

Performance enhancement of serial based FPGA probabilistic fault emulation techniques

Ioana Mot, Oana Boncalo, Alexandru Amaricai
Computer Engineering Department
University Politehnica Timisoara
Timisoara, Romania
{oana.boncalo, alexandru.amaricai}@cs.upt.ro

Abstract—Serial based FPGA fault emulation schemes for probabilistic errors rely on a random number generator – which is used for generation of fault bits - and a shift register - used for placing the fault bits to their corresponding fault location. It has two advantages with respect to parallel solutions: lower cost and better accuracy. The main disadvantage is represented by the high emulation overhead: for each emulation clock cycle, a number of clock cycles equal to the number of fault locations is required to load the shift register. This paper presents a technique for FPGA probabilistic fault emulation which reduces the emulation overhead, at the expense of accuracy. It is based on pseudo-random permutations within the shift register, while maintaining the number of active fault bits. We obtain a performance improve of one order of magnitude, while we have a cost increase of 27% and lower fault modeling accuracy.

Keywords—FPGA; fault emulation; probabilistic faults; sub-powered circuits;

I. INTRODUCTION

Very low supply voltages, to near and sub threshold regions, are used in order to tackle the ever-increasing energy constraints for today's semiconductor devices [1]. However, this aggressive scaling of supply voltage, coupled to the process variations associated to the deep nanometer devices, lead to a low reliability of these circuits. A common fault model associated to sub and near threshold CMOS circuits is represented by the probabilistic fault model: a gate performs its logic operation with a probability less than 1 [2]. Therefore, reliability analysis becomes critical in the design phases of for these devices.

Reliability evaluation may be performed using analytical methods, simulation and FPGA emulations, and prototype based assessment. Regarding the simulation methods, these present a good trade-off between the fault modeling capability and the analysis overhead; furthermore, it comes in the early design phases with respect to prototype based evaluation. Regarding the probabilistic circuits reliability analysis, simulation methods present good observability; however, they have very high computational overhead, as a high number of simulations are required [4][5][12]. FPGA emulation may be used to address this short-coming.

Several FPGA emulation schemes have been developed for analyzing probabilistic faults, such as [4][5][12]. These approaches used the random number generator (RNG), either true (TRNG) or pseudo (PRNG), in order to generate the fault bits. The [4][5] approach uses a linear shift feedback register (LFSR) for each fault location. This type of approach has two disadvantages: very high cost overhead for a high number of fault locations and partially correlated faults, due to the use of LFSR. In [12], a serial emulation scheme has used a TRNG for fault bit generation and shift register for inserting the fault bits to their corresponding fault locations. The length of the shift-register is equal to the number of fault locations. In order to have uncorellated errors, a fault insertion phase is executed before each emulation clock cycle is performed. The fault insertion phase consists of the insertion of the entire shift register with the generated fault bits. Therefore, for each clock cycle of emulation, a number of clock cycles equal to the size of the shift register (or number of fault locations) is required to for fault insertion. Speed-up can be obtained by using multiple TRNG – shift-register modules for fault generation and insertion. However, even in this case, the emulation overhead is high, as a number of clock cycles equal to the length of the largest shift-register is required for fault insertion.

This paper proposes an FPGA fault emulation technique which improves the performance of the serial emulation scheme. It is based on pseudo-randomly permuting the fault bits within the shift register. It uses TRNG for fault bits generation and a modified shift-register for fault insertion and permutation. As in [12], a fault insertion phase is performed. It requires a number of clock cycles equal to the size of the shift register. However, after the fault insertion phase, multiple emulation clock cycles are performed, instead of 1. During these emulation cycles, the fault bits inserted during the insertion phase, are pseudo-randomly permuted within parts of the shift-registers. This way, increased performance is obtained, on the expense of fault modeling accuracy – during an emulation cycle the faults are not uncorellated - and higher cost – extra logic within the shift-register for permutations is required.

The paper is organized as follows: Section II presents the related work; Section III is an overview of the serial based FPGA probabilistic fault emulation framework; Section IV is dedicated to the proposed performance enhancement; while cost and performance are discussed in Section IV.

