Experimental SEU Impact on Digital Design Implemented in FPGAs

Jiří Kvasnička, Pavel Kubalík, Hana Kubátová
Czech Technical University in Prague, Dept. of Computer Science and Engineering
e-mail: kvasnj1@fel.cvut.cz, xkubalik@fel.cvut.cz, kubatova@fel.cvut.cz

Abstract

The main aim of our research is to investigate the influence of SEU on a digital circuit implemented in FPGA. The FPGA resources occupied by design are divided into several groups. SEU impact is investigated for each group. To make a real dependability model the real effects of injected errors and faults have to be studied. The SEU emulator deals with single-bit change in bitstream. Emulation is performed in the user-selected area. Look-Up-Tables, cell interconnections, cell-to-bus connections and routing resources are considered. Other FPGA resources are not considered. Combinatorial circuits and MCNC benchmarks were measured due to our knowledge of FPGA resource limitation. All tests were performed on Atmel FPSLIC architecture.

1. Introduction

The size, versatility, and price of commercial Field Programmable Logic Arrays (FPGAs) allow replacing ASICs in many applications [1]. The SRAM, which the FPGA configuration is based on, is sensitive on Single Event Upset (SEU) [2]. The SEU can be caused by high-energy particles hitting the silicon and changing the logic state of the SRAM cell.

The main aim of our research is to design dependable circuit in FPGA. To make a real dependability model the real effects of injected errors and faults have to be studied. The SEU impact to FPGA was also performed by another research group [3]. Possible situations are divided into 6 groups here.

The self-checking (SC) circuit (a method based on a concurrent error detection (CED) technique) is used to detect an occurrence of a fault in the tested circuit. Only one copy of the SC circuit is not sufficient to increase dependability parameters. Thus, we use the Modified Duplex System (MDS) architecture [4].

There are three basic quantitative criteria in a field of CED: fault security (FS), self-testing (ST) and totally self-checking (TSC) properties [5]. These three aspects have to be used in an on-line testing field to evaluate the level of safety of the designed or the modeled system.

To determine whether the circuit satisfies the TSC property, detectable faults belonging to one of four classes A, B, C and D [6] have to be calculated.

The structure of the paper is following: The section 2 describes our FPGA fault analysis. Our emulator where the measurements were performed is described in section 3. The section 4 shows results from the experimental fault injection and dependability calculation, and the section 5 concludes the results and the contribution of the measurement.

2. FPGA fault analysis

Every bit of bitstream in the FPGA can be classified in two different ways. The first criterion is the FPGA resource, which the current bit lays in, and which function is the bit responsible for. The second criterion is the electrical character of the fault, when the design is mapped in the FPGA.

2.1. FPGA resources

The FPGA resources were divided into disjoint sets (listed below), which are specified by their location and function.

1) *LUT*: It holds the logic function in SRAM memory.
2) *Cell interconnection*: This is the configuration of Logic cells. These bits are responsible for the LUT correct inputs selection; feedback in logic cell, correct output selection (registered/nonregistered function, 3-input or 4-input LUT organization).
3) *BUS to Cell/Cell to BUS*: This is a bidirectional connection, which connects the Logic Cell to one of BUS plane.
4) *BUS crossing*: This is a connection in the centre of perpendicular bus crossing, which allows connections between these lines.
5) *BUS repeater*: This is a simple 4-port switchbox. It allows driving of each wire from every input.

6) *Forbidden*: These are bits, that have their own place in bitstream, but a physical SRAM cell is not assigned to them. We assume, that these bits does not exist therefore should not be tested.
7) *Unexplored*: other resources, which were not listed above and which have not yet been explored.

This list is not complete. *RAM*, *reset*, *clocks* and *I/O pad* (only partially covered here by *unexplored* set) are missing in the list. Our emulator almost doesn't use them. The time needed to determine the category corresponds to $O(1)$ time complexity.

2.2. Fault group

Primary goal of fault division is a separation of bits, which can never influence the function of the loaded design, and bits which can lead to the modification of the design. Such a classification is possible only when the complete structure of FPGA is known (and corresponding bitstream position is known, too). The distribution into fault groups is design specific (instead of the distribution of the FPGA resource sets, which is determined by the FPGA architecture).

