Critical-Bitstream-Based SEU Injection and Validation for Xilinx SRAM-Based FPGAs

Tingting Yu, Lei Chen, Xuewu Li, Shuo Wang, and Jing Zhou
Beijing Microelectronics Technology Institute, Beijing, P. R. China
Email: {yutingting1123, lixuewu1979, zhoujenny0915}@126.com, chenleinpu@vip.126.com, yeshi1@163.com

***Abstract*—SEU (Single Event Upset) injection system implemented in a single FPGA always suffers difficulties of partitioning circuit modules and obtaining target bitstream. This paper presents a critical-bitstream localization strategy to find out the injection target for Xilinx FPGAs. Two assumptions are proposed to obtain frame addresses and bit offsets of the critical bitstream corresponding to CUT (circuit under test). To verify the localization strategy, a SEU injection framework is also introduced. Experimental results on XQ5VLX110t show that 2977 bits are identified as critical bits and among them 343 bits are judged as SEU sensitive ones. While the process of random injection only finds 97 SEU sensitive bits. Comparing the data, the fault rate of the critical-bits injection is 52.8% higher than that of the random-bits injection. That indicates the proposed localization strategy is effective.**

***Index Terms*—SEU injection, critical Bitstream, placement constraints, precise localization, FPGA**

I. Introduction

SRAM-based FPGA suffers from SEU(single event upset) when used in radiation environment, turning the configuration bit into a contrary logic state, such as from “1” to “0” or “0” to “1” [1] and [2]. The flipped bit may change the circuit function, or may not. In order to test the SEU sensitivity of the circuit implemented in FPGA, the current method is bitstream-based SEU injection. It is more competitive than ground radiation testing, analytic technique and simulation based of model in some aspects, like the cost, accuracy, and efficiency [3].

Many SEU injection methods have been proposed already, yet there are always something unsatisfying. A common system which is carried out at the cost of big hardware overhead, such as the work in [4] and [5], is consist of three FPGA chips at least. One works as Controller, one works as DUT (Device under Test) and the last one works as Golden Device. This setup makes the flow of SEU injection clear and easy to implement. However, SEU sensitivity is related to the circuit structure, if the test circuit is implemented in DUT and Golden Device respectively, then it is not sure that the structure will keep same. If not, do comparing between the two circuits makes no sense. Another current method puts the CUT (Circuit under Test) and the Golden Circuit in a single FPGA, together with a controller and some other modules [6]-[9]. This method needs less overhead and shortens the design cycle. To separate the circuit modules into different regions on the board and obtain the configuration bits corresponding to CUT, a manual work of partition and mapping must be done firstly [10] and [11]. However, absolute partition and mapping is still an unsolved problem, because the FPGA manufacturers like Xilinx hide most of the correlation between bitstream and structure.

For the second method discussed above, this paper proposed a localization strategy to find CUT bits and further, the critical CUT bits, as sufficiently as possible. Section II gives detailed instructions of the proposed critical-bitstream localization strategy. Section III introduces a validation platform of SEU injection to test the proposed method. The experimental results and analysis with an 8-bit CLA (Carry-lookahead Adder) as the test circuit are shown in Section IV. Section V do some concluding and list some problems that need to be considered in the future.

II. The Proposed Critical-Bitstream Localization Strategy

For a SEU injection work, the circuits are designed in verilog first and then generated configuration bitstream in ISE, a tool provided by XILINX, automatically. Usually, the process of placement and rout is contained in the bitstream generation and that makes how the layout is completed not open to users. Only a placement result like Fig. 1 is provided. It seems that circuits are placed randomly in the entire board and it is impossible to find out CUT. This Section presents a localization strategy to place the different modules in detached and specific regions and help find out the corresponding configuration bits, especially the critical bitstream of CUT.

The strategy includes two steps: first, cursory localization and second, precise localization.

A. Do Placement and Mapping: The Cursory Localization

This paper uses a placement constraint “AREA_GROUP” written in the Xilinx user constraint file (UCF) [12] and [13]. An example is in Fig. 2. It leads the placement program to partition regions for different modules as the file writes. Before the constraints are

Manuscript received November 30, 2016; revised May 30, 2017.

doi: 10.18178/ijoee.5.1.29-33

added to the UCF, a general condition about the distribution of FPGA resources is necessary to study. Xilinx gives some information about the structure and the resource, which is helpful but not enough. The first assumption about the mapping between structure and bitstream is proposed. As the resources are structured in rows and columns, the bitstream also could be organized with rows, columns and frames. All the bitstream is first divided into two parts "top" and "bottom", then the "top half"/"bottom half" is arranged from row 0 to row n, and then each row is organized in columns. Different types of columns (CLB, BRAM, IOB, DSP, GTP, etc.) contain different numbers of frames. These frames are indexed with Minor address. Detailed correspondence is showed in Fig. 3. Based of this assumption, allocate an appropriate size of region for each module and write location constraints in UCF. That may require a few attempts. Then configuration data corresponding to the allocated regions are obtained.