II. FPGA EMULATED FAULT INJECTION

The considerable size and performance offered by newer FPGA devices has led to increase utilization for emulated fault injection purposes [3][6][7][8][9]. An emulated fault injection scheme consists of the following modules: the design under test (DUT), the fault injection scheme, and the result analyzer and observation module. Regarding result analysis and observation, an important issue is represented by the communication with the host PC. Some approaches prepare the entire fault injection campaign on the PC, as well as the result analysis is performed there [7][9]. Other approaches minimize the role of the PC in the reliability analysis; both the campaign preparation and most of the result analysis is done on the FPGA [3][8]. Another important feature is the fault generation and insertion. This feature is heavily dependent on the fault model and have a major influence in the cost overhead of the fault emulation scheme. Usually, simple fault models, such as stuck-at, lead to a small cost overhead of the emulation scheme.

This paper uses the probabilistic independent fault model. Works addressing FPGA fault emulation for these types of faults have been presented in [4][5][12]. The approach in [4] use a RNG for each fault location. The RNG is implemented using LFSR; therefore, the fault generation is based on a PRNG. Although it provides very good performance, its main disadvantages are the high cost and correlation in time between errors due to pseudo-randomness fault generators. Furthermore, the cost problem is aggravated by seed initialization for the LFSRs – using the same seed for all LFSRs will result in the same error pattern in all fault locations. In [5], a serialization of the fault generation is performed by the addition of a serial binomial generation. The discussed time analysis indicates that one probabilistic fault is generated for every simulation experiment. Furthermore, the proposed approach requires two clock domains are used, which further increase the cost overhead.

This paper proposes a performance enhancement of the [12] approach. It uses pseudo-random permutations in the shift-register where the fault bits are inserted in order to avoid the loading of all the shift-register before each emulation cycle. The serial based FPGA emulation scheme is discussed in the next section, while the proposed enhancement is presented in Section IV.

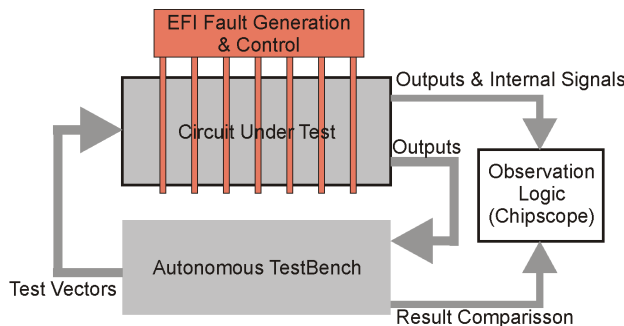


Figure 1 – FPGA Emulated Fault Injection Architecture Used in [12]

III. EMULATED FAULT INJECTION FRAMEWORK

We have developed the framework proposed in [12] in order to increase the performance of the analysis. It consists of the three main components – Fig. 1:

1. *Fault Generator and Control* – this module is responsible with the generation of the fault bits and their insertion in the corresponding fault locations;
2. *Autonomous testbench* – this module provides the appropriate test vectors, the error-free outputs and the result processing
3. *Observation logic* – the role of this module is to allow reliability metrics monitoring, as well as parameter changes for several fault injection campaigns.

The autonomous testbench is implemented using memory modules (internal BRAM or external memory) for storing the input vectors, as well as the golden outputs. The outputs of the fault injected module are compared with the correct outputs stored in the memory; based on this comparison, reliability metrics are derived.

Regarding the observation logic, our solution uses Xilinx Chipscope Pro logic analyzer [10]. It is based on the following dedicated cores: ILA – Integrated Logic Analyzer – which allows signal observation and triggers; ICON – Integrated CONtroller – which provides the communication between JTAG interface and the ILA core; and VIO – Virtual Input/Output – which provides the interface to monitor and to drive signal from the testbench. Our solution is based on Xilinx Chipscope because it provides the appropriate cores for monitoring, triggering and driving signals, which are optimal in terms of cost and performance for Xilinx FPGA devices.