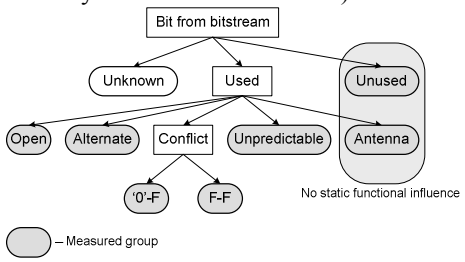

Figure 1. Fault groups

Every fault belongs to one fault group, as shown in Fig. 4. In the first approximation, the bit is placed in *Used*, *Unused* or *Unknown* group.

1) *Unused*: this bit does not concern the design area. Therefore neither static nor dynamic design changes are expected.
2) *Used*: This group belongs to bits, which are used by the design. Fault in these bits can lead to the design modification
3) *Unknown*: A class of bits, whose correct class cannot be evaluated. This, in simplification, can be caused by unknown state on the wire, or by missing information about the bit meaning in FPGA resources.

2.3. Subcategories of used bits

A group of used bits covers many versatile types of fault. Almost all of them can lead to design functional alternation. The only exception is *Antenna* group.

a) **Open**: The basic model of this class is a wire interruption, which can have many different origins in the FPGA architecture. The most lucid case is an open in bus crossing (Figure 2).

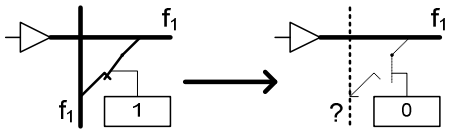

Figure 2. Open at the bus crossing

b) **Alternate**: These bits alternate the design without conflict on the bus. A 2:1 mux is shown in Figure 3 is a typical example of alternate group. Another case is a bit alternation in truth table of LUT.

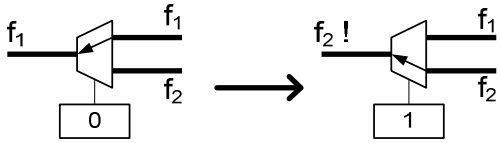

Figure 3. Alternate at 2:1 multiplexor

c) **Conflict:** This is a special category defined by connecting of two or more driven wires. This conflict leads to a short circuit between power supply and ground through the drivers. A simple conflict can occur on BUS crossings, when both horizontal and vertical wires are driven Figure 4. The "◦" operator in Figures 4 and following should be interpreted as dominance: if one of the operands can drain more current, that will be result of the "◦" operator.

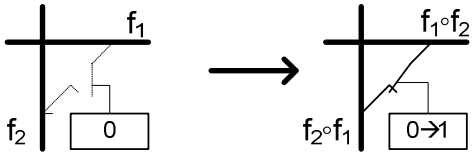

Figure 4. Conflict at the bus crossing

Conflicts can also occur in multiplexers when more than 1 input is selected, in bus networks. A conflict is separated into 2 subcategories: a)"0-F" in cases, where any function conflicts with constant '0', and b) "F-F", where conflict is between two non-constant functions.

d) **Unpredictable** faults are a special case of open, where the default logical value '1' is substituted with 'Z', see Figure 5.

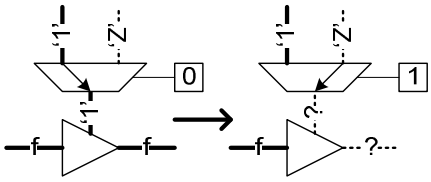

Figure 5. Unpredictable at the output enable buffer

e) **Antenna**: A special case of a used bit, where an unused wire is connected to the datapath. This fault statically has no influence on the design function. Only delays on wires can worse (Figure 6).

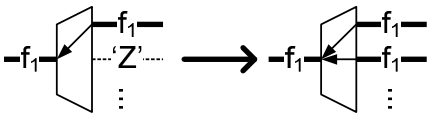

Figure 6. Antenna in the mux

3. Our Emulator

We used a hardware emulator described in [7], which was extended with ability to test bits descibed in chapter 2. The SEU fault is emulated in the FPGA through the dynamic reconfiguration. Each single bit in the bitstream represents one possible fault, which is injected into the FPGA by writing the inverted value into the design.

The FPGA is virtually divided into two areas: *fault injection* area and *fault safe* area. Each area is an exact half of the FPGA. The tested circuit is placed into the *fault injection* area, while the second (reference) copy of the design is placed in the *fault safe* area.

