Experimental Validation of a Tool for Predicting the Effects of Soft Errors in SRAM-Based FPGAs

L. Sterpone, M. Violante, R. Harboe Sorensen, D. Merodio, F. Sturesson, R. Weigand, and S. Mattsson

***Abstract*—Estimating the impact of Single Event Effects (SEEs) on SRAM-based FPGA devices is a major issue in order to adopt them in radiation environments such as space or high altitude. Among the available approaches, we proposed an analytical method to predict SEE effects based on the analysis of the circuit the FPGA implements, which does not require either simulation or fault injection. In this paper we provide an experimental validation of this approach, by comparing the results it provides with those coming from accelerated testing. We adopted our analytical method for computing the error cross section of a design implemented on SRAM-based FPGA devices. We then compared the obtained figure with that obtained by accelerated testing. Experimental analysis demonstrated that accelerated testing closely match the figures the analytical method provides.**

***Index Terms*—FPGA, radiation effects, single event upsets.**

I. Introduction

As SRAM-based Field Programmable Gate Array (FPGA) devices become more and more attractive for several applications targeting environments where ionizing radiation abounds (like space or avionics applications), the interest for techniques aiming at predicting the effects of radiation-induced soft errors (SE) in such devices is increasing.

On the one side, state-of-the-art SRAM-based FPGAs embed plenty of resources (RAM modules, processors, configurable logic and routing resources) that allow the implementation of a huge spectrum of applications, moreover they support *in-situ* reconfiguration capability. As designers are increasingly attracted by the idea of deploying the features SRAM-based FPGAs offer in harsh environments, they need tools to quantify easily the dependability of designs implemented on such devices.

On the other side, modern SRAM-based FPGAs are moving quickly from 90 nm to 65 nm technologies. This technological shift, which has positive impact on device density, power consumption, and performance, is raising issues concerning the dependability of the new devices. Since SRAM-based FPGAs are increasingly sensitive to radiation-induced SEs [1], [2], it is mandatory to adopt tools that allow estimating the dependability of circuits implemented on such devices.

When considering SEs affecting the memory elements embedded in FPGA-based systems, two aspects should be considered: the user memory elements and the configuration memory. The user's memory elements (such as registers, memory blocks, etc.) must be hardened against SEs (known as Single Event Upsets, SEUs) that may alter the information the circuit elaborates. This effect can be considered as temporary, since the disruption lasts as soon as a new correct value is written in the memory element the SE affected. Similarly, for the considered class of devices being the configuration memory composed of SRAM cells, its content may be affected by energetic particles hitting the FPGA's surface, and therefore the information the memory cells hold, which defines the function (i.e., the circuit) the FPGA implements, may be corrupted. Although this effect can be viewed as temporary, since it disappears as soon as a new configuration data is loaded in the FPGA's, in practical terms it is very critical as the configuration memory is generally written once (at power-up, only, unless mitigation techniques requiring to configure the device more frequently are used) and therefore in case the configuration memory is altered, it is not restored until the next power-up. Several approaches have been developed for the purpose of avoiding or masking these effects. Approaches likes Triple Modular Redundancy (TMR), which exploits passive redundancy, can be used to mask the effects of SEUs (i.e., prevent the generation of wrong outputs) by replicating the circuit three times and adding majority voters [3]. The TMR technique theoretically guarantees that any SEU affecting any of the circuit replicas is masked by the majority voters; however, recent works showed the limitations of TMR either in presence of single or multiple event provoked by SEs when SRAM-based FPGAs are considered [4], [5]. Moreover, techniques such as memory scrubbing are needed to restore the correct content of the configuration memory [6]. Whatever the hardening technique adopted is, designers need tools to validate the resulting system in order to guarantee the desired dependability degree is reached. Tools must be robust, i.e., they should be able to identify all the possible criticalities that may escape the mitigation techniques designers adopted. Moreover, they must accurately model the effects of SEs on the considered devices. Finally, they must be industry-grade tools, as they must attack the analysis of even very complex designs in a reasonable amount of computing time. The purpose of this paper is to present an experimental evaluation performed using radiation testing of the approach we proposed in [7] to assess the dependability of designs implemented on SRAM-based FPGAs.

The tool, *STAR-LX*, implementing the techniques presented in [7], provides an effective estimation of the SEU-induced effects

Manuscript received July 20, 2007; revised September 22, 2007.

L. Sterpone and M. Violante are with the Dip. Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy (e-mail: luca.sterpone@polito.it; massimo.violante@polito.it).

R. Harboe Sorensen, D. Merodio, F. Sturesson, and R. Weigand are with the European Space Agency, NL-2220 AG Noordwijk, The Netherlands (e-mail: reno.harboe.sorensen@esa.int; david.merodio.codinachs@esa.int; fredrik.sturesson@esa.int; roland.weigand@esa.int).

S. Mattsson is with Saab Ericsson Space, S-411 14 Göteborg, Sweden (e-mail: stanley.mattsson@space.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2007.910122

0018-9499/$25.00 © 2007 IEEE

inside a SRAM-based FPGA system. The analyses STAR-LX performs are workload independent, therefore the time needed to attack even a very complex circuit is negligible when compared to simulation-based techniques, moreover it is two orders of magnitude faster than the first implementation of the approach we presented in [7]. This feature makes STAR-LX suitable for attacking high complexity circuits of industrial interest. From the accuracy point of view, at the core of the tool lies a technological database we developed and validated [8]; therefore, STAR-LX is able to identify very accurately the FPGA's resources (and the circuit's resources they corresponds to) that when affected by SEs overcome the mitigation techniques designers adopted [7].

