FPGA BITSTREAM ANALYSIS AND EMULATION OF SEU EFFECT

Jiří Kvasnička
Computer Science and Engineering, 2nd class, full-time study
Supervisor: Hana Kubátová

Dpt. of Computer Science and Engineering, Faculty of Electrical Engineering, CTU Prague
Karlovo nám. 13, 121 35 Prague 2, Czech Republic

kvasnj1@fel.cvut.cz

Abstract. This work analyzes an effect of single-bit error in an FPGA configuration memory. Several models for bitstream faults are described. A detailed analysis of various kinds of fault is performed, with respect to the different elements in the FPGA. A hardware emulator for all these fault models is proposed and presented. Experimental results of a MCNC benchmarks set testing are presented. Results of s1488 benchmark are discussed in detail.

Keywords. SEU, SEU emulation, bitstream analysis, FPGA fault model

1 Introduction

FPGA (Field programmable grate array) devices are mainly based on a SRAM (Static Random Access Memory). SRAM reliability is limited, as they are susceptible to SEU (Single Event Upset), even at the ground level [1]. Probability of SEU varies with the actual solar flare, latitude (approximately 6× worse conditions at the North Pole, than at the equator) and mostly with the altitude [2].

Three main methods of dependability parameters estimation are known: software *simulation*, hardware *emulation* and *irradiation* in dedicated radiation test facility.

Software simulator efficiency strongly depends on model used (which is a closely guarded intellectual property of FPGA producers) and is often limited to LUTs (Look-Up Tables) in RTL (Register Transfer Logic) level, with no routing in the FPGA [6]. The better model is used, the longer time is needed for the simulation.

The Irradiation gives us an exact insight into the design functionality under the SEU. The knowledge of complete FPGA structure is not required, nor is the FPGA control logic knowledge. The preparation of the test requires the PCB with the FPGA with the remote data readout. The testing itself requires a dedicated radiation facility and is performed in irradiation chamber[3]. Results from the irradiation are valuable and conclusive for a real device functionality in a hostile environment, but is not easily available.

Hardware SEU emulator stands between the irradiation testing and the simulation. Unfortunately, a complete knowledge of the FPGA structure is still required. This is the key problem for the emulator realization. In comparison with a software simulator, results are obtained from the mapped design; therefore result should be closer to the irradiation testing. For larger benchmark, speedup of testing is also significant.

2 FPGA structure - areas

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/non-registered 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 center of perpendicular bus crossing, which allows connections between these lines.
5. *BUS repeater*: This is a simple 4-port switch box. It allows driving of each wire from every input.
6. *Forbidden*: These are bits, that have their own place in the bitstream, but a physical SRAM cell is not assigned to them. This is caused by the AT40K bitstream byte organization. It is also possible that these bits have other (for us unknown) meaning. I assume, that these bits physically do not exist therefore their testing should be avoided.
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 on the list. The reason of this incompleteness is a fact, that the FPGA device bitstream has not yet been completely analyzed. Therefore they are not tested. However, it does not mean automatically, that SEU occurrence in a non-listed region could not destroy the design. Therefore a small uncertainty in measured data has to be considered.

3 FPGA faults

A primary goal of fault division is a separation of bits, which can never have an influence on the function of the loaded design, and bits which can lead to the modification of the design.

The distribution into these groups is defined by the design. On the other side, the distribution of the FPGA resource sets (listed in previous section) is determined only by the FPGA architecture.

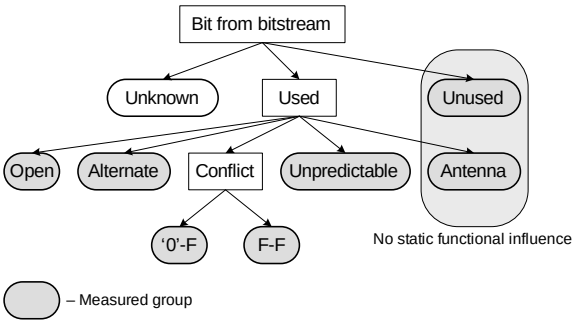

Figure 1: Fault groups

No class can be correctly evaluated without knowledge of wire state (Function, constant or High-Z) and current bit value. The computation is therefore based on the design bitstream analysis, because the distribution into these groups is specified by the physical design layout of the FPGA.

Every fault belongs to one fault group, as shown in Fig. 1. In the first approximation, the bit is placed in *Used*, *Unused* or *Unknown* group.

1. *Unused* bits do not concern the design area. Neither static nor dynamic design changes are expected. However, such a fault could potentially lead to higher current consumption.

Further division of this category was not implemented, because there is no method which would allow measuring of such a fine difference among non-revealing faults.

