resource blocks*. Each resource block has a set of resources of the same type (CLBs, DSPs, ...), as well as a number of *wiring matrices* associated to these resources. In this work we have distinguished two kinds of blocks:

- *CLB-Blocks* (Figure 2), which are composed of a CLB and 2 wiring matrices associated to it.
- *DSP-Blocks* (Figure 3), which are composed of 2 DSPs and 5 wiring matrices associated to them.

A frame is the minimal unit of configuration. This means that, in order to inject a bitflip in a circuit via partial reconfiguration, the configuration port must load, *at least*, the configuration information of the frame that contains the involved configuration bit.

The configuration information associated to a number of adjacent resource blocks located in the same column is contained in a set of frames, each one of them comprising 40 32-bit words (hence, $40 * 32 = 1.280$ bits per frame). Thus, a frame contains configuration information of 20 *CLB-Blocks* or 4 *DSP-Blocks*. Figure 4 illustrates the organization of these two kinds of resource blocks, as well as the equivalence existing between these resource blocks and the configuration information associated to them.

Through a deeper analysis, we have noted that the matrix of resource blocks is irregular; in other words, each resource block is not configured with the same number of frames. Thus,

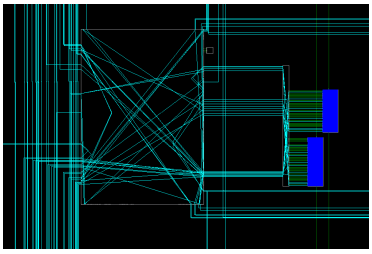

Fig. 2. Internal structure of a *CLB-Block* observed with the Xilinx™FPGA Editor tool. The blue rectangles represent the two slices that are comprised in a CLB, whereas the grey-bordered black rectangles represent wiring matrices

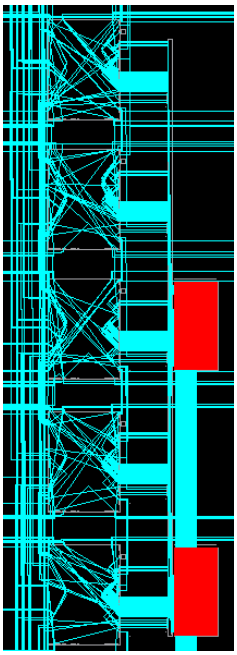

Fig. 3. Internal structure of a *DSP-Block* observed with the Xilinx™FPGA Editor tool. In this case, the red rectangles represent DSPs

36 frames are needed to configure a set of 20 *CLB-Blocks*, whereas 28 frames are associated to the configuration of a set of 4 *DSP-Blocks*. Therefore, each set of 20 *CLB-Blocks* is reconfigured with $36 * 1.280 = 48.060$ configuration bits, whereas in order to configure a set of 4 *DSP-Blocks*, $28 * 1.280 = 35.840$ configuration bits are needed.

Thus, with this information, NESSY is able to iterate accross the whole bitstream in order to emulate SEUs in the FPGA resources in a controlled way.

V. Experimental Results

In this section we present the SEU-emulation experimental results obtained for two versions of the same circuits: when they are implemented with and without using FPGA DSPs. In our experiments we have used two circuits:

- A FFE filter, which is a complex circuit used in spatial applications. It features several consecutive filtering steps. In our experiments, we have emulated SEUs in three different versions of this circuit: featuring 2, 4 and 8 filtering steps, respectively.

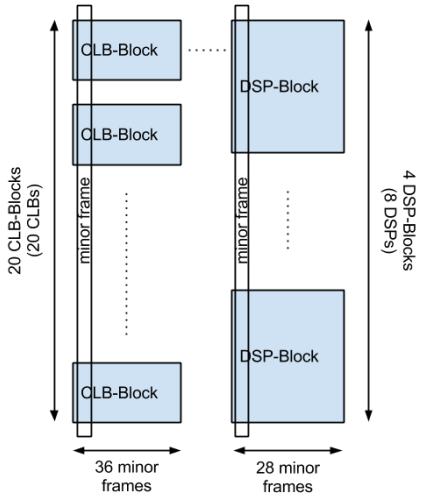

Fig. 4. Blocks of resources in a Xilinx™Virtex 5 and their associated segments of configuration information

- A Finite Input Response (FIR) filter with only one filtering step. We have used this circuit because it fits *exactly* in a DSP, thereby using the 100% of the available resources existing in it. This circuit implements the function $y(n) = a(n) \cdot b(n) + c(n)$. Thus, it has allowed us to obtain an upper-bound of the reduction of the size of the circuits when implementing them using FPGA DSPs.