The main contribution of this paper is an experimental validation of STAR-LX on a complex design. We adopted a test vehicle and we performed two set of experiments. The first one consisted in measuring the error cross-section of the test vehicle through radiation testing. The second one consisted in calculating the error cross-section through STAR-LX.

As a result of these two experiments, we observed that the calculation we made closely matches the experimental measurements, indicating that the approach we propose can be an accurate and efficient estimation tool for the impact of SE in SRAM-based FPGAs.

The manuscript is organized as follows. Section II presents an overview of already developed works concerning the evaluation approaches of designs implemented on SRAM-based FPGA devices. Section III presents an overview of our approach. Section IV details the modules of STAR-LX for predicting the effects of SEs. Section V presents the test vehicle, the accelerated radiation testing we performed and the experimental evaluation. Finally, conclusions are drawn in Section VI.

II. Previous Works

Several approaches are available for analyzing the effects of SEs in FPGA devices. Accelerated radiation ground testing is an effective solution for correctly estimating the SE's sensitivity of both the memory elements used by a design SRAM-based FPGA implements as well as the FPGA's configuration memory [1], [9], [10]. This kind of techniques requires a prototype of the system under analysis, which is exposed to a flux of radiations, originated either by radioactive sources or by particle accelerators, which interacts with both the designs memory elements and the configuration memory. Radiation-testing strategies aiming at validating the robustness of a design (i.e., computing its error cross-section) are usually based on the continuous monitoring of the outputs of the circuit implemented on the FPGA under test, which is continuously stimulated by a given set of inputs. As drawbacks, radiations have the capability of permanently damaging the device under test and the costs (the experimental setup, and the beam time) are not negligible. Moreover, due to the constantly shrinking size of memory elements it is becoming extremely difficult to set-up experiments where particles hit correspond to single upsets, and therefore interpreting the collect data is becoming a difficult task.

As an alternative to radiation testing, several fault-injection approaches were recently proposed. Fault injection is an attractive technique for the evaluation of design characteristics such as reliability, safety and fault coverage [11]. The process involves inserting faults into particular targets in a system and monitoring the results to observe the produced effects. All these approaches emulate the effects of SEs in the FPGA's memory as bit-flips in the bitstream that is downloaded in the FPGA during its programming phase. Some of them use run-time re-configuration [12], while others modify the bitstream before downloading it in the device configuration memory or during download operations [13], [14]. Although the fault-injection approaches permit to evaluate the effects of SEs in all the memory bits, the time needed by the fault-injection process is still huge, even in the case the process is optimized by the use of partial re-configuration.

To overcome the time-consuming processes needed by the fault-injection approaches and to avoid the high cost of radiation testing, analytical approaches based on synthesis tools and software programs, only, are proposed in [15], [16], [7]. In [15] a static estimation of the design's susceptibility to SEs is proposed assuming that all the bits of a design are susceptible at all times. Differently, in [16] an approach is proposed that identifies the paths sensitive to SEs by calculating the error-rate probability of all circuit nodes and by combining it with the error-propagation probability of each net within the design. Then, the obtained information is coupled with the sensitivity of the FPGA's configuration memory bits. These approaches are either very pessimistic or able to provide only probabilistic estimations of SEU effects. In [7] we developed an approach able to analyze the topology of the design implemented on the SRAM-based FPGA, in particular when TMR design techniques are adopted [17], [3] and we coupled this analysis with a set of reliability constraints. Thanks to this approach, our technique is able to achieve the same accuracy of more time-consuming approaches like fault injection, while the execution time is orders of magnitude smaller.

III. An Overview of the Proposed Approach

The tool we developed in order to predict the effects of SE is based on the methodology we presented in [7]. The main purpose of the proposed approach is to analyze the effects of SEs in all the resources a SRAM-based FPGA embeds, as soon as a model of the placed and routed design is available.

The flow of the proposed methodology is depicted in Fig. 1. It is composed of the following modules:

1) *Native Circuit Description*: It is a file containing the structural and layout descriptions of the circuit, which consist of logic functions (either combinational or sequential) and connections between them. Both the logic functions and the connections between them are described in terms of resources placed and routed on the FPGA area.
2) *Redundancy Cluster-Extractor*: It is a module that reads the Native Circuit Description and extracts the place and route information related to each cell of the FPGA architecture. That information is processed by means of a clustering process that groups the data depending on the FPGA topology architecture and on the redundancy structure of the adopted hardening technique.

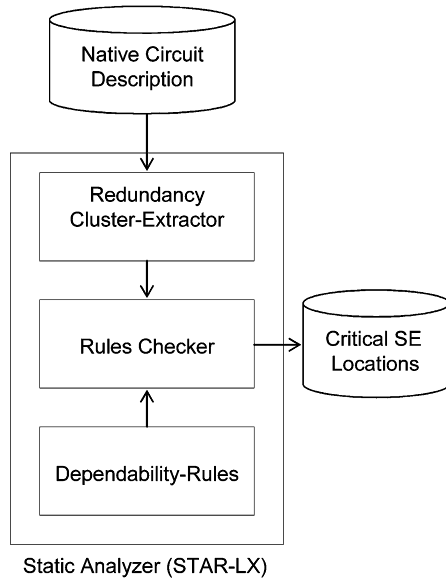

Fig. 1. The flow diagram of the developed Static Analyzer (STAR-LX).

