Reliability Analysis of Reconfiguration Controller for FPGA–Based Fault Tolerant Systems: Case Study

Richard Panek, Jakub Lojda, Jakub Podivinsky, Zdenek Kotasek
Faculty of Information Technology, Brno University of Technology, Centre of Excellence IT4Innovations
Bozetechova 2, 612 66 Brno, Czech Republic
Email: {ipanek, ilojda, ipodivinsky, kotasek}@fit.vutbr.cz

***Abstract*—This paper deals with a reliability analysis of a reconfiguration controller which can be a component of a fault-tolerant control system. This controller is designed for an FPGA to be capable of using partial dynamic reconfiguration of the FPGA to mitigate potential faults in the FPGA's configuration memory. These faults, which are called SEUs, can be induced by radiation effects. Therefore, fault tolerance measurement or estimation is very important for designing circuits for critical environments. Thus, the reliability of the reconfiguration controller itself is significant; therefore the Fault Tolerance ESTimation (FT-EST) framework is used for reliability evaluation, which is procured by the discovery of a number of critical configuration bits. Two approaches are used and compared: evaluations of used LUT only, and evaluations of all configuration bits. We ascertained a 20x reduction in time consumption at the expense of a proportional decrease in the amount of critical configuration bits discovered. The obtained results are nearly equivalent.**

***Keywords*—Fault-Tolerant, Partial Dynamic Reconfiguration Controller, Fault Tolerance Property Estimation, FT-EST.**

I. Introduction

Space or critical application system developers take advantage of Field Programmable Gate Arrays (FPGAs) because of their high performance, parallel computation ability, power consumption reduction and possibility to change their configuration, which is called reconfiguration. However, space is a very harsh environment for commonly used SRAM-based FPGAs, of which the configuration memory, where the information about the customer circuit is stored, is assembled from Static Random Access Memory (SRAM) cells. These cells are sensitive to Single-Event Upsets (SEUs) which are induced by radiation. The SEUs cause memory bit flipping which can mean customer circuit damage when the configuration memory cell used is affected. The intensity of these effects depend on the FPGA type and the energy of the hit particles [1].

Many approaches or techniques are developed for the utilisation of SRAM-based FPGAs in harsh environments [2] and these are called Fault Tolerance (FT) [3] techniques. The FT is based on the application of unreliable devices, their redundancy and additional components to attain more reliable final systems, which may be called Fault-tolerant Control Systems (FTCSs) [4]. The FTCS is assembled from (1) reconfigurable controller, which is FPGA in our case, (2) Fault Detection and Diagnostics (FDD) and (3) Reconfiguration Controller (RC) which is responsible for the repair of detected faults by FDD. Reconfiguration is a pivotal property of FPGAs, which enables fixing injured configuration memory. Partial Dynamic Reconfiguration (PDR) in particular is advantageous because during reconfiguration of the damaged part of the configuration memory, the remaining undamaged part of the circuit stays in operation without interruption.

In testing the reliability of the FTCS waiting for a fault to appear naturally would be unbearably time-consuming or impossible. For this purpose, there are techniques for injecting faults artificially into the system. The most realistic method is a radiation test based on the insertion of Device Under Test, which in our case is an FPGA with FTCS, in front of the radiation emitter and observation of the impact. Radiation tests can be used for FPGAs or processor FT testing, as presented in [5]. Because the demands on radiation testing are considerable, it is advantageous to use fault simulation for FT testing. Moreover, the authors of [6] compared both approaches and show that both test methods produce comparable results. In the research [7], a simulation-based injection technique, which is able to simulate the impacts of faults on designs in both RTL and netlist levels of abstraction, is presented. The authors of [8] present an accurate SEU simulation method that combines simulation with topological analysis of the design. The authors of [9] utilise simulation in combination with functional verification for the purposes of fault impact evaluation. The modelled faults are injected during the verification run in the simulation environment. The technique utilising JTAG for observation and to modify signals in the design is presented in [10]. Another approach is described in [11] where the authors insert some additional logic gates into an original VHDL-written design to simulate several fault models. In the following techniques, faults are injected directly into an FPGA and furthermore, these techniques do not require the modification of the original design. The technique described in [12] accelerates the speed of the design evaluation by moving the control fully to the FPGA. Another HW-based evaluation method, also based on FPGA technology, is presented by the authors of [13]. Furthermore, the authors of [14] describe a platform called Flipper, which works with two interconnected FPGA boards.

This paper is organised as follows: the case study of this work, which is the reliability test of the RC is presented in Section II. The experimental results are summarised in Section III. Finally, Section IV concludes this paper.

