

# A New Method of Modeling Radiation-Induced Soft Errors in FPGA by Architecture Analysis

Jan Pospisil, Tomas Vanat, Jan Schmidt

Department of Digital Design

Czech Technical University in Prague, Faculty of Information Technology

Prague, Czech Republic

Email: jan.pospisil@fit.cvut.cz, tomas.vanat@fit.cvut.cz, jan.schmidt@fit.cvut.cz

**Abstract**—Recently, FPGA devices are more often used in applications demanding dependability and safety. These FPGAs are, nevertheless, manufactured using CMOS technology with SRAM memory cells, which are prone to ionizing radiation. In our work we propose a method of modeling the behavior of FPGA in radiation harsh environment based on parameters obtained from experiments on real hardware. The proposed method utilizes academic toolchain VPR. By modifications of this toolchain, and from SEU characteristics gathered from "in vivo" experiments, a modeling and simulation platform for future designs can be constructed, with close-to-reality results.

**Keywords**—FPGA, ionizing radiation, single event upsets, reliability modeling, FPGA architecture, VPR

## I. INTRODUCTION

The majority of recent FPGAs use SRAM memory for both configuration and data storage. In a radiation harsh environment, these memory cells and other structures can be affected by ionizing radiation. The change of their state can cause a user design malfunction – so called "soft errors" or SEE (single event effects). High energy particle can temporarily create conductive paths in silicon and alter the value of a signal. If such a change occurs inside the memory cell, the stored value can be altered permanently – this is called SEU (single event upset).

SRAM cells are used not only by a user design; the biggest portion of this memory is used for the configuration data. Thus the biggest threat of SEU is in SRAM configuration memory. As a large portion of this memory is used for interconnection configuration, an SEU can cause not only data change, but also a structural change of the circuit.

There are several design techniques to build a dependable application on an unreliable hardware. To validate the application in practice, we must predict the dependability parameters, and furthermore to verify that no fault can lead to dangerous behavior – whatever "dangerous" may mean in the given situation.

The design of a dependable application must always balance dependability and cost – in the terms of silicon area, power consumption or performance. Therefore, design space exploration with respect to dependability techniques is important and shall be supported by the design environment.

There are two main ways to estimate dependability parameters of a design. The first one is the accelerated life test (ALT), which is based on monitoring the device under increased level

of radiation. If the appropriate energy spectrum of particles is used, then this method gives very accurate results. The problem is that the method is very expensive and not easily accessible. The source of accelerated particles is usually a big and a lot of energy consuming facility. The test must be done for each design or even each variant of the design and requires a full implementation of the design in silicon.

The alternative to ALT is to use a simulation model. To create an accurate model, it is necessary to have detailed physical structure of the tested hardware. With FPGAs, this is a problem, as the structure of the devices cannot be disclosed. As a consequence, the majority of available models are only approximate ones. To overcome this deficiency, we use a relatively low-level model which can be tuned to match the results of radiation test of specially prepared calibration designs. This way, we still need a high-energy particles source, but only once for the calibration of the model.

A method to classify fault behavior by an arbitrary predicate under any combinational fault model have been published [1]. This method, however, depends on detailed model of fault behavior. Using the proposed technique, we will be able to provide a calibrated model to this method and more realistic estimates of dependability and safety of the design in question.

In this paper we propose to obtain a model of the implemented design useful for fault analysis. The model results from duplicating the back-end (physical design) tool of the standard flow, in this case by an open tool. It works on a global FPGA model, which can mimic existing devices and which can be annotated by information necessary for the fault analysis.

We will describe the global FPGA model first, then its use during a design analysis. The model and the entire method will not be useful without a calibration procedure, which will also be presented.

## II. PROPOSED METHOD

Given a design and a target FPGA device, the purpose of our work is to predict the behavior of the implemented design under radiation exposure. The main device for this modeling is a netlist of the implemented design, with all parts annotated by information of faults that can occur at that part.

To construct such a model, a suitable detailed model of the target FPGA architecture is necessary, together with a physical design tool working on that model. The constructed model is then calibrated by ALT. One calibrated model thus covers all design on the target architecture.

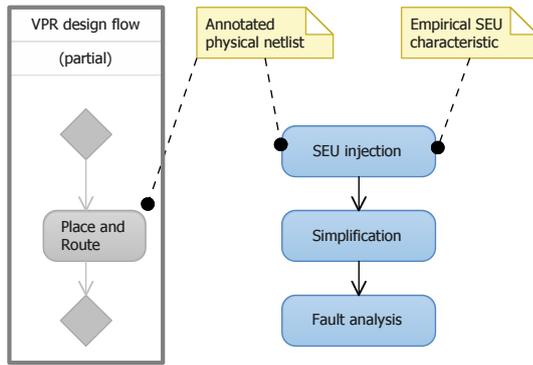


Fig. 1. Modified VPR design flow - proposed method

### A. The FPGA model

Three areas of the FPGA architecture must be covered: data storage (RAMs and flip-flops), configurable logic blocks and interconnection. The design model of data storage commonly corresponds to the real implementation, and therefore can easily be modeled. In recent devices, configurable logic blocks are composed of look-up tables (LUTs) and multiplexers. Their structure – and in some cases also their configuration strings – is often published. The difficult part is therefore the interconnection.