A. SEU-Emulation Results

Table I shows the SEU-emulation results for the evaluated circuits. Rows 1, 3 and 5 correspond to the FFE implemented without using DSPs, with 2, 4 and 8 filtering steps respectively. Rows 2, 4 and 6 do likewise with the versions of the circuits that use DSPs. Column 2 shows the number of SEUs emulated in the whole PRR where the circuits under test are placed and routed, whereas Columns 3 and 5 only account for the SEUs emulated on the FPGA resources actually used by the circuits under test. In order to know this information, we have used the Xilinx™FPGA Editor tool. Next, Columns 4 and 6 show the number of errors affecting *CLB-Blocks* and *DSP-Blocks*, respectively. Finally, Column 7 shows the percentage of effective SEU emulations that resulted into errors in the evaluated circuits. In the versions of the circuits implemented using DSPs we have computed a weighed average using the different number of errors found in *CLB-blocks* and *DSP-Blocks* with respect to the total. Note that *these percentages have been computed with respect to the number of effective SEU emulations*, in order not to distort these results when using PRRs when the evaluated circuits do not *exactly* fit.

In Table I we can appreciate that these percentages do not significantly increase nor decrease in the versions of the circuits that are implemented using DSPs. Thus, we can assert that *implementing circuits on Xilinx™Virtex 5 FPGAs using DSPs rather than CLBs does not significantly modify their robustness against SEUs at the structural level.*

TABLE I
SEU EMULATION RESULTS

Circuit	*# Total Injections*	*CLB-Blocks*		*DSP-Blocks*		*% Total SEU Effects*
		# Effective Injections	*# SEU Effects*	*# Effective Injections*	*# SEU Effects*	
FFE (2) without DSPs	276.480	258.048	7.892	N/A	N/A	3,06%
FFE (2) with DSPs	128.000	43.776	1.763	17.920	838	4,22%
FFE (4) without DSPs	555.520	516.096	15.954	N/A	N/A	3,09%
FFE (4) with DSPs	128.000	82.944	2.659	35.840	1.560	3,55%
FFE (8) without DSPs	1.013.760	963.072	52.209	N/A	N/A	5,42%
FFE (8) with DSPs	256.000	154.368	7.529	71.680	4.634	5,38%

However, the versions implemented using DSPs have an important advantage: they are considerably more efficiently implemented in terms of area than the equivalent versions that do not use DSPs. In fact, our experiment with the FIR filter allowed us to know that the logic contained in each *DSP-Block* is equivalently synthetized using 106 *CLB-Blocks*. Since 8.960 configuration bits are needed in order to configure a *DSP-Block* and $106 * 2.304 = 244.224$ bits are needed to configure 106 *CLB-Blocks*, the amount of the configuration information needed to configure those resources is reduced by 96,33% with respect to the latter case. This percentage can be seen as an upper-bound of the potential reduction in the configuration memory space that can be attained when implementing embedded circuits on Xilinx^{TM}Virtex 5 FPGAs using DSPs. In the experimental results presented in this paper, we have achieved an average reduction of 76,53% in the required configuration information.

To sum up, we can affirm that a circuit implemented in a Xilinx^{TM}Virtex 5 FPGA is more robust to SEUs when it is implemented using DSPs, due to its greater implementation efficiency in terms of area. Thus, in this case there exists a lower probability that a highly ionized particle collides with the circuit and therefore causes a SEU effect on it.