3) *Dependability-Rules*: It is a database of constraints related to the topology architecture of the FPGA that must be fulfilled by the placed and routed circuit in order to be resilient to the effects provoked by SEs.
4) *Rules-Checker*: It is the algorithm that reads each cluster and analyze every bit of the user memory and the configuration memory the FPGA has. It returns a list of SEs (*Critical Single Error Locations*) that introduces critical modifications to the FPGA resource, and therefore to the circuit the FPGA implements, which may overcome the adopted TMR hardening technique.

The STAR-LX tool is based on a SRAM-based FPGA architectural generic model consisting of three kinds of resources, as shown in Fig. 2: *logic blocks, switch boxes* and *wiring segments*.

The logic blocks model the Configurable Logic Blocks (CLBs) and contain the combinational and sequential logic required to implement the user circuit. The input and output signals are connected to adjacent switch boxes through wiring segments. The switch boxes are switch matrices where several *programmable interconnect points* (PIPs) (e.g., pass transistor), called *routing segments* controlled by the configuration memory, are available. We modeled the resources within a SRAM-based FGPA as vertices and edges of a graph. We have *logic vertices* that model the FPGA's logic blocks, *routing vertices* that model the input/output points of the switchboxes, *routing edges* that model the PIPs and *wiring edges* that model the FPGA's wiring segments. More details about the FPGA architectural model can be found in [7].

IV. STAR-LX: The Tool for Predicting the Effects of Soft-Error

The STatic AnalyzeR tool (STAR-LX) works on the graph-model representation of the FPGA internal structure

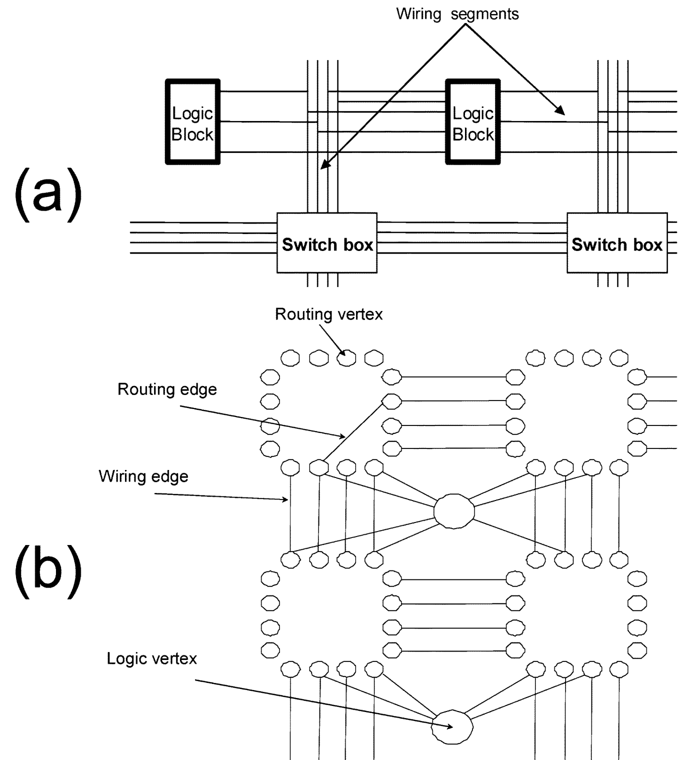

Fig. 2. (a) The generic FPGA architecture model and (b) the correspondent graph representation.

which has been described in the previous section. This representation is able to capture all the aspect of the internal architecture of modern FPGAs coming from different manufacturers. The tool is based on the dependability rules database that are described in the Sub-Section IV.A, and it embeds a clustering algorithm which is described in the Sub-Section IV.B.

A. Dependability Rules

The FPGA architectural characteristics are described within a *Dependability-Rules* database. As discussed in [5] the Dependability Rules must be enforced by a circuit implemented by a SRAM-based FPGA in order to be resilient to the effects of SEUs. In particular, the rules guarantee that any SEU affecting either the memory elements the circuit uses or the FPGAs configuration memory is not able to propagate to the circuit's outputs. The dependability rules we implemented are the results of an in-depth investigation of the effects SEUs in the memory elements in SRAM FPGA-based designs. We observed that it is possible that one and only one configuration memory bit B controls two or more routing segments. Thus a SEU affecting B modifies two or more routing segments provoking multiple effects. A detailed analysis of these effects can be found in [7]. However, when TMR hardening technique is used, further considerations should be done. A TMR circuit may include voter partition logics. A *voter partition logic* may be defined as all the logic resources (both sequential and combinational) and interconnection resources located between two voter's structures. Considering the TMR scenario described in Fig. 3, a voter partition logic consists in the logic domains $\mathrm{D_j}$ with $\mathrm{j} \in \{1,2,3\}$

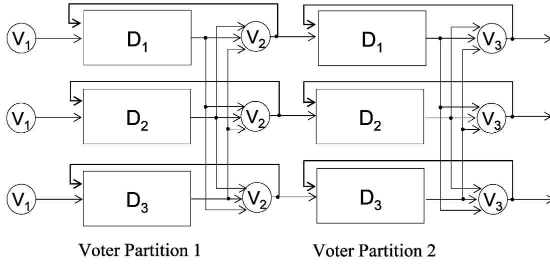

Fig. 3. The X-TMR structure scenario.