2. *Used* bits are primary created by the design. Any bit from this group influences an active part of the design. These bits stand in a critical positions, where switching this resource can affect the design. Almost all of them can lead to design functional alternation. A further detailed distinction of the used category follows:
   (a) *Open*: Represents a wire interruption, which can have many different origins in the FPGA architecture. The most illustrative cases are opens in a bus crossing, opens in multiplexors and transfer gates.
   (b) *Alternate*: These bits alter the design without any conflict on the bus. Examples in FPGA: changing an input in a 2:1 multiplexor and LUT truth table alternation.
   (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. The result is hard to predict, unless a detailed FPGA layout is known (especially strength of drivers). A conflict can occur on BUS crossings, in multiplexers when selecting more than one input, in bus repeater etc. The conflict can be separated into 2 subcategories:
      – "F-F", where conflict is between two non-constant functions;
      – "0-F", where any function conflicts with constant "0".
   (d) *Unpredictable*: A special case of open, where the change of bit lead to change of the selector (transfer gate or mux) from constant logical value ("1' or "0") to unconnected wire ("Z").These faults were separated from other faults only because of unknown physical layout of these elements.
   (e) Antenna: A where an unused wire is connected to the data-path. This fault statically has no influence on the design function. Only delays on wires can worse, because an extra load capacity is appended.

3. *Unknown* bits form a class for bits, whose correct class cannot be evaluated. This can be caused by unknown state on the wire, or by missing information about the bit meaning in FPGA resources.

The time needed to determine the category, which the bit belong in, surprisingly corresponds also to $O(1)$ time complexity, at least in AT40 FPGA. Although the computation time does not depend on the FPGA size, the time constant, however, strongly depends on the FPGA architecture and analyzed bit location.

4 Fault coverage of the emulator

The above described experimental analysis of bitstream covered 95.5% of the whole bitstream. A detailed area distribution is shown at Fig. 2. Only IO pads and others (i.e. 4.5% of bitstream) are not tested, nor analyzed for possible faults.

An example of the measured distribution of the FPGA resources is shown at Fig. 3. The s1488 benchmark is presented. In contrast with Fig. 2, only faults modifying the design are included in the distribution.

Fault categories of the same benchmark s1488 are shown at Fig. 4. Only bits that modify 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 modify the design (see the definitions in section 3)

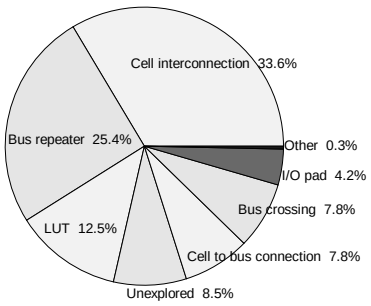

Figure 2: Resources in AT40K Bitstream. *IO pads* and *others* were not tested, *forbidden* are not listed

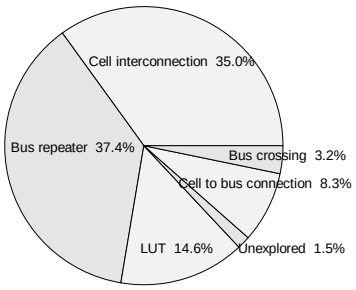

Figure 3: Distribution of the s1488 bitstream according to FPGA resources, where a fault modifies the design

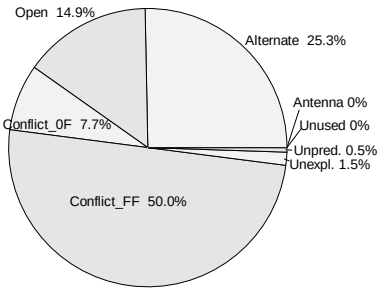

Figure 4: Distribution of fault categories, where the fault modify the function of the s1488 benchmark

Table 1: Total bits count in s1488 benchmark

[bits]	LUT	Cell int.	Cell /bus	Bus cross	Repea-ter	Forbid-den	Unexp-lored
Unused	13304	23286	6155	6434	18916	1104	–
Alternate	4360	3417	–	–	–	–	–
Open	–	730	1643	363	1077	–	–
Conflict 0F	–	2228	–	–	–	–	–
Conflict FF	–	5133	817	1181	9403	–	–
Antenna	–	12108	2425	3062	5824	–	–
Unpredictable	–	570	–	–	–	–	–
Unknown	–	–	–	–	–	–	11740

5 Results

A set of measurements on 10 MCNC benchmarks with parity predictor was performed in the FPGA fault emulator. A huge amount of data was obtained from the SEU emulator. A larger benchmark (s1488) was picked from the set of tested benchmarks. Most interesting parameters will be shown here. At the end of this section, a summary of all tested benchmark will be shown.

5.1 s1489 benchmark results

A Class distribution of all bitstream bits of s1488 benchmark is presented in Table 1. Each column represents one resource from section 2 and each row represents one fault category from section 3. The number of total bits covers both bits, which modify the benchmark function, and those with no influence on the benchmark function. Impossible combinations of categories (e.g., conflict at LUT) are marked by "–"symbol.

Table 2 shows the ratio between bits, which modify the design during the fault injection, to all bits in corresponding area and fault category. The layout of the results in Table 2 is similar to Table 1.