VI. CONCLUSIONS AND FUTURE WORK

This paper presentes an experimental study of the robustness increase against SEUs of digital circuits implemented in Xilinx^{TM}FPGAs when they are implemented using the DSPs that are embedded in the reconfigurable device. The presented experimental results have demonstrate that, for several versions of a FFE filter, the structural robustness against SEUs remains the similar in both versions of the same circuit, but the number of configuration bits needed by the version implemented using DSPs decreases by 76,53% on average. For this reason, the probability of occurrence of a SEU decreases in the circuits that are implemented using FPGA DSPs. As future work, we want to carry out a similar study, but this time involving SEU emulation in the embedded Block RAMs of the FPGA.

ACKNOWLEDGMENT

This work was supported by the Spanish Ministry of Education, Science and Sports under grants AYA2009-13300-C03-02 and TIN2009-09806.

REFERENCES

[1] V. Alaminos, F. Serrano, J. A. Clemente, and H. Mecha, "Nessy: An implementation of a low-cost fault-injection platform on a virtex-5 fpga," in *the Conference on Radiation and its Effects on Components and Systems (RADECS)*, Sept. 2012.

[2] J. F. Ziegler and W. A. Lanford, "The effect of sea-level cosmic rays on electronic devices," *Journal of Applied Physics*, vol. 52, no. 6, pp. 4305–4312, 1981.

[3] E. A. Stott, N. P. Sedcole, and P. Y. K. Cheung, "Fault tolerance and reliability in field-programmable gate arrays," *IET Computers & Digital Techniques*, vol. 4, no. 3, pp. 196–210, 2010.

[4] A. Tiwari and K. Tomko, "Enhanced reliability of finite-state machines in fpga through efficient fault detection and correction," *IEEE Transactions on Reliability*, vol. 54, no. 3, pp. 459 –467, Sept. 2005.

[5] M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, S. Pastore, and G. Sechi, "Evaluation of single event upset mitigation schemes for sram based fpgas using the flipper fault injection platform," in *Defect and Fault-Tolerance in VLSI Systems, 2007. DFT '07. 22nd IEEE International Symposium on*, sept. 2007, pp. 105 –113.

[6] M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, D. Codinachs, S. Pastore, C. Poivey, G. Sechi, G. Sorrenti, and R. Weigand, "Experimental validation of fault injection analyses by the flipper tool," *Nuclear Science, IEEE Transactions on*, vol. 57, no. 4, pp. 2129 –2134, aug. 2010.

[7] C. Lopez-Ongil, L. Entrena, M. Garcia-Valderas, M. Portela, M. Aguirre, J. Tombs, V. Baena, and F. Munoz, "A unified environment for fault injection at any design level based on emulation," *IEEE Transactions on Nuclear Science*, vol. 54, no. 4, pp. 946–950, Aug. 2007.

[8] L. Sterpone, M. Aguirre, J. Tombs, and H. Guzman-Miranda, "On the design of tunable fault tolerant circuits on sram-based fpgas for safety critical applications," in *Design, Automation and Test in Europe, 2008. DATE '08*, march 2008, pp. 336 –341.

[9] L. Sterpone and M. Violante, "A new partial reconfiguration-based fault-injection system to evaluate seu effects in sram-based fpgas," *IEEE Transactions on Nuclear Science*, vol. 54, no. 4, pp. 965–970, Aug. 2007.

[10] N. Battezzati, L. Sterpone, and M. Violante, "A new low-cost non intrusive platform for injecting soft errors in sram-based fpgas," in *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*, 30 2008-july 2 2008, pp. 2282 –2287.

[11] K. Morgan, M. Caffrey, P. Graham, E. Johnson, B. Pratt, and M. Wirthlin, "Seu-induced persistent error propagation in fpgas," *IEEE Transactions on Nuclear Science*, vol. 52, no. 6, pp. 2438–2445, Dec. 2005.

[12] R. Velazco, G. Foucard, and P. Peronnard, "Combining results of accelerated radiation tests and fault injections to predict the error rate of an application implemented in sram-based fpgas," *IEEE Transactions on Nuclear Science*, vol. 57, no. 6, pp. 3500–3505, Dec. 2010.

[13] L. Benini and G. De Micheli, "Networks on chip: a new paradigm for systems on chip design," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2002, pp. 418–419.