The interconnection architecture has fairly stabilized in recent FPGAs. It has usually two levels of hierarchy, with local interconnection between adjacent logic blocks and global interconnection over the entire chip. There is also tendency to simplify the interconnection structure (unidirectional wires, depopulated wires [2]). Simpler interconnect is simpler to use in automated tools and simpler to model. Incidentally, its fault models are also easier to construct and to simulate. The long chains of passive switches known from past devices disappeared. Also, the selection of inputs for logic blocks is done nearly uniformly, with only slight differences in the multiplexer implementation.

The modeling process is based on the VPR tool (Versatile Packing Placement and Routing), which is a part of VTR suite (Verilog to Routing) [3]. This tool allows user to create custom FPGA architecture and perform timing-driven packing, placement and routing on it. Both the VPR tool and any format used in it are open, so the whole toolchain can be modified or only a part of it can be used.

### B. Usage of the model

For each design, the (commercial) tools for the target architecture are run to obtain an implementation, which will be finally used to program the target device. Parallel to this, we need to construct a model of the implementation for fault analysis by VPR. The more information we carry from the original implementation, the more accurate the fault analysis model will be. The necessary minimum is the netlist (the result of HDL synthesis). Most commercial tools can give us the physical netlist (the result of technology mapping) and even placement. All this information can be used to constrain VPR and to make the model more relevant.

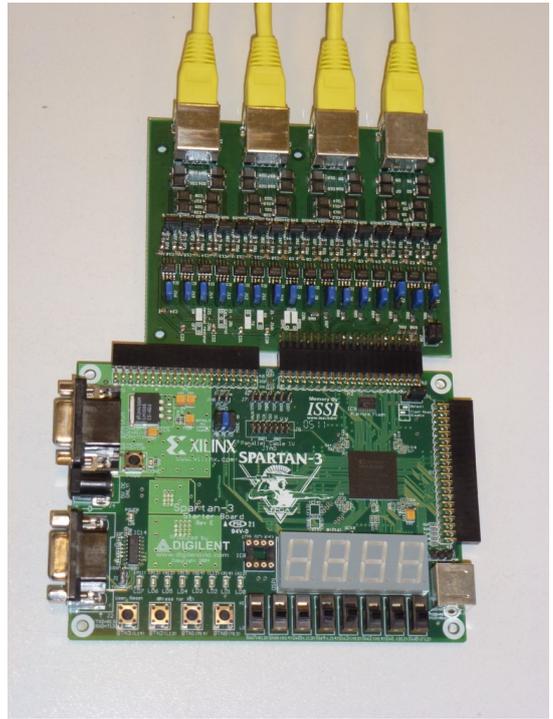


Fig. 2. Used test board with interface board attached

Given an annotated implementation of a net, alterations of configuration memory content in elements constituting the signal path can be transformed into fault models analyzable by standard methods. SEU in a LUT configuration memory are transformed into changes in logic function performed by that LUT. Before the resulting annotated netlist of the design can be used for fault analysis, it should be simplified. The final design flow can be seen in Figure 1.

The final fault analysis can be done using Monte Carlo fault simulation, Satisfiability-based methods [1] and other methods.

The relevance of the analysis, of course, depends on the relevance of the fault analysis model with respect to the actual implementation in the target device, despite the differences in FPGA architecture model and physical design procedure. This must be ensured by the calibration of the modeling process.

### C. Radiation test verification and FPGA model calibration

For verification of the FPGA model, we will correlate data from the ALT of real device under accelerated particle flux and those predicted by our model, for multiple designs. The correlation designs will be specially crafted to tell what elements in the FPGA model shall be altered.

1) *Used hardware:* Our ALT device is based on the Digilent Spartan3 Starter Board, equipped with Xilinx Spartan3 XC3S200 chip, made by 90nm technology. It's a little bit older technology, but it is easily available and relatively cheap, for the case that something goes wrong. This FPGA will be exposed to accelerated particle flux with known intensity and energy spectrum.

