Sub-Threshold CMOS Circuits Reliability Assessment Using Simulated Fault Injection Based on Simulator Commands

Sergiu Nimara, Alexandru Amaricai, Mircea Popa
Politehnica University Timisoara, Romania
sergiu.nimara@student.upt.ro, alexandru.amaricai@cs.upt.ro, mircea.popa@upt.ro

***Abstract*—Lowering the supply voltage below the threshold voltage of the transistors brings important benefits regarding the power consumption. However, the main issue of sub-threshold CMOS circuits is the abrupt reliability decrease. This paper proposes a simulated fault injection approach for reliability assessment of gate-level designs supplied at low voltages. The proposed method uses previously determined probabilities of failure of sub-threshold logic gates in order to perform fault injection campaigns based on simulator commands and scripts for several types of adders. The overhead of this method is 6x – 30x with respect to the fault-free circuit simulation time. We have validated our technique's accuracy by comparing the results with those of equivalent fault injection methodologies, based on HDL code alteration.**

I. INTRODUCTION

Reducing the power consumption of digital integrated circuits has become one of the hot topics in the last years and it is critical for the survival of the semiconductor industry. The method which has been employed more and more often lately is represented by the reduction of the supply voltage of the transistors, which brings important power savings by lowering both the static and the dynamic components of the total power. Transistors supplied at a voltage only slightly higher than the threshold voltage, in the so-called near-threshold region, are proved to reduce the total power by an order of magnitude. However, the minimum energy point of the circuit usually occurs in the sub-threshold region, where the supply voltage is lowered below the threshold voltage of the transistor [1][2].

The significant power savings obtained in the sub-threshold region are accompanied by two important drawbacks: the performance loss and the dramatic decrease of the reliability parameters. Nanoscale transistors operating at very low supply voltages have an increased susceptibility to process-voltage-temperature (PVT) variations, which are usually caused by low I_on / I_off ratios, lithographic irregularities, varying dopant concentrations or heat flux fluctuations [3]. These factors impose analyzing the behavior of these circuits in a stochastic manner, knowing that the correct logic value at the output of a gate will be obtained with a probability less than one.

The dependability parameters of digital systems have been thoroughly investigated in the literature, the most widely used approach being based on fault injection [4]. Simulation-based fault injection represents the preferred method for performing dependability assessment because it permits the identification of design flaws in the early stages of the development of a digital system. Simulated fault injection (SFI) techniques have been successfully implemented for a wide range of systems, from SRAM based FPGAs [5] to quantum circuits [6].

This paper proposes a SFI technique for the reliability analysis of digital circuits functioning in the sub-threshold regime. During previous work [7][8], the probabilities of failure for logic gates supplied at very low voltages have been derived using SPICE simulations and they have been used for a gate-level SFI techniques based on code alteration (mutants and saboteurs). In contrast with the previous paper, this approach doesn't interfere with the Verilog hardware description language (HDL) code of the system, but it uses simulator commands and scripts to inject errors at the outputs of the logic gates contained by the fault-free design.

The errors are injected during several simulation campaigns for ripple carry adders (RCA) and carry select adders (CSeA), according to the previously determined probabilities of failure. The novel approach is validated by comparing the reliability parameters with the ones obtained in paper [7]. Our implementation has been realized using two different methods: one that uses a dedicated Verilog module for the generation of the moments when faults are injected and one based entirely on simulator commands. The main advantage of the second method is that it brings no additional overhead to the HDL code of the design being tested, so it requires lower computational resources. The simulation time claimed by each campaign is measured and the simulation overhead of this novel methodology is analyzed. Although higher than the mutant-based fault injection methodology, the temporal cost associated with the proposed solution is affordable for small and medium size circuits simulated on modern computers.

This paper is organized as follows: Section II explains the reliability issues encountered by sub-threshold circuits and the methods used for an effective quantification of these problems. Section III describes the phases and the particularities of the proposed fault injection technique, while the results and their interpretation are discussed in Section IV. The last part of the paper, Section V, analyzes some concluding aspects and gives directions of future development.

II. Reliability Issues in Sub-Threshold CMOS Circuits

Shrinking geometries, low supply voltages and high frequencies of operation all contribute to an increase in the number of faults occurrences. Among the three types of faults, permanent, intermittent and transient, the faults from the third category are responsible for over 85% of all computer failures [11].

As transistors scale more and more, down into the nanometer region, each technology generation manifests an increased vulnerability to process variations. Within-die (WID) process variations are caused by both systematic effects (i.e. lithographic irregularities) and random effects (i.e. fluctuations in the dopant concentration) [3]. These variations have a negative impact mainly on two process parameters, the threshold voltage and the effective channel length, which affect the transistor switching speed and the static power consumption [3].

The fact that the transistor switching speed is affected by variation means a reduction of the operating frequency of the circuit. The delay constraint must be set accordingly to the time window required by a transistor to perform the correct switch. If the delay is too small, the probability that the transistor switched correctly decreases. In the sub-threshold region, this delay constraint means a frequency of operation which is reported in the range of tens of kHz to a few MHz [1][2]. This is the reason why the probability of failure for a logic gate supplied at very low voltages must be expressed as a function of the delay. These probabilities have been derived in [7] for basic logic gates affected by PVT variations, supplied at 0.25, 0.3 and 0.35 V, respectively. The results in [7] show the trade-off which can be made between performance and failure rate. A delay greater than 3 ns brings an important performance penalty (low frequency), but is associated with a probability of correctness of approximately 100 %, while a smaller delay means good performance with high failure rate.