II. Case study

The main objective of this work is RC reliability estimation with regard to FT. Therefore, the FT-EST [15] is utilised for FT evaluation. This research is important as a base for the following hardening in the reliability of the RC itself. The

978-1-7281-6083-2/20/$31.00 ©2020 IEEE

RC which is taken into consideration is a Generic Partial Dynamic Reconfiguration Controller (GPDRC) [16], which is developed and used by our research group. Moreover, there are some possibilities for other improvements and uses. The evaluation platform for this work is an ML506 board produced by Xilinx with Virtex-5 FPGA (XC5VSX50T). Their development software for FPGAs, such as ISE Design Suite, PlanAhead and FPGA editor are also utilised for design and design diagnostics.

A. FT-EST adaptation

The FT-EST framework was originally intended for the automation of FT design, however it suits all the requirements to evaluate the RC. As an extra bonus, because of its focus on FT design automation, the framework requires very little user interaction during the testing after the test-bench is prepared. Thus, for its utilisation to evaluate the RC which needs to keep its internal state for correct reconfiguration control, feedback between the golden unit and the Input Generation Unit (IGU) is used. This feedback provides correct IGU function, which must react with stimuli generation to the current RC state, which in this case is the unit under test. In addition, the IGU itself must be prepared to generate eligible stimuli for RC testing, such as fault detection emulation, subsequent notification to the RC of the part of circuit that requires reconfiguration, and golden bitstream memory emulation. The IGU is waiting for a data request from the RC that comes through the feedback between the Golden Unit and the IGU. It then sends the respective randomly generated data to the RC (i.e. the Golden Bitstream).

B. GPDRC—implementation analysis

The way the GPDRC can be implemented in the FPGA is shown in Figure 1 which is a representation exported from the FPGA editor. Here, a cut-out of the Virtex-5 FPGA area, one row and 11 columns specifically, can be seen. In each column, there are a Switch Matrix (white rectangle on the left side of column), which is responsible for functional block interconnection, and functional blocks (e.g. Configurable Logic Blocks [CLBs], Digital Signal Processors [DSPs] or Block Random Access Memories [BRAMs]). In every CLB there are two SLICEs with Lookup Table (LUT) and Flip-Flop (FF). For the completeness, each column is composed of frames, the number of which differs based on the type of functional blocks in it. In one column, there are either (i) 36 frames for the CLB, or (ii) 28 frames for the DSP, or (iii) 30 frames for the BRAM. Every frame is 1312 bits long because it is composed of 41 32-bit configuration words. The number of frames in the column is important to provide the necessary amount of SEU injections to ensure exhaustive reliability verification. Every bit in the configuration memory can be affected by an SEU. Specific previous values are from [17].

Thus, in Figure 1 there are 2 columns with DSPs (from the left, the third and the sixth columns), one column with Block RAMs (the ninth column) and 8 columns with CLBs (the remaining columns). Moreover, the utilisation of particular functional blocks and how they are routed by switch matrices and lines (cyan colour) can be seen. Obviously, the utilisation of resources is not uniform. The implementation of the GPDRC does not use DSP blocks, so from these columns only some switch matrices are used for the connection of neighbouring resources. Then, there is the seldom used SLICE in the top left and the bottom right corner. The utilisation of another component is shown in Table I, which is the statistic from the PlanAhead implementation. There are mainly LUTs, Flip-Flops, SLICEs and BRAMs. Obviously, only approximately 25% of the resources are used. This implies that most of the configuration bits are unused and therefore, faults which strike these bits might not impact the correct GPDRC functioning.