A special communication board will be attached to the Spartan3 Starter Board to transfer acquired test data from

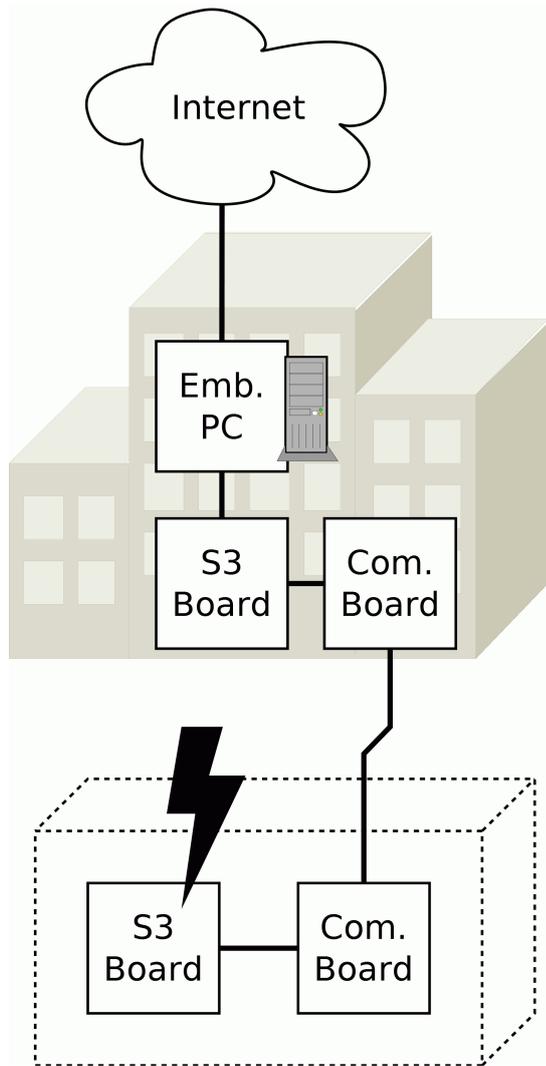


Fig. 3. A diagram of the ALT system

the tested device to the evaluation device in separate location (approx. 40 meters away), shielded from the ionizing radiation. The communication board contains sixteen differential line drivers with selectable data flow direction. These drivers are grouped into four groups by four drivers and each group is connected to one RJ-45 connector. Two signals are connected to the PROG and DONE control signals of the FPGA and the rest is connected to the general purpose IO pins. Figure 2 shows the Spartan3 Starter Board with attached communication board.

The device used for evaluating the test data is based on the same hardware, but it has loaded a different design and it sends reports via a serial port into a PC. A diagram of the entire system is in Figure 3.

2) *Basic calibration design:* In first experiments, we plan to verify whether we are able to evoke the necessary fault rates in entire chip and how their frequency depends on the intensity of particle flux. For this, we need a circuit similar to that used by iRoC Technologies in their radiation tests [4]. Their array of multipliers, however, is quite slow for testing all of the combinations and – what is worse – it is a purely

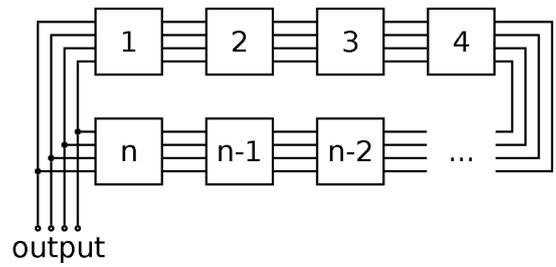


Fig. 4. The architecture of the calibration design

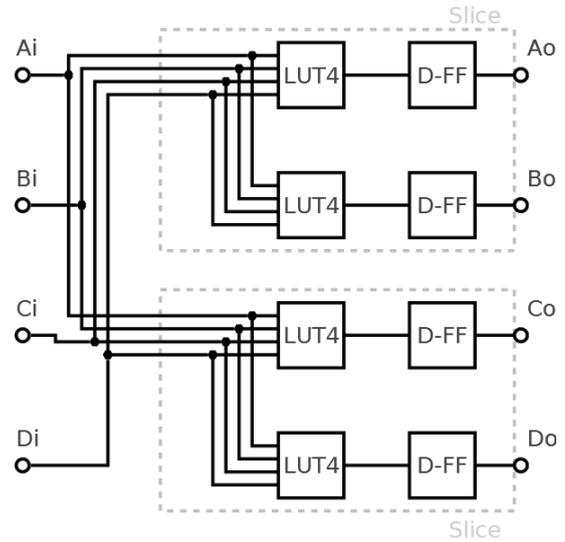


Fig. 5. A single stage of the calibration design

combinatorial circuit, which can detect only the faults in the configuration memory. There are also other places, which can be influenced by the single event effects – the data D flip-flops.

The design we use is a long circled pipeline (Figure 4), consisting of many identical stages that fills up the whole chip. Their function is to perform a simple conversion between two four bit wide binary codes (Figure 5). The conversion is chosen so that it uses full capacity of four input LUTs. If any of the bits in these LUTs is changed, the conversion would be wrong. The conversion function is symmetric, so after two conversions (two stages of pipeline), the output is same as input, two clock cycles delayed.