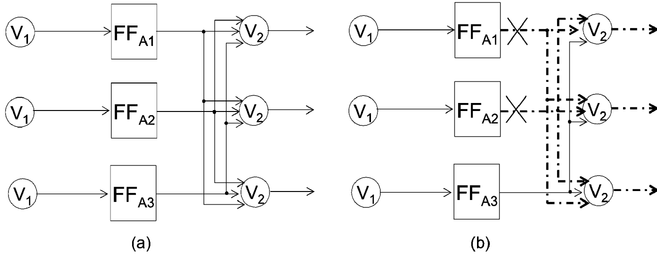

Fig. 4. SEU induced Open effect example on a TMR scenario. (a) original condition, (b) open effect.

comprised between voter structures $\mathrm{V_i}$ and $\mathrm{V_{i+1}}$. The modification that may be introduced are deeply investigated in [7] and can be grouped in three distinct cases: *Short, Open* and *Open/Short*.

These modifications may introduce critical behavior in the TMR structure illustrated in Fig. 3.

As an example, considering the TMR scenario depicted in Fig. 4(a). A SEU may induce an open effect in two signals (i.e., the output signals of the FFs A1 and A2) provoking the multiple error in all the outputs of the TMR structure.

B. The STAR-LX Tool

The STAR-LX algorithm is illustrated in Fig. 5. When analyzing a circuit, the STAR-LX tool performs three distinct phases: *reading native resources*, *redundancy clusters, rules checker*.

The reading native resources phase reads the native circuit description and create two sets: one containing the routing resources related to each voter partition logic $(\mathrm{P_i})$, and one containing the logic resources related to each TMR domain $(\mathrm{D_j})$.

The redundancy cluster phase creates two clusters that store information about any area (x, y) of the FPGA matrix, where x and y identify a row and a column of the FPGA array: $\mathrm{CS}(x,y)$ contains the information related to both the user memory and to the configuration memory controlling the logic and routing resources in (x,y); $\mathrm{HS}(x,y)$ contains the routing graph correspondent to the selected (x,y) FPGA location. The routing graph contained in each cluster $\mathrm{HS}(x,y)$ is colored according to the information related to the voter partition logic and to the TMR domain. In general, two nomenclatures are used. The first is a mark that is assigned considering that the circuit is designed according to the TMR principle, three different colors are used for all the vertices belonging to each TMR domain. The second, is an index that identifies the correspondent voter partition logic.

```
STAR_LX()
  {
   /*Reading_native_resources*/
   set_voter_partitions (Pi)
   set_tmr_domains (Dj)
   /*Redundancy-Clusters*/
   create_cluster_sets (CS(x,y), HS(x,y))
   for each voter_partition VP ∈ Pi
     for each tmr domains TD ∈ Dj
      {
       HS(x,y) = cluster_hierarchy_tree(VP, TD)
       CS(x,y) = cluster_databits(VP, TD )
               }
   /*Rules-Checker*/
   for each cluster C ∈ CS(x,y)
     for each bit location B ∈ C
       {
        UL = create_upset_list (B, HS(x,y))
        Check_dependability (UL, CS(x,y))
       }
```

Fig. 5. The STAR-LX algorithm.

Finally, the rules-checker phase analyzes the effects that may be induced by SEUs affecting the user or configuration memory cells. This analysis is performed verifying if the dependability rules are satisfied for all the possible modifications induced within the routing graph description contained in the clusters $\mathrm{HS}(x,y)$ interested by the SEU's induced modification.

C. Redundancy Cluster-Extractor

The redundancy cluster-extractor is the second phase of the STAR-LX algorithm we developed and it is dedicated to the generation of the two clusters $\mathrm{CS}(x,y)$ and $\mathrm{HS}(x,y)$ according to the internal routing and logic architecture of the SRAM-based FPGA.

This phase consists of two steps. Firstly, the function `create_cluster_sets` creates two clusters (CS and HS). Where CS consists of several bit matrices, where each bit of the matrix is organized reflecting the user and configuration memory cells architecture correspondent to the kind of SRAM-based FPGA device analyzed. Viceversa, HS consists of a routing graph architecture (the general model is depicted in the Section III) organized in several cluster, reflecting the real logic and routing organization of the FPGA device used. In detail, each cluster $\mathrm{HS}(i,j)$ describes a portion of the whole routing model of the considered FPGA device used.

Secondly, the redundancy cluster-extractor phase consists of the generation of the two sets according to the circuit that is mapped on the SRAM-based FPGA used. This step is performed by two functions executed for each voter partition VP and TMR domain TD:

1) `cluster_hierarchy_tree`(VP, TD): this function reads the resources description belonging to a voter partition VP and a TMR domain TD and generates a detailed routing graph. The architecture of the detailed routing graph consists of different marks used for each vertices and edges as explained in the Sub-Section IV.B. Furthermore, each vertex and edge belonging to a cluster $\mathrm{HS}(x,y)$ where