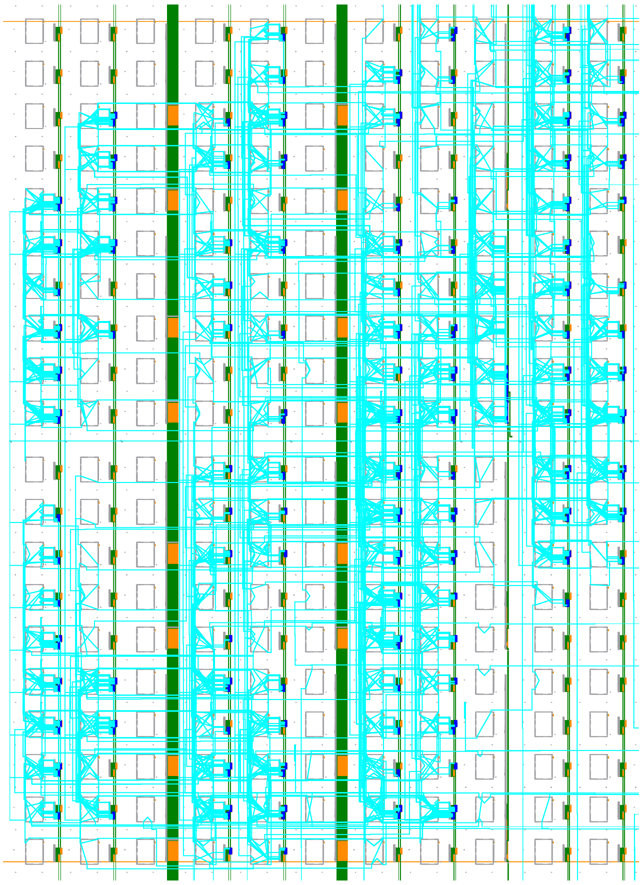

Figure 1: Routed GPDRC visualisation by FPGA editor.

TABLE I: Resource utilisation of GPDRC implementation.

Site Type	Available	Required	% Utilisation
LUT	1280	328	26
FD_LD	1280	288	23
SLICEL	160	44	28
SLICEM	160	44	28
RAMBFIFO36	4	1	25

C. Comparison of alternative approaches to fault injection

The secondary objective of this work is to compare two fault injection approaches. The first is an overall evaluation of all possible SEUs. Thus, the ascertainment of the impact of any potential fault is necessary. Another approach is based on the assumption that only faults in used components can cause an entire system failure. Therefore, SEU injection into used configuration memory bits is sufficient for a system reliability estimation. The number of all potential configuration bits is summarised in Table II. Therefore, about half a million fault injection runs and impact assessments are required for a complete reliability analysis. This number is deduced from the amount of FPGA area occupied by the RC.

TABLE II: Number of SEU injections for complete evaluation.

Type	Columns	Frames	Bits for injection
SLICE	8	36	377,856
BRAM	1	30	39,360
DSP	2	28	73,472
		Total:	490,688

For the second approach, a tool for searching used LUTs based on RapidSmith is important. Thus, LUTs are deemed as the main used part of the design because most functions are implemented by them. This tool discovers that the GPDRC implementation uses only 23,616 bits of configuration bits for LUTs. The number of utilised configuration bits in particular columns is shown in Table III. As expected, the used LUTs are found only in columns with SLICEs. Therefore, the percentage representation is related to the number of bits in columns with SLICEs and the total bits in all these columns.

TABLE III: Used LUT bits in particular columns.

Column No.	1	2	4	5	7
Used LUTs	5568	1792	1856	2496	3328
	11.79%	3.79%	3.93%	5.28%	7.05%

Column No.	8	10	11	Total
Used LUTs	1472	4480	2624	23616
	3.12%	9.49%	5.56%	6.25%

When comparing the number of only used LUTs with all the bits in the selected implementation area, the saving is 95.19%. Therefore, the required time for reliability estimation is about 5% of time for the complete evaluation. Nevertheless, there might be a difference between slow reliability evaluation and fast estimation that will be assessed in experiments.

III. Results of the Experiments

The numbers of critical bits are the values obtained from the FT-EST experiments of this work that determine the number of bits in which failure may occur. It is necessary to relate these numbers to the utilised area of the FPGA or more precisely, to the utilised FPGA configuration memory bits. In most cases, faults in unused configuration memory have no impact on reliability, which is therefore calculated from the ascertained critical bits and FPGA resource usage statistics.

A. Absolute critical bit count

The first results, the number of GPDRC critical bits obtained due to FT-EST are shown in Table IV. Moreover, in the addition to the summary of critical bits, there is the division of these bits into corresponding types of resources and specific columns. In the first column of this table there is a summarising result of the search for critical bits only in used LUTs by design. Thus only about 5.5% of the used configuration bits represent used LUTs by design. Furthermore, this value also represents the estimation of reliability, because it tells us that 5.5% of faults result in system failure. That is because this value is related to utilisation of the FPGA, unlike the rest of this table where the values are related to the overall area where circuit can be, which is the same with the number of overall testing bits. In the second part of this table, critical bit division into occurrence at BRAMs, DSPs or SLICEs are discovered. As can be seen, DSP blocks, which are unused because only interconnection links are used there, have only a negligible amount of critical bits. Then the relative amount of critical bits in BRAM and SLICE are the same. Eventually, critical bit distribution into particular SLICE columns is shown. When these values are compared with the amount of used LUTs in Table III, obviously there is no coherence between them. Thus, the numbers of critical bits are independent of the amount of used LUTs in particular SLICE columns.

