

An FPGA based fault emulator

J. Kvasnička, P. Kubalik, H. Kubátová

Department of Computer Science and Engineering
Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2
e-mail: (kvasnj1, xkubalik, kubatova)@fel.cvut.cz

Abstract— A hardware fault emulator based on programmable devices (FPGA) is presented. The emulator performs a single-bit fault injection in bitstream on top of the implemented circuit, emulating the SEU event. The combinational circuits mapped in FPGA are tested and SEU-fault resistance is observed.

I. INTRODUCTION

The main motivation for single-bit fault observing is the Single Event Upset [1], [2] (SEU) existence. Systems based on FPGA are sensitive to SEU, because the configuration of operating FPGA is being held in SRAM cells. The SEU appearance can lead to a system malfunction. The chance of SEU occurrence is not limited to radiation-hostile environment. SEU was observed even at the ground level [3].

The FPGA emulator uses a reconfiguration method to emulate an SEU impact. The FPGA design typically does not occupy the whole FPGA device. Here is the difference between fault injections in the Register Transfer Level and in the mapped design. Each bit in bitstream could be one possible fault. The mapped design will always come with unused bits.

The primary goal is to observe the SEU resistance with regard to the bitstream utilization. That leads to a software bitstream analysis, which has to be performed before testing.

Faults are quantified in 4 categories (described in [4] in detail): *Hidden* (A), *detected* (B), *undetected* (C) and *temporary detected* (D). This fault classification assumes tested circuits to be protected by some kind of CED technique (in our case the even parity predictor is used). *Fault Security* (FS), *Self Testing* (ST) and *Totally Self-Checking* (TSC) properties are assessed from these 4 classes.

In comparison with other FPGA fault emulator [5], our incoming fault emulator extends the range of the fault set to cell interconnection and bus connections.

II. THE FPGA EMULATOR

The core of the FPGA fault emulator is an FPSLIC device (Atmel's SoC combining FPGA and AVR). The fault injection and the fault classification is performed in the FPSLIC device.

The faults are injected only into the tested circuit (benchmark), see Fig. 1. The problem of possible logic mixing and aliasing was solved by tested circuit floorplanning into the

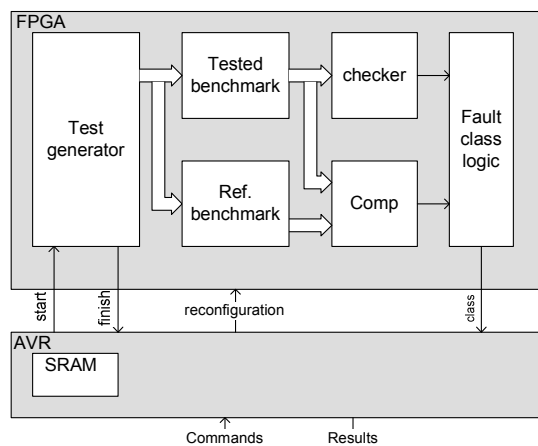


Fig. 1. FPGA part of fault emulator

separate area of the FPGA.

The set of all possible faults in a bitstream can be divided into the 2 subset: the *safe* set (fault injection should not lead to shorts in the FPGA) and *risk* set (fault injection should lead to shorts and therefore the behavior is unknown).

The *safe* fault set consist of *design-independent* subset (typically LUTs, several 2-to-1 MUXes can be also included), which can be tested anywhere, no matter where the design lays; and *design-dependent* subset (most of unused logic; drivers, turn points and repeaters of unused wires and busses, clocking, reset and many others configuration bit), where each bit in the design has to be wisely considered, whether the fault is safe to test.

Each fault test from the risk fault set should be followed by a test, which should guarantee non-destruction function of the FPGA due to possible shorts. In our case the non-destructivity test will be guaranteed by the cycle of “dummy”-fault full test (bitstream is without fault).

At the present time the safe-or-risk fault decision is being moved from AVR to PC. Until recently, the only design-independent safe fault set testing wasn't memory intensive and only small part of the bitstream holding was sufficient. The design-dependent fault analysis requires more memory for operation than the FPSLIC SRAM memory can withstand.

III. PARTIAL RESULTS

The limited results from our previous work were obtained,

TABLE I:
PARTIAL RESULTS OF FAULT SIMULATION

Circuit	Inputs	Outputs	Original circuit [LUTs]	Parity generator [LUTs]	Number of all faults	A (hidden faults)	B (detected faults)	C (undetected faults)	D (temporary detected)
alu1	12	8	8	47	656	0	656	0	0
alu2	10	8	44	47	1072	109	935	0	28
alu3	10	8	45	45	1044	130	877	8	29
Apla	10	12	48	25	900	141	625	5	129
br1	12	8	50	15	810	141	456	69	144
s1488	14	25	310	50	4286	638	3060	85	503
s1494	14	25	276	53	3938	645	2785	67	441
s2081	18	9	22	25	536	22	494	0	20
s386	13	13	57	18	976	170	646	25	135

but these results are limited only to the safe (only LUTs) fault injection [2]. These results cover only approx. 10% of the bitstream.

Results of our previous hardware fault emulation [6] are shown in TABLE I. "Circuit" is the benchmark name, "Inputs" and "Outputs" are numbers of primary inputs and primary outputs, "Original circuit" means a number of used LUTs for original circuit, "Parity generator" means a number of used LUTs for the parity generator, "Number of all faults" are all tested faults and "A, B, C, D" are classes derived by our fault classification.

IV. EXPECTED RESULTS

Our actual work is the extending of the tested fault list to faults, which belongs to the safe faults set (and which is not limited to LUTs) and risk set.

The goal of our work is a functional FPGA simulator, which would cover at least 50% of bitstream allocated by benchmark. 100% coverage is not feasible in this design due to some shared logic with testing environment (shared clock, some IO ports used, occupied busses) and the design itself (benchmark is driven from test generator, not IO ports etc.).

Much more hidden faults (category a) is expected in safe fault set testing. Therefore more soft distinction among the hidden faults might have become necessary (used by design or not used by design).

An answer to the question, whether SEU can lead to the irreversible destruction of the FPSLIC chip or not will be solved and presented during the WiP Euromicro conference.

Results will be used in our future work, which is a software simulator at the bitstream level. Such a simulator would be great contribution to a dependability analysis of design mapped to FPGA. This approach could lead to the more precise fault-tolerant design and its evaluation with respect to the real dependability parameters.

REFERENCES

- [1] Bellato, M., Bernardi, P., Bortalato, D., Candelaro, A., Ceschia, M., Paccagnella, A., Rebaudogo, M., Sonza Reorda, M., Violante, M., Zambolin, P.: "Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA." Design Automation Event for Electronic System in Europe 2004, pp. 584-589.
- [2] QuickLogic Corporation.: Single Event Upsets in FPGAs, 2003, www.quicklogic.com
- [3] Normand, E.: "Single Event Upset at Ground Level," IEEE Transactions on Nuclear Science, vol. 43, 1996, pp. 2742-2750.
- [4] Kafka L., Kubalík P., Kubátová H., Novák O.: "Fault Classification for Self-checking Circuits Implemented in FPGA", Proceedings of IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop. Sopron University of Western Hungary, 2005, pp. 228-231.
- [5] Kafka, L., Novak, O.: "FPGA-based fault simulator", In Proceedings of the 2006 IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems DDECS2006, CTU Prague 2006, vol. 1, pp. 274-278.
- [6] Kvasnicka, J.: "Highly Reliable Design Based on FPGA circuits", CTU FEE, 2006, (in Czech).