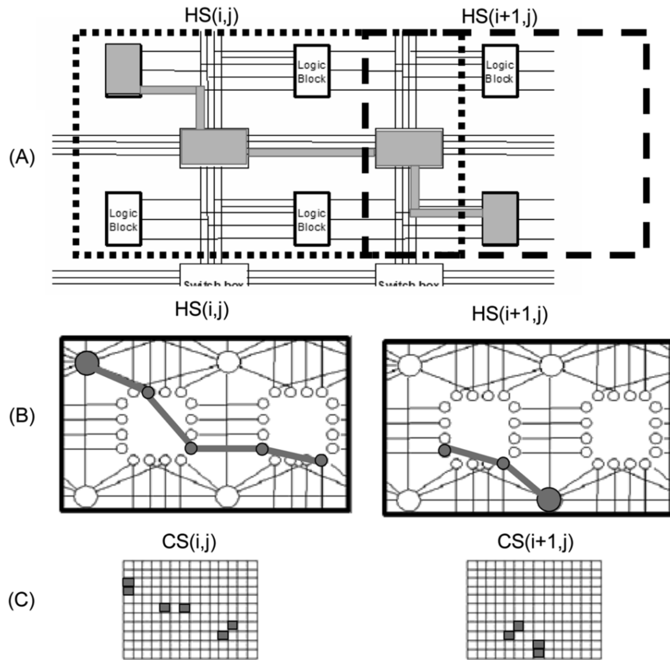

Fig. 6. An example of generation of the cluster sets HS and CS. (A) A model of an interconnection path between two logic vertices. (B) The clusters HS generated according to the circuit model, each cluster $\mathrm{HS}(i, j)$ describes a portion of the whole routing graph of the FPGA architecture considered. (C) The clusters CS generated according to the circuit model, each cluster $\mathrm{CS}(i, j)$ defines a bit matrix where each bit is marked as used (i.e., grey color) if the correspondent routing edge or logic vertex is used by the routing graph model of the implemented circuit.

x = i and y = j, is characterized by a tag that specify the correspondence with the bit contained within the data-bit clusters $\mathrm{CS}(i, j)$.

2) `cluster_databits`(VP, TD): this function reads the resources description belonging to a voter partition VP and for each location of the FPGA matrix architecture, it generates a cluster of bits CS. Each cluster consists of a bit matrix where all the bits are organized reflecting the SRAM-based FPGA configuration and user memory architecture. The function `cluster_databits` updates each cluster $\mathrm{CS}(x, y)$ by setting the bits correspondent to the user memory or configuration memory resources belonging to a voter partition and a TMR domain of the analyzed circuit.

At the end of the execution of the redundancy cluster extractor phase, the clusters CS and HS define an accurate model of the circuit that is mapped on the SRAM-based FPGA that consider both the user and configuration memory characteristics as well as the routing and logic organization of the FPGA adopted.

As illustrated in Fig. 6, the clusters HS and CS are generated according to the routing and logic topology of the circuit mapped on the FPGA architecture. Considering the circuit model illustrated in Fig. 6(A), which consists of a routing path between two logic vertices, the circuit description is read and stored within two clusters at the position i, j and $i + 1, j$. In particular, the clusters HS and CS are generated on the basis of the routing and logic architecture of the circuit implemented on the SRAM-based FPGA device. An example of the cluster generation executed by the functions `cluster_hierarchy_tree`(VP,TD) and `cluster_databits`(VP, TD) is illustrated in the Fig. 6(B) and (C) respectively.

D. The Rules-Checker

The rules-checker is the most crucial part of the developed Static Analyzer tool. It is the third phase of the STAR-LX algorithm and it analyzes if the dependability rules are satisfied for all the possible SEUs affecting the user and the configuration memory bits.

This analysis is performed by the STAR-LX algorithm functions : `create_upset_list()` and `Check_dependability()` that are executed for all the bits contained within the cluster set $\mathrm{CS}(x, y)$. In details the function `create_upset_list()` performs the following three steps:

1) It selects a bit in the position i, j within the bit-matrix of the cluster $\mathrm{CS}(x, y)$.
2) It marks the selected bit i, j as SEU sensitive.
3) It generates an upset list (UL) that consists of the modifications introduced within the routing graph architecture. These modifications depend on the kind of resource interested:
   a) *Routing*: the upset list is updated with the routing edge/edges that is/are added or deleted from the circuit routing graph model due to the modification induced by the bit-flip.
   b) *Logic*: the upset list is updated with the kind of logic components interested. In that case, the modifications include : Look-Up Tables (LUTs), Multiplexers (MUXs) or Logic Configurations (CFGs).

As soon as the upset list UL is generated, the dependability rules are checked by the function `Check_dependability()`. This function executes the following steps:

1) It updates the cluster $\mathrm{HS}(x, y)$ introducing the modification included in the upset list UL.
2) The vertices of the clusters $\mathrm{HS}(x, y)$ involved in the modification are marked as faulty.
3) The routing tree contained within the cluster set $\mathrm{HS}(x, y)$ is updated generating a SEU propagation tree that contains all the paths stemming from the vertices marked as faulty, to the first voter's structure. If the leaves of the propagation three include more than one graph coloring and only one voter partition logic index, the correspondent bit is added to the Critical SEU Locations. Indeed, when this condition is met we have that the SEU effects propagated to two or more circuit domains within the same voter partition logic, and therefore the TMR principle is no longer enforced. For each SEU considered as critical, the critical SEU locations include the position within the user or configuration memory of the SRAM-based FPGA used, as well as the indication of the kind of resources (logic or routing) and the name of the circuit component or net involved.