The pipeline has no input. It is preloaded with data upon reset and these data stay in it until the next reset. It has only one output, through which the data are sent to the control device, where they are evaluated. When some failure occurs, the data is changed and from the behavior of the circuit, we can determine whether the error is in configuration memory or in a D flip-flop. When the faulty behavior persists after device reset, performed automatically by the control part, the error is located in the configuration memory. Then it is necessary to reload the bitstream, which can also be done remotely and automatically.

The fault behavior of the device can be further analyzed off-line. This analysis will distinguish between stuck-at faults at the inputs of logic blocks or in the interconnection and functional faults in the LUTs. Due to the alternating nature of

the code, most stuck-opens can be distinguished as such.

The design is easily applicable to any FPGA size and architecture. When 6-input LUTs are used, the only difference is to increase the data width of the pipeline and variable length of the pipeline fits to all FPGA sizes.

3) *Calibration procedure:* To test and finally ensure match between prediction and reality, we need multiple designs, both in the model and in the real device. The biggest problem is the role of interconnect and the faults in it.

The first step is a design with identical logic, but different interconnection. Both VPR and industrial tools allow to manipulate placement. A family of designs can be constructed starting with optimum placement (and hence the minimum of interconnection). Further members can be derived by making the placement less optimal, for instance by random exchanges of placement locations. Within this family, all members have the same probability of SEU in logic blocks, but increasing probability of SEU in the interconnection, which will be used to alter the model accordingly (wire lengths and numbers, number of multiplexer stages, multiplexer control style).

A step further in the same direction will involve manipulation of the Rent exponent [5] of the circuit. The pipeline can be branched and joined again using XOR operation without loss information. This way almost unlimited values of Rent exponent can be achieved, leading to full utilization of routing resources on the FPGA chip.

Finally, parts of the FPGA device not covered by these designs but required by applications must be tried: carry chains, memories, multipliers, clock sources. It is not difficult to focus on one type of resource, provided enough time on the accelerated ionizing particle source is available.

### III. FUTURE WORK

The problem of SEU and radiation tolerant programmable systems is nowadays very actual in the ALICE ITS project [6]. In this project thousands of pixel detectors which produces a large amount of data during operation is used. So far, processing data from these detectors was performed by specialized ASIC chips, but the next version is intended to be constructed on a reconfigurable hardware. This problem is being solved in Nuclear Physics Institute in Řež. Our models and testing

designs should be very useful there. We have already arranged collaboration and prepared first accelerated life tests with our testing design, which is very similar to the task, required from the FPGAs in the ALICE detector.

### IV. CONCLUSIONS

We have prepared a new method of creating a model for testing designs using programmable devices (FPGAs) against single event upsets and other radiation-induced faults. Currently, the testing device is nearly ready and the first accelerated life test (ALT) under the high-energy particle radiation is being prepared in collaboration with the Nuclear Physics Institute in Řež. After some data will be acquired from ALT testing, the VPR model will be calibrated and improved. This model can be then used for testing SEU immunity of dependable devices based on FPGA.

### ACKNOWLEDGEMENTS

Research described in the paper is supervised by doc. Ing. Hana Kubátová CSc. and Ing. Jan Schmidt, Ph.D., FIT CTU in Prague and supported by the CTU SGS grant No. SGS13/101/OHK3/1T/18.

### REFERENCES

- [1] Schmidt, J. and Fišer, P. and Balcárek, J., "The Influence of Implementation Technology on Dependability Parameters," in *Digital System Design (DSD), 2012 15th Euromicro Conference on*, 2012, pp. 368–373.
- [2] Betz, V. and Rose, J. and Marquardt, A., *Architecture and CAD for Deep-Submicron FPGAs*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 1999, vol. 497.
- [3] Rose, Jonathan and Luu, Jason and Yu, Chi Wai and Densmore, Opal and Goeders, Jeffrey and Somerville, Andrew and Kent, Kenneth B. and Jamieson, Peter and Anderson, Jason, "The VTR project: architecture and CAD for FPGAs from verilog to routing," in *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays*, ser. FPGA '12. New York, NY, USA: ACM, 2012, pp. 77–86. [Online]. Available: <http://doi.acm.org/10.1145/2145694.2145708>
- [4] iRoC Technologies, "Radiation Results of the SER Test of Actel, Xilinx and Altera FPGA instances," 2004, <http://www.actel.com/documents/RadResultsIROCreport.pdf>.
- [5] Landman, B.S. and Russo, Roy L., "On a Pin Versus Block Relationship For Partitions of Logic Graphs," *Computers, IEEE Transactions on*, vol. C-20, no. 12, pp. 1469–1479, 1971.
- [6] Rossegger, S., "The ALICE ITS upgrade," in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*, 2011, pp. 513–517.