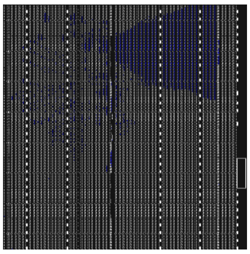


Figure 1. An automatic placement result.

```
#placement constraints
inst "U1/*" area_group="area_group_1";
area_group"area_group_1"
range=slice_x<0>y<60>:slice_x<107>y<119>;
inst "U2/*" area_group="area_group_2";
area_group "area_group_2"
range=slice_x<92>y<0>:slice_x<107>y<7>;
inst "U3/*" area_group="area_group_3";
area_group "area_group_3"
range=slice_x<0>y<0>:slice_x<3>y<3>;
inst "U4/*" area_group="area_group_4";
area_group "area_group_4"
range=slice_x<4>y<0>:slice_x<7>y<3>;
```

Figure 2. Example of placement constraints.

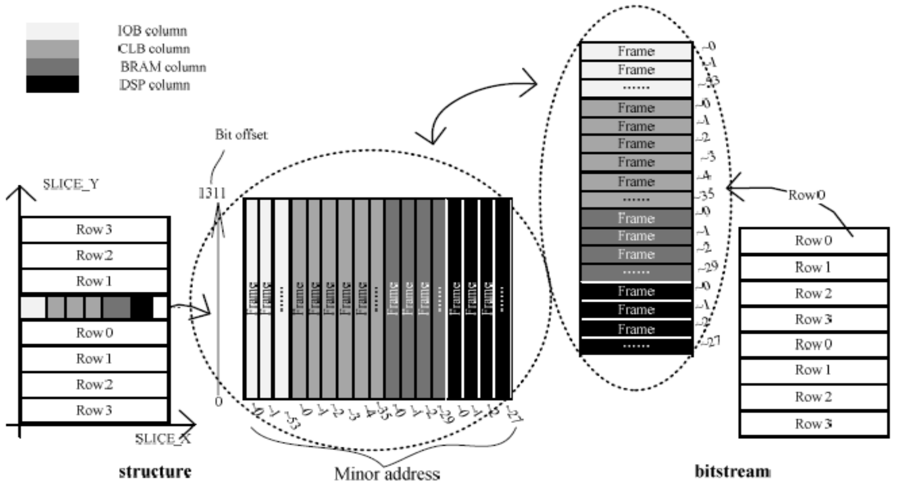
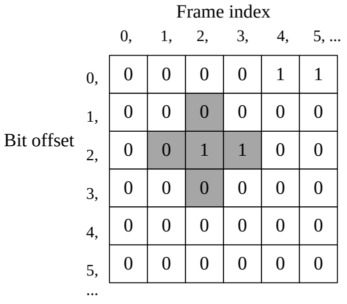


Figure 3. Assumption 1: Mapping between structure and bistream.

B. Do Balance: The Precise Localization

The cursory localization identifies the scope of configuration data of CUT. But not all bits in the scope are related to the design and some of them are don't care bits. In the data, some bits are logic "1" and others are logic "0". It is not definite that the "1" is configuration bit and "0" is don't care bit. By analyzing the void bitstream which contains no circuit design, a fact shows all bits are logic "0". That means logic "1" has more probability to be seen as a occupied one when a design is configured into FPGA. To achieve a balanced result, the second assumption is proposed: every "1" and the four bits around it are defined as critical bits (see Fig. 4). Furthermore, there is another meaning of this assumption: adjacent bits more likely share the same physical structure, such as LUT, Flip Flop or Multiplexer.

In this way, the frame addresses and bit offsets of the critical bitstream has been located.



Frame index

Bit offset	0,	1,	2,	3,	4,	5, ...
0,	0	0	0	0	1	1
1,	0	0	0	0	0	0
2,	0	0	1	1	0	0
3,	0	0	0	0	0	0
4,	0	0	0	0	0	0
5, ...	0	0	0	0	0	0

Figure 4. Assumption 2: Definition of critical bits.