V. Experimental Results

In this section, we reported the results we obtained from the experimental analysis we performed with STAR-LX and the ac-

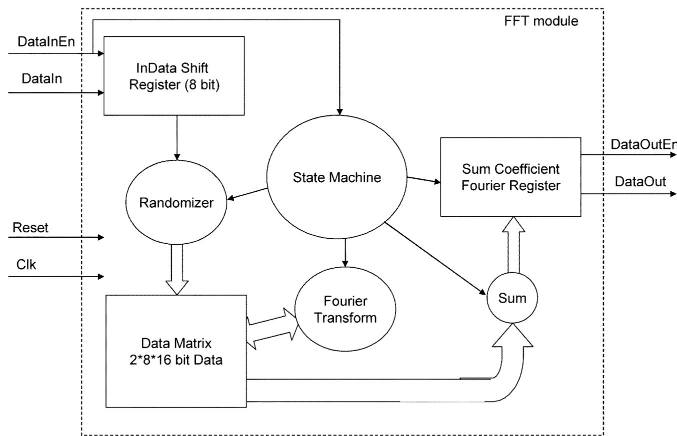

Fig. 7. The Data flow of the FFT module used as test vehicle.

celerated radiation testing setup. Firstly, we present the functionality and characteristics of the test vehicle circuit we used. Secondly, we described the radiation testing setup and we reported the comparison of the results in term of estimated error cross-section between the accelerated radiation testing and the analysis performed by STAR-LX. We focused then on the detailed results produced by STAR-LX on the selected test vehicle circuit.

A. *The Test Vehicle*

In order to analyze a real-life circuit, we selected a design that includes both combinational and sequential logic. For this purpose we adopted as test vehicle a Fast Fourier Transform (FFT) circuit, which is composed of a single working module. The rationale behind this choice is to have most of the logic in the test vehicle operating almost all the time. This means that almost any upset in the design will affect the FFT computation. Conversely, the I/O cells are seldom used (less than 5% of the time). Therefore, for the sake of this paper I/Os are not considered to be tested. The flow of the FFT-module is illustrated in Fig. 7, and it is stimulated as follows:

1) The in-data shift register is loaded from the *DataIN* by *DataInEn* inputs.
2) The Randomizer generates randomized data from the In Data Shift Register and stores it in Data Matrix.
3) A Fourier Transform is performed on the Data Matrix. The result is rewritten to the Data Matrix.
4) The sum of each result from the Fourier transform is added to the Sum of Fourier Register.
5) Step 2 and 4 is repeated until FFT has been performed on 1024 samples.
6) When all Fourier transforms have been performed, the Sum of Fourier Register is shifted out on *DataOut* output with *DataOutEn* asserted.

The FFT needs about 1 700 clock cycles for going through all the above steps.

We designed a main ad-hoc module, called *FFmatrix*, whose purpose is to test the user-logic flip-flops (FF logic) available in the device used to implement the FFT. The module is made of two identical copies of a 512-bit-long shift register. Data to the first bit of the shift register comes from an input pin and the output of the last bit goes to an output pin. The module also includes a comparator circuit, so that each bit in the two copies of the shift register is compared with each other. Any mismatch between two bits is piped to an output pin indicating the error.

TABLE I
CHARACTERISTICS OF THE TEST VEHICLE FFT MODULE

Resource	Used [#]	Available [#]
FFs	8,784	28,672
LUTs	27,705	28,672
IOBs	252	484
MULTs 18x18	84	96
GCLK	3	16

The test vector for the FFmatrix is composed of a checkerboard pattern. The outputs from each of the copies are compared with a reference output and the comparator's output is verified to be always low (no error indication).

We designed the test vehicle using the Xilinx Triple Modular Redundancy (XTMR) Tool [17], [3] that mitigates the effects induced by SEUs in both the user-logic flip-flops and the configuration memory. In details, the complete device is implemented introducing internal majority voters to prevent error accumulation in registers; moreover, all the constant values, the input, and the outputs are replicated three times.

The characteristics of the implemented circuit, in terms of used resources are illustrated in Table I, where we reported the number of Flip-Flops (FFs), Look-Up Tables (LUTs), Input/Output Blocks (IOB), embedded multipliers (MULTs) and clock signals (GCLK).

B. *The Radiation Experiment Setup*

We irradiated the test vehicle through heavy ions at the RADEF accelerator of the University of Jyväskylä in Finland (JYFL). Heavy ions tests were performed on Virtex-II XQR2V3000 devices delivered by Xilinx in 676 plastic flat packages and fabricated on a 0.15 μm/0.12 μm CMOS 8-layer metal process with epitaxial layers.

We setup the radiation platform in such a way that the DUT was continuously scrubbed during irradiation at a rate of about 5 times per second. The total number of bits to be re-configured each time was of 6 763 457.

We performed the radiation experiments using the heavy ions source of the RADEF accelerator and we measured the cross-section per bit of the FPGA configuration memory cells as function of the LET values and considering a scrub cycles of 200 ms. The experimental measurements we obtained are illustrated in Fig. 8.