B. Reliability calculation improvement

At first view, the reliability appears that only about 2% of the bits are critical from the complete test, or 5.5% from the accelerated used LUT-only test. However, the utilisation of resources which were summarised in Table I is not embraced. Thus, the utilisation of particular components ranges around 25%. After taking this utilisation into consideration, the relative amount of critical bits increases. The accurate values are shown in Figure 2. At the top it is possible to compare the probability of failure from the used LUTs only with the complete test, which takes into consideration the utilisation of the FPGA resources. Then there is a comparison between particular types of resources. Nevertheless, no DSP blocks are shown because their utilisation is zero and moreover, the number of their critical bits is negligible in comparison with the other types of resources. Finally, there are divisions of critical bit occurrence between particular columns with SLICE. With the exception of two columns, the percentage of critical bits ranges near one value.

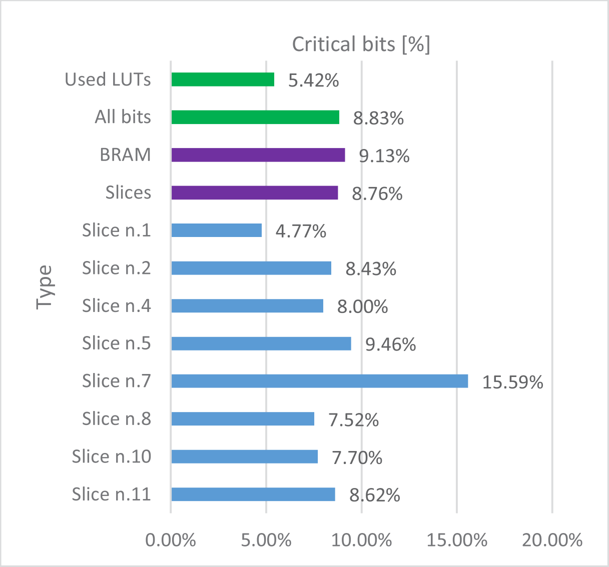

Figure 2: Relative critical bit representation from used FPGA configuration memory bits.

These results affirm that the considerable acceleration of the reliability test, which is based on taking only limited-use resources into account causes inaccuracy. However, the

TABLE IV: Absolute critical bit count and percentage expression of critical bits in test representation.

Used LUTs	All bits									
1279	9545 (1.95%)									
	BRAM	DSP	SLICEs							
	898	39	8608 (2.28%)							
			1st	2nd	4th	5th	7th	8th	10th	11th
(5.42%)	(2.28%)	(0.05%)	586 (1.24%)	1035 (2.19%)	982 (2.08%)	1162 (2.46%)	1914 (4.05%)	924 (1.96%)	946 (2.00%)	1059 (2.24%)

required time for evaluation is 20 times shorter and the total difference in the percentage of critical bits is only 3.4 percentage points. This is satisfactory precision for fast reliability estimation.

IV. Conclusions

This paper is focused on reliability analysis of the FPGA reconfiguration controller. This controller is designed for utilisation in the same FPGA as the circuit which will be fixed due to reconfiguration when a fault in the configuration memory occurs. In this case the reliability of the controller itself is important. For this purpose, the FT-EST platform is utilised. The FT-EST is capable of ascertaining the number of critical bits, which means the count of configuration bits in which a fault causes an entire system failure. However, this is the particular value which has to relate to the number of all utilised evaluated circuit configuration bits. For this reason, two approaches are compared. First, a complete evaluation of all configuration bits is executed. When the result was related to the utilisation of the FPGA resources, which is approximately 25%, the critical bits were 8.83% of the used configuration bits of the reconfiguration controller. From the other approach, when only used LUT configuration bits were evaluated, the critical bits were 5.42%. However, this test is 20 times faster than the complete test even though the accuracy is lower. Thus, when a fast estimation of circuit reliability is required, the used LUT-only evaluation is highly advantageous.

In future research, we plan to increase the fault tolerance of the reconfiguration controller by using spatial redundancy and ability of reconfiguration, which can be utilised for fixing controller faults itself.

Acknowledgements

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II), the project IT4Innovations excellence in science – LQ1602 and the BUT project FIT-S-17-3994 and the JU ECSEL Project SECREDAS (Product Security for Cross Domain Reliable Dependable Automated Systems), Grant agreement No. 783119.