III. The Validation Platform of SEU Injection

To verify the proposed localization strategy, a SEU injection platform is built. The framework is made up of a hardware system running in FPGA, a software program on host computer and a serial communication cable as the connecter. The general view is in Fig. 5.

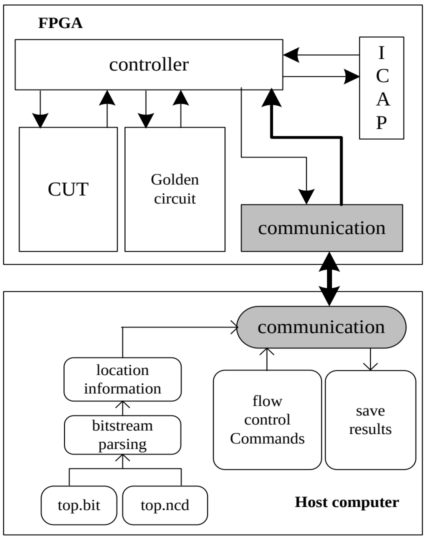


Figure 5. General view of the SEU injection platform.

A. Circuit in FPGA

There are five modules insides the FPGA, including controller, communication module, ICAP (Internal Configuration Access Port) interface, CUT and Golden Circuit.

1) Controller

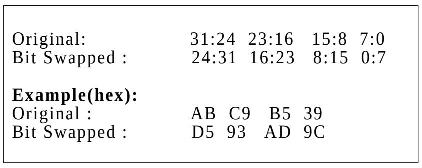
Controller is the center hinge which links all the other modules. The operation mechanism is a state machine. It sends commands of readback and reconfiguration and gets frame data of configuration memory through ICAP. Also it controls the process of communicating with host computer. The transfer data includes user-specific commands which are for triggering the state machine, injection targets and injection results. Test vectors for CUT and Golden circuit are generated in controller, too. In addition, a functionality of comparing the outputs of CUT and Golden circuit is also contained in the controller.

2) UART

This module is the medium for exchanging data between FPGA and Host computer. It receives commands from the computer to trigger the state machine of the controller, receives location information of target bits and uploads results to computer when a injection process is finished. The module's communication protocols are specified by designers and the major mission is to complete conversion of serial and parallel.

3) ICAP

ICAP is an internal configuration port works in the way as SelectMAP (a common FPGA configuration interface) except the separated data buses. The signals of clock (CLK) and input data (ICAP_I) are given by the controller, the output data (ICAP_O) is the original frame data and the state indicator (BUSY) tells if ICAP is working correctly. An important point needs attention is that Bit Swap must be done before shifting data into ICAP [14]. Given a 32-bits word in form of [MSB: LSB], the bit sequence should be rewritten in a specific regular, like Fig. 6 depicts. If this transform is ignored, both the two operations readback and reconfiguration will fail. For the data output from ICAP, such as the readback frame data, a Bit Swap cannot be forgotten either.



```
Original:              31:24  23:16   15:8  7:0
Bit Swapped :          24:31  16:23   8:15  0:7

Example(hex):
Original :             AB  C9   B5  39
Bit Swapped :          D5  93   AD  9C
```

Figure 6. Bit swapping.

4) CUT and golden circuit

As test circuits, CUT and Golden Circuit should achieve an identical function and be of a same topological structure in theory. They run under test vectors which are generated by the controller using a specific regular (LFSR or other complex algorithms [15]). If there is a discrepancy between the two circuit outputs, the current SEU inserted bit is identified as a SEU sensitive bit.

B. Program in Host Computer

The software program provides a user interface to perform SEU injection clearly. Main work it does is to send commands and location information to guide the flow and to save the injection results to files. Location information is frame address and bit offset obtained using the localization strategy described in Section II.

SEU injection flow follows these steps:

- Step 1: FPGA receives addresses of the target bitstream sent from the Host.
- Step 2: Readback a frame through ICAP and store the data for next step.
- Step 3: Modify the stored frame by flipping one bit.
- Step 4: Reconfigure the selected frame with the modified data, run and compare the circuit outputs.
- Step 5: Repair the frame and upload the result. Repeat step3 ~ step5 until all the target bitstream have been inserted.

IV. Results and Analysis

The validation platform is built on XQ5VLX110T. The experiment setup, results and analysis are explained in the following subsections.