Considering a LET of 10 MeV/cm^2/mg, we estimated a configuration memory bit upset cross section of 1E-8 cm^2. The reader should notice that the configuration memory bit upset cross section increase exponentially with the LET, thus for higher value of flux, the number of configuration memory bit upset for each scrub rate increase provoking Multiple Bit Upset during the same scrub cycles. For the purpose of this work, we considered the experiment with the lowest flux. This assumption is mandatory in order to have approximately only one configuration memory bit corrupted during a single

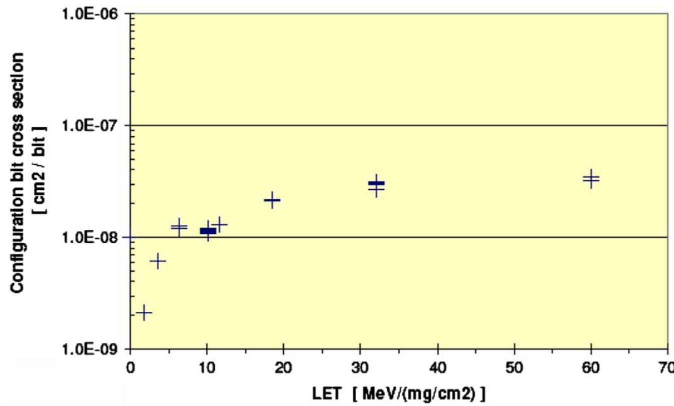

Fig. 8. Cross-Section per bit of configuration SRAM cells as a function of LET values.

TABLE II
CRITICAL SEUS IDENTIFIED BT STAR ON THE FFT MODULE

	Critical SEUs			
	CLB Resources [#]		Routing Resources [#]	
	LUT	FF	Open	Short
Critical SEUs	0	0	1,138	1,888

execution cycle. The flux was 63 ions/cm^2/s. and the device was irradiated up to a fluence of 1E5 ions/cm^2.

Such a flux corresponds to 0.88 configuration memory bit upsets for each scrub rate, and it is correspondent to a fluence of 1E5 ions/cm^2.

During testing the FFT modules using the above described set-up at the considered fluence, we measured 3 data failures during 3 different scrub cycles (although hardened according to the XTMR, the FFT device produced unexpected outputs), therefore if we considered the whole device area in the device under test (DUT) as $\mathrm{DUT_{AREA}}$, the measured error cross section for the FFT module is of $3\mathrm{E}-5\pm1.7\mathrm{E}-5$ cm^2.

C. The Experimental Validation

In order to validate the developed STAR-LX tool, we compared the error cross section we measured during the radiation testing experiments with the error cross section computed on the basis of the number of critical SEUs reported by the STAR-LX analysis.

The results we obtained by the analysis performed with STAR-LX are reported in Table II, where we indicated the number of Critical SEU that corrupt the protection offered by the XTMR tool, ordered by the kind of resource interested: Control Logic Block (Flip-Flops and Look-Up Tables) and Routing Resources (Open and Short effects). As reported in Table II, the total number of configuration memory bits predicted as critical if affected by SEUs is 3 026. Being the configuration memory cross section equal to 1E-8 cm^2/bit we can compute the STAR-LX predicted error cross section as: 1E-8 cm^2/bit $\times$ 3 026 bit = 3.026 E-5 cm^2.

The STAR-LX predicted error cross section closely matches the experimental error cross section obtained by analyzing the results gathered though radiation testing.

D. The Detailed STAR-LX Analysis Results

In this section, we reported the detailed report of the analysis performed by STAR-LX on the test vehicle circuit.

The STAR-LX tool took about 130 mins in order to perform the complete analysis of the implemented FFT circuit. None critical SEU affecting the logic resources has been observed, confirming that the logic circuitry has been effectively hardened by the XTMR tool. All the critical SEUs reported involve routing resources. The STAR-LX tool reported for these critical SEUs a detailed classification that is summarized in the Table III, where the SEU effects have been classified according to the produced routing modification (Multiple Open or Short) and in relation to the involved signals. The SEUs effect are classified as:

- *TMR Failure*: the SEU provokes an error within the same voter partition logic according to the dependability rules defined in the Sub-Section IV.A.
- *Warning One Domain*: the SEU provokes an error within the same domain of the TMR architecture. In that case, two different signals of the same TMR domain are involved in a failure but its effect is not propagated to the circuit outputs, since it is protected by the first voter circuitry stemming from the SEU effect location.
- *Warning Different Partition*: the SEU provokes an error within two different domains of the TMR architecture, but in two different voter partition logic. Thus the SEU effects is protected by the voter circuitry of each voter partition logic.
- *TMR Warning Same Signal*: the SEU provokes an error within two different domains of the TMR architecture and within the same voter partition logic, but it is related to the same signal. This effect happens only in the case of the short effect.

As the reader can notice, most of the SEUs are producing a short effects within the routing architecture of different voter partition logics. In order to analyze the most critical module of the implemented circuit, the STAR-LX tool generates a report that identify the involved interconnections. The classification result is reported in Table IV where we classified the Critical SEU effects considering the interconnections involved and grouped into the logic modules of the test vehicle circuit as described in the Sub-Section V.A.

Furthermore, it is possible that a critical SEU affects interconnections of the same module. In particular, in the case of the State_Machine module the number of critical SEU is particularly high. This phenomena is due to the lower protection level that is offered by the XTMR feedback voters in the case of finite state machine (FSM) circuitry. However in the case of the other modules, the number of critical SEUs observed is lower than 9.

The major criticalities have been observed between the interconnections of the modules State_Machine and Fourier_Transform. The reason of this effects should be led back to the control rule of the State_Machine module within the circuit, since the interconnections of the State_Machine module span over the other FFT modules, thus increasing the probability of having a multiple errors between two interconnections of different modules.