References

[1] H. Quinn, “Radiation effects in reconfigurable FPGAs,” *Semiconductor Science and Technology*, vol. 32, no. 4, p. 044001, mar 2017. [Online]. Available: https://doi.org/10.1088/1361-6641/aa57f6

[2] F. Siegle, T. Vladimirova, J. Ilstad, and O. Emam, “Mitigation of Radiation Effects in SRAM-Based FPGAs for Space Applications,” *ACM Comput. Surv.*, vol. 47, no. 2, pp. 37:1–37:34, Jan. 2015.

[3] I. Koren and C. M. Krishna, *Fault-Tolerant Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[4] Y. Zhang and J. Jiang, “Bibliographical Review on Reconfigurable Fault-tolerant Control Systems,” *Annual Reviews in Control*, vol. 32, no. 2, pp. 229 – 252, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578808000345

[5] H. Quinn, W. H. Robinson, P. Rech, M. Aguirre, A. Barnard, M. Desogus, L. Entrena, M. Garcia-Valderas, S. M. Guertin, D. Kaeli, F. L. Kastensmidt, B. T. Kiddie, A. Sanchez-Clemente, M. S. Reorda, L. Sterpone, and M. Wirthlin, “Using benchmarks for radiation testing of microprocessors and fpgas,” *IEEE Transactions on Nuclear Science*, vol. 62, no. 6, pp. 2547–2554, Dec 2015.

[6] J. Tarrillo, J. Tonfat, L. Tambara, F. L. Kastensmidt, and R. Reis, “Multiple fault injection platform for sram-based fpga based on ground-level radiation experiments,” in *2015 16th Latin-American Test Symposium (LATS)*, March 2015, pp. 1–6.

[7] T. Nidhin, A. Bhattacharyya, R. Behera, T. Jayanthi, and K. Velusamy, “Verification of Fault Tolerant Techniques in Finite State Machines Using Simulation based Fault Injection Targeted at FPGAs for SEU Mitigation,” in *Electronics and Communication Systems (ICECS), 2017 4th International Conference on*. IEEE, 2017, pp. 153–157.

[8] C. Bernardeschi, L. Cassano, A. Domenici, and L. Sterpone, “Accurate Simulation of SEUs in the Configuration Memory of SRAM-based FPGAs,” in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 115–120.

[9] A. Benso, A. Bosio, S. Di Carlo, and R. Mariani, “A Functional Verification based Fault Injection Environment,” in *Defect and Fault-Tolerance in VLSI Systems, 2007. DFT’07. 22nd IEEE International Symposium on*. IEEE, 2007, pp. 114–122.

[10] M. Liu, Z. Zeng, F. Su, and J. Cai, “Research on Fault Injection Technology for Embedded Software based on JTAG Interface,” in *Reliability, Maintainability and Safety (ICRMS), 2016 11th International Conference on*. IEEE, 2016, pp. 1–6.

[11] S. Rudrakshi, V. Midasala, and S. Bhavanam, “Implementation of FPGA based Fault Injection Tool (FITO) for Testing Fault Tolerant Designs,” *IACSIT International Journal of Engineering and Technology*, vol. 4, no. 5, pp. 522–526, 2012.

[12] C. López-Ongil, M. Garcia-Valderas, M. Portela-García, and L. Entrena, “Autonomous Fault Emulation: A New FPGA-based Acceleration System for Hardness Evaluation,” *Nuclear Science, IEEE Transactions on*, vol. 54, no. 1, pp. 252–261, 2007.

[13] T. Schweizer, D. Peterson, J. M. Kühn, T. Kuhn, and W. Rosenstiel, “A Fast and Accurate FPGA-based Fault Injection System,” in *Field-Programmable Custom Computing Machines (FCCM), 2013 IEEE 21st Annual International Symposium on*. IEEE, 2013, pp. 236–236.

[14] M. Alderighi, F. Casini, S. d’Angelo, M. Mancini, S. Pastore, and G. R. 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*. IEEE, 2007, pp. 105–113.

[15] J. Lojda, J. Podivinsky, O. Cekan, R. Panek, and Z. Kotasek, “FT-EST Framework: Reliability Estimation for the Purposes of Fault-Tolerant System Design Automation,” in *2018 21st Euromicro Conference on Digital System Design (DSD)*, Aug 2018, pp. 244–251.

[16] M. Straka, J. Kaštil, and Z. Kotásek, “Generic Partial Dynamic Reconfiguration Controller for Fault Tolerant Designs Based on FPGA,” in *NORCHIP 2010*. IEEE Computer Society, Nov 2010, pp. 1–4.

[17] *Virtex-5 FPGA Configuration Guide*, XILINX, May 2017, UG191.