A. Scenarios

The configuration bitstream is generated in ISE 14.7 and the software is written in C#. A DB9 interface and RS232 Serial Port Cable are used to transfer data between FPGA and Host computer. The system clock is 100MHZ and the serial rate is 115200. In this case, an 8-bit CLA is used as CUT and Golden circuit. Test vectors are pseudo-random numbers generated by LFSR.

B. Results of Critical-Bitstream Localization

First step of the proposed localization strategy implements the circuit layout with a series of specified placement constraints. The outcome is depicted in Fig. 7. In this case study, modules lay in detached regions: U1 represents the controller, U2 is the communication module, U3 and U4 are CUT and Golden Circuit.

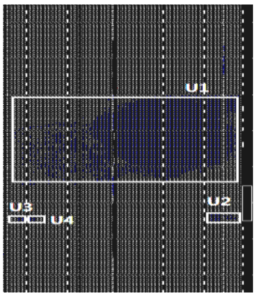


Figure 7. The circuit layout after using placement constraints.

Table I shows the logical utilization of these modules. The 8-bit CLA (CUT) is a combinational circuit and it is not in a large scale. Only 30 LUTs (4 CLBs) are occupied. Therefore the cursory localization of bitstream obtains 9216 bits. Follow the definition of critical bitstream, the precise localization obtains a target set of 2977 bits from the 9216 bits.

TABLE I. LOGIC ALLOCATION

Items	**Slice Logic Occupation (#)**		**Slice Logic Utilization (%)**	
	Registers	**LUTs**	**Registers**	**LUTs**
Controller	5,744	11,320	8.31%	16.37%
UART	88	124	0.13%	0.18%
CUT	—	30	—	0.05%
Golden Circuit	—	30	—	0.05%

C. Validations of SEU Injection

We inject SEUs on all the 9216 bits in advance and get 466 sensitive bits in all. This value is the basis for subsequent comparison between critical-bits injection and random-bits injection. As listed in *subsection B*, the localization strategy finds 2977 critical bits, so we also generated 2977 random bits (they are limited inside the 9216). Table II shows, there are 343 faults of the critical-bits injection and that takes up 73.6% in all sensitive bits. While the percentage of random-bits injection is only 20.8%. The improvement of 52.8% demonstrates the proposed two assumptions are reasonable and verifies the effectiveness of the localization strategy.

According Table II, for each bit, the operation time is about 48ms on average. As every injection transfers frame address, bit offset and injection result between FPGA and Host computer through Serial Port, this communication mechanism may limit the speed.

TABLE II. SEU INJECTION RESULTS

2977 injected bits, 466 sensitive bits		
Items	**critical bits injection**	**random bits injection**
Time(s)	144	143
Faulty bits(#)	343	97
Percentage (%)	73.6	20.8

V. CONCLUSION

This paper presents a critical-bitstream localization strategy and elaborates a validation platform of SEU injection. The localization strategy allows to insert SEU only on critical bits corresponding to CUT. Experiment with an 8-bit CLA as test circuit shows SEU injection on critical bits has a much higher fault rate than random bits injection. That suggests the proposed localization strategy is reasonable and it improves the efficiency. Each injection which includes communication with Host needs 48ms averagely. Some problems are considered to be settled in future work:

The serial rate may be the bottleneck of speed. For more complex test circuits, the time overhead will be a problem. So a faster communication mechanism is essential.

Apply more benchmark circuits to test the localization strategy and study SEU sensitivity.

ACKNOWLEDGMENT

Tingting Yu thanks to Prof. Lei Chen for the teaching and support during her Postgraduate period at the Department of Field Programmable Logic Array, Beijing Microelectronics Technology Institute, Beijing.

REFERENCES

[1] N. Jing, *et al.*, "Quantitative SEU fault evaluation for SRAM-based FPGA architectures and synthesis algorithms," in *Proc. 21st International Conference on Field Programmable Logic and Applications*, 2011, pp. 282-285.

[2] M. Straka, J. Kastil, and Z. Kotasek, "SEU simulation framework for xilinx fpga: First step towards testing fault tolerant systems," in *Proc. 14th Euro micro Conference on Digital System Design*, 2011, pp. 223-230.

[3] C. Hobeika, *et al.*, "Multi-abstraction level signature generation and comparison based on radiation single event upset," in *Proc. IEEE 20th International On-Line Testing Symposium*, 2014, pp. 212-215.