Fault bits generation is achieved using a TRNG developed by Xilinx (Fig 2) [11]. It consists of a XOR based ring oscillator and a Linear Hybrid Cellular Automata. Regarding the randomness of numbers generated, this design is reported to have passed most of the DieHard tests used for cryptographic applications. The generated random number is compared with a probability constant (which value is dependent on the probability of fault); the appropriate fault bit is thus generated.

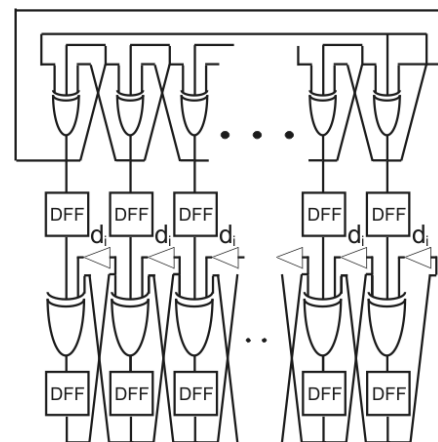


Figure 2 – XOR based TRNG [11]

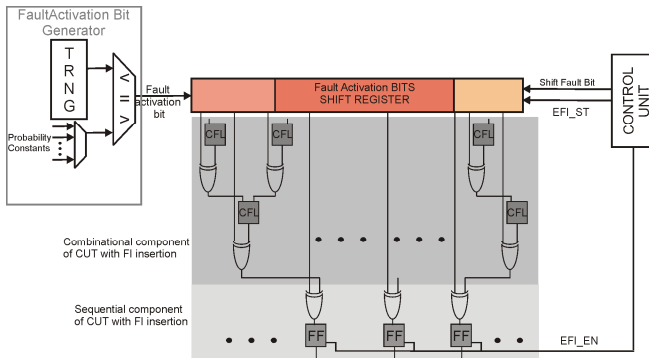


Figure 3 – Serial Scheme for Emulated Fault Injection (CFL – combinational fault location; FF – flip-flop)[12]

The insertion of the fault bit is performed using a shift-register. The length of the shift register is equal to the number of fault locations. A XOR gate is used for fault emulation – a logic XOR is performed between the fault bit stored in the shift-register and the output of the injected module. In order to have uncorrelated faulty pattern, the shift-register is loaded before each emulation cycle. Thus, a number of clock cycles equal to the number of fault locations are required before each emulation clock cycle. The fault generation and insertion is depicted in Fig. 3.

The main advantages of this approach with respect to [4][5] are lower cost and a better fault modeling accuracy (as the faults are uncorellated both in time and location). The main disadvantage is represented by the low emulation performance, due to the fault insertion phase. In order to reduce the fault insertion time, k TRNG – shift registers may be used. Thus, the size of the shift-registers is reduced by a factor of k , which significantly reduces the overall loading time.

IV. PERFORMANCE ENHANCEMENT OF EMULATED FAULT INJECTION FRAMEWORK

The main performance penalty in the serial based fault emulation scheme is represented by the fault insertion phase required before each fault emulation clock cycle. We improve the performance by eliminating several fault insertion phases. This is achieved by applying pseudo-random permutations on the shift-register used for fault insertion. The proposed fault emulation is performed as follows:

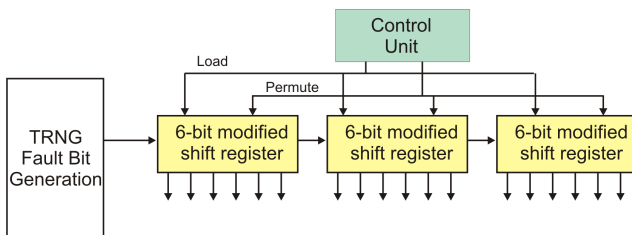


Figure 4 – Modified Serial Scheme for Emulated Fault Injection

1. Fault insertion – the shift-register is loaded with the TRNG generated fault bits; this phase is identical to the one used in the serial based emulation scheme
2. Emulation phase – n emulation clock cycles are performed; the fault bits within the shift register are pseudo-randomly permuted each clock cycle.