The emulator at Figure 7 was implemented in the ATMEL FPSLIC device. SEU is emulated by writing new bitstreame byte with alternated bit

In this paper, the resulting class of the fault is interpreted only as a "fault is modifying the design" or "fault is not modifying the design.".

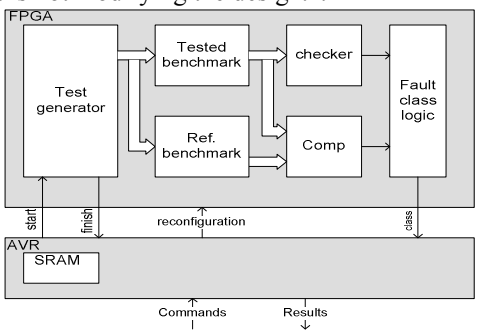

Figure 7. The FPGA fault emulator overview

The above described experimental analysis of bitstream covered 95.5% of the whole bitstream (Fig. 9). Only *IO pads* and *others* (i.e. 4.5% of bitstream) are neither tested, nor analyzed for possible faults.

A measured distribution of FPGA resources in s1488 is shown at Figure 10. Only faults modifying the design are included in the distribution.

Fault categories of the same benchmark (s1488, as at Figure 9 above) are shown at Figure 10. Only bits modifying the design are concerned. *Unused* bits (51% of the FPGA bitstream) and *antenna* bits (17% of the whole FPGA bitstream) do not act in the chart, because none of these bits can actually modified the design.

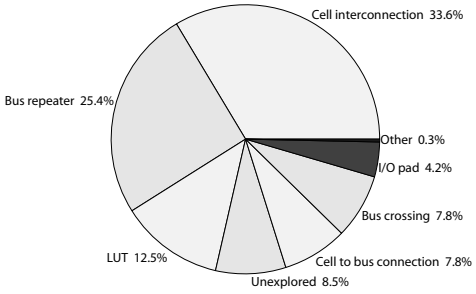

Figure 8. Resources in Bitstream

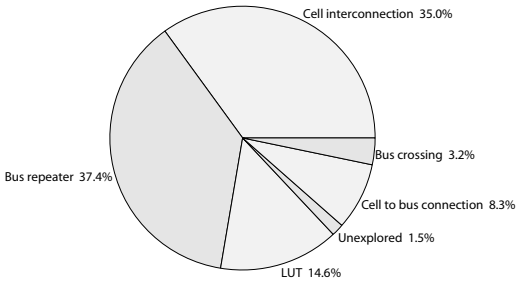

Figure 9. Distribution of area impacted fault

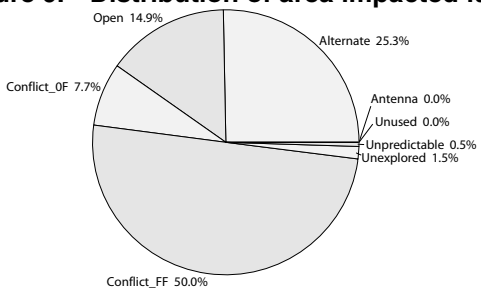

Figure 10. SEU sensitive part distribution

4. Results

A set of measurements of MCNC benchmarks with parity predictor was performed in the FPGA fault emulator. The availability computations [5] were used to compare our MDS architecture [4] with a standard duplex system and with TMR (Triple Modular Redundancy) system.

Experimental results are divided into two groups. First group divides faults by the type of area impacted by SEU (Table 1). Here "*BENCH*" is the name of the benchmark circuit, "*FS*" is the probability that a fault is detected by a code word calculated for each part individually, "*AO ALL*" is the area overhead of whole design, "*FS ALL*" is the probability that a fault is detected by a code word calculated for the whole design, "*ASS*" is the steady-state availability and "*Impr. ASS*" indicates the percentage improvement of *ASS* against standard duplex system. 0% improved ASS means, that system is equivalent to standard duplex system. 100% improved ASS means, that system is equivalent to standard triplex system.