[4] E. Johnson, M. J. Wirthlin, and M. Caffrey, "Single-event upset simulation on an FPGA," in *Proc. International Conference on Engineering of Reconfigurable Systems and Algorithms*, 2002, pp. 68-73.

[5] J. Zhou, L. Chen, and S. Wang, "Precision fault injection method based on correspondence between configuration bitstream and architecture," in *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2013, pp. 267-267.

[6] M. S. Shirazi, B. Morris, and H. Selvaraj, "Fast FPGA-based fault injection tool for embedded processors," in *Proc. 14th International Symposium on Quality Electronic Design*, 2013, pp. 476-480.

[7] A. Sari and M. Psarakis, "A fault injection platform for the analysis of soft error effects in FPGA soft processors," in *Proc. IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, 2016, pp. 1-6.

[8] U. Kretzschmar, *et al.*, "Compact and fast fault injection system for robustness measurements on SRAM-based FPGAs," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2493-2503, 2014.

[9] C. Leong, *et al.*, "Fast radiation monitoring in FPGA-based designs," in *Proc. Conference on Design of Circuits and Integrated Systems*, 2015, pp. 1-6.

[10] G. Cieslewsk and A. D. George, "SPFFI: Simple portable FPGA fault injector," in *Proc. Military and Aerospace Programmable Logic Devices Conference*, 2009.

[11] E. A. Ghazaani, Z. Ghaderi, and S. G. Miremadi, "A non-intrusive portable fault injection framework to assess reliability of FPGA-

based designs," in *Proc. International Conference on Field-Programmable Technology*, 2013, pp. 398-401.

[12] J. L. Nunes, *et al.*, "FIRED-Fault Injector for reconfigurable embedded devices," in *Proc. IEEE 21st Pacific Rim International Symposium on Dependable Computing*, 2015, pp. 1-10.

[13] Xilinx, Inc. Constraints Guide. cgd v10.1. [Online]. Available: www.xilinx.com

[14] Xilinx, Inc, *Virtex-5 FPGA Configuration User Guide*, ug191, August 20, 2010.

[15] L. Cassano, "Analysis and test of the effects of single event upsets affecting the configuration memory of SRAM-based FPGAs," in *Proc. International Test Conference*, 2014, pp. 1-10.

Tingting Yu received her B.S. Degree in electronic information engineering from Tianjin University, Tianjin, P. R. China, in 2014. She is currently pursuing the M.S. Degree in electronic science and technology at Beijing Microelectronics Technology Institute, Beijing, P. R. China. Her current research interest is single event upset emulation in Field Programmable Gate Arrays.

Lei Chen received the B.S. Degree in electronics and information technology, the M.S. Degree in circuits and systems, the Ph.D. degree in computer science and technology from 1996 to 2006 in Northwestern Polytechnical University, Xi'an, P. R. China. His second Ph.D. degree in electronic science and technology was received in 2010 from Beijing Microelectronics Technology Institute, Beijing, P. R. China. Since 2008, he has been working with the very-large-scale integration design, test and reliability studying in Beijing Microelectronics Technology Institute as a professor and researcher. From September 2012 to September 2013, he was a visiting scholar in Electrical Engineering Department, University of California, Los Angeles. He has participated in more than 20 research projects provided by government or other research institutions. He is an author or co-author of 14 patents and more than 30 papers in national and international conferences.

Xuewu Li received the B.S. Degree in University of Electronic Science and Technology of China from microelectronics technology and the M.S. Degree in Harbin Institute of Technology from Integrated Circuit Engineering. Now he is a researcher of Beijing Microelectronics Technology Institute and his research interests include Integrated circuit design, FPGA architecture design and testing.

Shuo Wang received the B.S. Degree in electrical and information engineering from Harbin Engineering University in 2007, the M.S. Degree in microelectronics and solid electronics from Beijing Microelectronics Technology Institute, Beijing, P. R. China, in 2010. From February 2014 to February 2015, he was a visiting scholar in Electrical Engineering Department, University of California, Los Angeles. His current research interests include VLSI design, FPGA CAD algorithms and SEU mitigation techniques.

Jing Zhou received the B.S. Degree in integrated circuit design and integration system from Xi'an University of Electronic Science and Technology, Xi'an, P. R. China, in 2009. And she received the M.S. Degree in electronic science and technology at Beijing Microelectronics Technology Institute, Beijing, P. R. China, in 2013. She has been working with FPGA soft error effect estimation and FPGA EDA tool design in Beijing Microelectronics Technology Institute since 2013 as an engineer.