TABLE III
ROUTING SEU EFFECTS CLASSIFIED BY THE STAR-LX TOOL

	SEU effects Classification [#]	
Signal Classification	Multiple Open	Short
TMR Failure	1,138	1,888
Warning One Domain	107	1,484
Warning Different Partition	91	2,945
TMR Warning Same Signal	0	52

TABLE IV
STAR-LX DETAILED CLASSIFICATION OF THE CRITICAL SEU EFFECTS

	Design Net Critical SEU Effects [#]					
	Data_Register	Randomizer	Data_Matrix	State_Machine	Fourier_Transform	Sum_Coefficient
Data_Register	3	4	484	301	112	42
Randomizer	-	2	4	24	42	4
Data_Matrix	-	-	9	437	202	198
State_Machine	-	-	-	301	631	210
Fourier_Transform	-	-	-	-	6	8
Sum_Coefficient	-	-	-	-	-	2

VI. CONCLUSION

In this paper we proposed a validation of the approach developed in [7] through accelerated radiation testing. We presented the Static Analyzer tool (STAR-LX) describing its characteristics from the algorithm and executions point of views. The proposed STAR-LX tool is able to analyze the dependability degree of an industrial-like design when implemented on a SRAM-based FPGA. We validated the proposed tool by means of radiation testing experiments. In details, we presented the results of two experiments. The first one considered the estimation of the error cross section of a test vehicle using radiation testing. The second consisted in predicting the error cross section through the STAR-LX tool developed in [7]. Experimental results demonstrated that the STAR-LX predicted error cross section closely matches the experimentally measured one, thus confirming the goodness of the STAR-LX approach. Moreover, the experiments showed the capability of STAR-LX of establishing the correlation between SEUs and circuit resources, thus providing a valuable feedback to designers.

REFERENCES

[1] M. Ceschia, A. Paccagnella, S.-C. Lee, C. Wan, M. Bellato, M. Menichelli, A. Papi, A. Kaminski, and J. Wyss, "Ion beam testing of ALTERA APEX FPGAs," in *Proc. Nuclear and Space Radiation Effects Conf. Data Workshop*, Phoenix, AZ, Jul. 2002, pp. 45–50.

[2] E. Fuller, M. Caffrey, P. Blain, C. Carmichael, N. Khalsa, and A. Salazar, "Radiation test results of the Virtex FPGA and ZBT SRAM for space based reconfigurable computing," presented at the Military and Aerospace Applications Programmable Devices and Technologies Conf., Laurel, MD, Sep. 1999.

[3] *"X-TMR Tool User Guide,"* ug156.

[4] H. Quinn, "Limitations in Single-Chip Triple Modular Redundancy," presented at the Xilinx Status Meeting, Los Alamos, NM, Jun. 2007.

[5] L. Sterpone and M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs," *IEEE Trans. Comput.*, vol. 55, no. 6, pp. 732–744, Jun. 2006.

[6] *Correcting Single-Event Upsets Through Virtex Partial Reconfiguration*, XAPP216, 2000, Xilinx Application Notes.

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

[8] L. Sterpone, M. Violante, and S. Rezgui, "An analysis based on fault injection of hardening techniques for SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 4, pp. 2054–2059, Aug. 2006.

[9] R. Katz, K. LaBel, J. J. Wang, B. Cronquist, R. Koga, S. Penzin, and G. Swift, "radiation effects on current field programmable technologies," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 6, pp. 1945–1956, Dec. 1997.

[10] R. Katz, K. Label, J. J. Wang, B. Cronquist, R. Koga, S. Penzin, and G. Swift, "Radiation effects on current field programmable technologies," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 6, pp. 1945–1956, Dec. 1997.

[11] D. K. Pradhan, *Fault-Tolerant Computer System Design*. Englewood Cliffs, NJ: Prentice Hall, ISBN 0-13-057887-8.

[12] F. Lima, C. Carmichael, J. Fabula, R. Padovani, and R. Reis, "A fault injection analysis of virtex FPGA TMR design methodology," in *Proc. IEEE European Conf. Radiation and its Effect on Components and Systems*, Sep. 2001, pp. 275–282.

[13] P. Bernardi, M. S. Reorda, L. Sterpone, and M. Violante, "On the evaluation of SEUs sensitiveness in SRAM-based FPGAs," in *Proc. 10th IEEE On-line Testing Symp.*, Funchal, Madeira Island, Portugal, 2004, pp. 115–120.

[14] M. Alderighi, S. D'Angelo, M. Mancini, and G. R. Sechi, "A fault injection tool for SRAM-based FPGAs," in *Proc. 9th IEEE On-Line Testing Symp.*, Kos Island, Greece, 2003, pp. 129–133.

[15] P. Sundararajan and B. Blodget, "Estimation of mean time between failure caused by single event upset," Xilinx Application Notes, Jan. 2005, XAPP559.

[16] G. Asadi and M. B. Tahoori, "An analytical approach for soft error rate estimation of SRAM-based FPGAs," presented at the Military and Aerospace Applications Programmable Logic Devices Conf., Sep. 2004.

[17] C. Carmichael, "Triple module redundancy design techniques for virtex FPGAs," Xilinx Application Notes, Nov. 2001, XAPP197.