Second group divides faults by type of SEU manifestation in design see Table 2. Here "*BENCH*" is the name of the benchmark circuit, "*FS*" is the probability that a fault is detected by a code word calculated for each part individually, "*AO ALL*" is the area overhead of whole design, "*FS ALL*" is the probability that a fault is detected by a code word calculated for whole design, "*ASS*" is the steady-state availability and "*Impr. ASS*" indicates the percentage improvement of *ASS* against standard duplex system. The results show that the self-checking circuits and their real utilization is possible to use in dependable applications.

5. Conclusion

We have experimentally proved that the steady-state availability of the modified duplex system is significantly better, than the availability of the standard duplex system, even when the technology of the FPGA is considered and aprox. 95% of the whole bitstream is exposed to the SEU emulation. We are able to obtain precisely how many bits can change the design, which is actually mapped and run into the FPGA. This is a significant pre-requirement in dependability modeling and calculations. The obtained results opens a new field of application in adjusting the place&route tool to compile a design, which would be slightly more resistant to SEU - but with noticeable worse delays in FPGA and maximum frequency decrease.

Acknowledgment

This research has been partially supported by MSMT under research program MSM6840770014 and FI-IM4/149 grant.

References

[1] Ratter, D.:"FPGAs on Mars", www.xilinx.com,Xcell Journal Online, 2004.

[2] Normand, E.: "Single Event Upset at Ground Level," IEEE Transactions on Nuclear Science, vol. 43, pp. 2742-2750, 1996.

[3] Bellato, M., Bernardi, P., Bortalato, D., Candelaro, A., Ceschia, M., Paccagnella, A., Rebaudego, 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, 2004.

[4] P. Kubalik, R. Dobias and H. Kubatova.: "Dependable Design for FPGA based on Duplex System and Reconfiguration", In Proc. of 9th Euromicro Conference on Digital System Design, Los Alamitos: IEEE Computer Society, pp. 139-145, 2006.

[5] Pradhan, D. K.: "Fault-Tolerant Computer System Design", Prentice-Hall, Inc., New Jersey, 1996.

[6] Kafka L., Kubalík P., Kubátová H., Novák O.: "Fault Classification for Self-checking Circuits Implemented in FPGA", Proc. of IEEE DDECS Workshop. Sopron University of Western Hungary, pp. 228-231, 2005.

[7] Kubalík, P., Kvasnička, J., Kubátová, H.: " Fault Injection and Simulation for Fault Tolerant Reconfigurable Duplex System", In Proc. of DDECS 2007, pp. 357-360,

Table 1: Faults divided by type of impacted area

Bench	*FS [%] calculated for area group*						*AO ALL [%]*	*FS ALL [%]*	*ASS ALL*	*IMPR ASS [%]*
	LUT	*Cell int*	*Bus Cell*	*Bus crs*	*Bus rep*	*Other*				
alu1	100	98,1	91,8	81,1	73,6	82,9	334,7	89,1	0,99995521	-105,9
apla1	81,7	80,6	77,9	57,3	58,5	77,6	54,5	71,7	0,99998531	32,4
b12	96,9	93,0	89,4	72,5	70,0	75,0	-3,0	84,8	0,99999669	84,8
br11	68,1	64,7	56,9	48,7	46,1	46,7	23,6	57,0	0,99998571	34,3
bw	83,0	79,3	74,0	50,7	58,3	62,0	15,7	71,3	0,99999164	61,6
s1488	83,6	81,6	77,2	63,1	63,8	66,8	22,6	74,1	0,99999153	61,1

Table 2: Faults divided by SEU manifestation in design

Bench	*FS [%]*						*AO ALL*	*FS ALL*	*ASS ALL*	*IMPR ASS*
	ALTER	*OPEN*	*C_0F*	*C_FF*	*UNPRE*	*UNKN*				
alu1	100,0	86,9	96,2	82,7	100,0	82,9	334,7	89,1	0,999955208	-105,9
apla1	82,3	70,1	80,2	65,4	76,9	77,6	54,5	71,7	0,999985307	32,4
b12	95,8	83,2	87,7	78,6	95,3	75,0	-3,0	84,9	0,999996694	84,8
br11	67,9	56,5	62,0	50,8	72,7	46,7	23,6	57,0	0,999985711	34,3
bw	82,6	71,3	69,1	65,7	77,3	62,0	15,7	71,3	0,999991643	61,6
s1488	83,1	72,6	81,9	68,9	79,4	66,8	22,6	74,1	0,999991532	61,1