Interpretation of Table 1: although the place and route tool from Atmel reports 329 used logic cells, more logic cells are occupied. Additional logic cells are used on routing. However, less LUT bit number (only 4360 bits) is in alternate fault group instead of more than 5280 bits expected. This less number is caused by utilization of LUT, which is not always used as a 4-input LUTs or two 3-input LUTs.

Table 2: Ratio of bits altering the design to all bits in category

[%]	LUT	Cell int.	Cell /bus	Bus cross	Repea-ter	Forbid-den	Unexp-lored
Unused	0.0	0.0	0.0	0.0	0.0	0.0	–
Alternate	84.8	79.0	–	–	–	–	–
Open	–	98.8	97.6	100.0	100.0	–	–
Conflict_0F	–	87.9	–	–	–	–	–
Conflict_FF	–	64.9	59.6	38.6	89.1	–	–
Antenna	–	0.0	0.0	0.0	0.0	–	–
Unpredictable	–	19.6	–	–	–	–	–
Unknown	–	–	–	–	–	–	3.3

Interpretation of Table 2: The assumption, that the antenna and unused bits have no influence on the design function, was confirmed by these experiments (all the unused field and antenna field are zero). Open fault have significantly higher probability that can change the design. On the other hand, a conflict between two functions has unexpectedly low probability of design changes. A possible answer why the conflict has low probability of design changes could be a different current drain of drivers.

5.2 Results of all tested benchmarks

Table 3 shows several benchmarks and their ratio of bits, which can change the design, to all bits in corresponding fault category. The first line in the Table 3 shows a total LUT bit number in alternate category. This line serves only as an indicator of the benchmark size.

Second line in Table 3 shows absolute size of the benchmark. LUT bit count was used here for comparing of benchmark sizes.

6 Conclusions

By the presented SEU emulator, I was able to obtain precisely how many bits can change a design, that is actually mapped and running inside the FPGA. This is a significant pre-requirement in dependability modeling and calculations. Moreover, we are able to separate bits, which cannot change the design from the whole bitstream, in estimated near $O(n)$ time complexity, where n is size of the FPGA.

Table 3: Ratio of altering bits in fault categories for different benchmarks

[%]	5xp1	alu1	alu2	alu3	b11	b12	br1	bw	s1488	s1494
Used LUT bits [bits]	388	676	1102	1090	420	650	822	834	4360	4042
Unused	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Alternate	90.0	92.3	84.7	84.5	83.6	86.0	81.0	80.5	82.3	81.0
Open	97.6	99.3	98.1	97.8	98.4	98.2	99.6	98.9	98.7	98.9
Conflict 0F	93.6	94.4	87.2	87.4	89.2	86.1	87.7	88.1	87.9	88.1
Conflict FF	84.7	86.0	80.6	79.5	79.9	82.2	77.7	81.5	76.5	76.4
Antenna	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Unpredictable	24.7	19.8	10.9	15.3	25.8	48.9	11.1	20.4	19.6	22.6
Unknown	0.2	0.3	0.6	0.5	0.2	0.3	0.5	0.4	3.3	3.0

The obtained results opens a new field of application in adjusting the synthesis and place&route tool to compile a design, which would be slightly more resistant to SEU - but with noticeable worse delays in FPGA and the maximum frequency decrease. Availability parameters show, that almost all our tested benchmark are more dependable than standard duplex system.

References

[1] M. Bellato, P. Bernardi, D. Bortolato, A. Candelori, M. Ceschia, A. Paccagnella, M. Rebaudengo, M. Reorda, M. Violante, and P. Zam- bolin. Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA. *Proceedings of Design, Automation and Test in Europe Conference and Exhibition,* 1:584–589 Vol.1, Feb. 2004.

[2] K. A. LaBel. *SECCA - single event effect criticality analysis*, NASA/GSFC, 1996. `http://radhome.gsfc.nasa.gov/radhome/papers/seecai.htm`

[3] Goddard Space Flight Center Radiation Effects Facility. `http://radhome.gsfc.nasa.gov/radhome/ref/GSFC_REF.html`

[4] P. Kubalík, J. Kvasnička, and H. Kubátová. Fault injection and simulation for fault tolerant reconfigurable duplex system. *Proceeding of IEEE Design and Diagnostics of Electronic Circuits and Systems DDECS '07*, pages 357–360. Los Alamitos: IEEE Computer Society, 2007.

[5] 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, pages 228-231, 2005

[6] L. Kafka and O. Novak. FPGA-based fault simulator. *Proceeding of Design and Diagnostics of Electronic Circuits and systems, 2006 IEEE*, pages 272–276, 2006.

[7] P. Graham, M. Caffrey, J. Zimmerman, D. E. Johnson, P. Sundararajan, and C. Patterson. Consequences and Categories of SRAM FPGA Configuration SEUs. *Proceeding of Military and Aerospace Applications of Programmable Logic Devices (MAPLD).* Washington DC, Sep. 2003.