Using this approach, a performance improvement of up to n times is obtained with respect to the approach in [12]. Modified shift-registers which perform the pseudo-random permutations are used. In order to preserve the probabilities obtained for the fault bits, after each random permutation the number of 1's within the shift register remains the same (e.g. a "100000" vector will permute in "000010" or "010000" or "000001" or "001000" or "000100"). The new fault emulation scheme is presented in Fig. 4.

The increase in performance is achieved at the cost of higher cost (due to extra logic required to perform the permutations) and loss in the fault modeling accuracy. The latter is due to the fact that, except the emulation clock cycle after the fault insertion phase, correlations between faults do exist. In order to reduce the cost of the modified shift-register, pseudo-random permutations are performed on groups of 6-bits. Thus, the initial shift-register is replaced with a module consisting of 6-bit modified shift-registers (Fig.4). One reason to use blocks of 6 bits is represented by the modern FPGA structure, for which each logic element is composed of 6-input LUT and a D flip-flop. Each 6-bit block within the fault insertion module performs different types of permutations (e.g. one block performs the permutation from "001100" to "101000" while other block perform the permutation from "001100" to "010010").

V. RESULTS & DISCUSSION

We have performed emulated fault injection on a 7-level 96 bits pipelined barrel shifter. This circuit is used for routing messages in LDPC decoders. The reliability analysis of this module is important in the context of developing reliable LDPC decoders using unreliable components [13]. The considered fault locations are the inputs for the pipelined registers. Therefore, the number of fault locations is equal to 672. For each level within the barrel shifter we have used one TRNG – shift-register module for fault injection.

The fault emulation schemes have been implemented on a Xilinx Virtex-5 VLX-50T, speed grade -2, device. Xilinx ISE 14.7 has been used for synthesis and implementation.

TABLE I – IMPLEMENTATION RESULTS FOR BARREL SHIFTER

	Cost	Overhead
Modified Serial	3260 LUT-FF pairs 4 BRAM	470%
Base Serial [12]	2560 LUT-FF pairs 4 BRAM	370%
Barrel shifter	690 LUT-FF pairs	0%

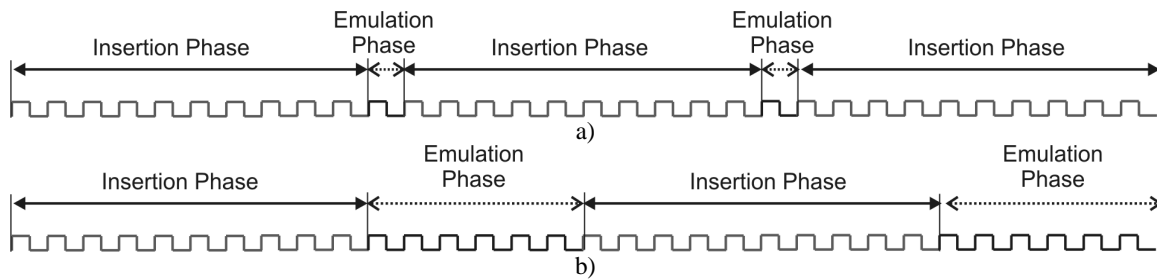


Figure 5 – Insertion and emulation phase for the serial FPGA fault emulation scheme (a) and modified serial FPGA fault emulation scheme (b)

Table I shows the cost of the fault emulation schemes applied to these circuits. We observed that the proposed one has a cost increase of 27% with respect to the serial based fault emulation scheme proposed in [12]. The overhead is around 370% for the approach in [12] and 470% for the proposed approach.

Regarding performance, Fig 5 depicts the difference between the serial based fault emulation scheme proposed in [12] (a) and the proposed one (b) for 10 fault locations. For the analyzed barrel shifter, the serial based fault emulation scheme requires 96 clock cycles for fault insertion before one emulation clock cycles. The proposed solution uses 20 clock cycles of emulation before a new fault insertion phase (consisting of 96 clock cycles) is performed. Therefore, 18 times less clock cycles are required for our approach with respect to the [12] solution.

The faults are uncorrelated only in the first clock cycle of the emulation phase. In the following clock cycles of the emulation phase, the faults become correlated. This is due to the pseudo-random permutations within the shift registers. Therefore, the fault modeling capability is reduced with respect to [12].

The full parallel approach in [4] using LFSR as RNG report an overhead of around 1000% for the ISCAS 89 s1196 benchmark circuit, for 18 fault locations, and around 8000% for flip-flops and 1000% for combinational LUT for the b14 benchmark circuit, for 220 fault locations.

VI. CONCLUSIONS

This paper presents a performance enhancement approach of the serial based FPGA probabilistic fault injection. It overcomes the low performance of the [12] approach by removing the vast majority of the fault insertion phases. This is achieved by applying pseudo-random permutations within the fault bits shift register during the emulation clock cycles. This way, at least one order of magnitude less clock cycles are required for fault emulation. In order to reduce the cost, we have applied pseudo random permutations for 6-bits blocks. The modified shift register is composed of these 6-bit permutation blocks. The main disadvantages of the proposed fault emulation scheme is represented by higher cost with respect to the serial based approach of 27%, as well as lower probabilistic fault modeling capability.

ACKNOWLEDGMENT

This work has been supported by the Seventh Framework Program of European Union under Grant Agreement 309129, project *i-Risc*. We would like to thank Xilinx for the software tools provided through the Xilinx University Program.

REFERENCES

- [1] H. Khaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, S. Bokhar "Near Threshold Voltage Design: Opportunities and Challenges" Proc. Design Automation Conference (DAC), 2012, pp. 1153-1158
- [2] A. Bhanu, M.S.K. Lau, K.V. Ling, V.J. Mooney, A. Singh "A More Precise Model of Noise Based PCMOs Errors" Proceedings 5th Int. Symp. On Electronic Design, Test and Application (DELTA), pp 99-102, 2010
- [3] C. López-Ongil, M. García-Valderas, M. Portela-García, L. Entrena, "Autonomous Fault Emulation: A New FPGA-Based Acceleration System for Hardness Evaluation," IEEE Trans. On Nuclear Science, Vol. 54, No. 1, pp. 252-261, Feb. 2007
- [4] D. May, W. Stechele, "An FPGA-based Probability-aware Fault Simulator", International Conference on Embedded Computer Systems (SAMOS), 2012.
- [5] D. May, W. Stechele, "A resource-efficient probabilistic fault simulator, " 23rd International Conference on Field Programmable Logic and Applications (FPL), September, 2013.
- [6] A. Ejlali, S. G. Miremadi, "Error propagation analysis using FPGA-based SEU-fault injection," Microelectronics Reliability, vol. 48 pp. 319-328, June 2008.
- [7] P. Civera, L. Macchiarulo, M. Rebaudengo, M. S. Reorda, and M. Violante, "An FPGA-Based Approach for Speeding-Up Fault Injection Campaigns on Safety-Critical Circuits," Journal of Electronic Testing: Theory and Applications 18, Kluwer Academic Publishers, pp. 261-271, 2002.
- [8] M. S. Shirazi, B. Morris, H. Selvaraj, "Fast FPGA-Based Fault Injection Tool for Embedded Processors," 14th Int'l Symposium on Quality Electronic Design, 2013.
- [9] M. Sauer, V. Tomashevich, J. Müller, M. Lewis, A. Spilla, I. Polian, B. Becker, and W. Burgard, "An FPGA-Based Framework for Run-time Injection and Analysis of Soft Errors in Microprocessors," IEEE 17th International On-Line Testing Symposium, July, 2011.
- [10] <http://www.xilinx.com/tools/cspro.htm>. Xilinx ChipScope.
- [11] C. Baetoni "Method and Apparatus for True Random Number Generation" US Patent 7389316, 2008
- [12] O. Boncalo, A. Amaricai, C. Spagnol, E. Popovici "Cost effective FPGA probabilistic fault emulation" Norchip Conference, Oct, 2014
- [13] C.L.K. Ngassa, V. Savin, D. Declercq "Min-Sum-based decoders running on noisy hardware" IEEE Global Communication Conference, 2013