III. Gate Level Simulated Fault Injection Based on Simulator Commands

In order to evaluate the dependability attributes of digital systems, fault injection represents a wide-spread approach. The three main categories of fault injection techniques are: physical or hardware implemented fault injection (HWIFI), software-implemented (SWIFI) and simulation-based [9]. The techniques based on simulation have the advantage that they can be used to evaluate the circuit under test (CUT) during the design phase, using computer-aided design (CAD) software, bringing important benefits in terms of time and costs. For an effective simulation, accurate input parameters, validation of the results and suitable fault models and fault patterns are required. SFI techniques fall into two main categories, according to [9]: approaches that require modifications of the HDL code (based on mutants and saboteurs) and

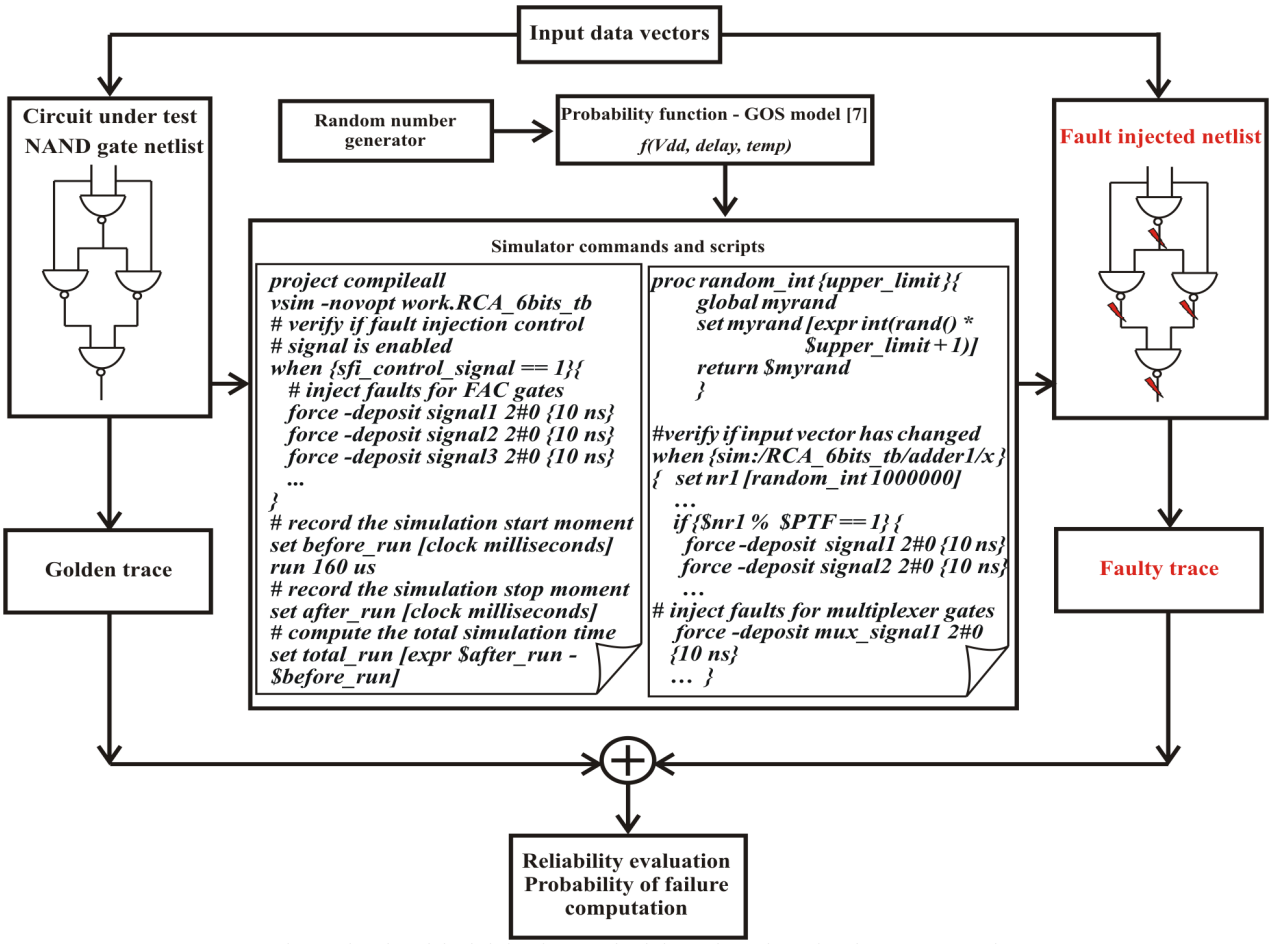

Fig. 1 Simulated fault injection methodology, based on simulator commands

TABLE I.
SIMULATION RESULTS FOR 6-BIT RCA

Vdd [V]	Delay [ns]	Gate failure probability [%]	Result bits failure probabilities [%]							Circuit failure prob. [%]	Circuit failure prob. obtained in [7] [%]	Run-time [s]
			6	5	4	3	2	1	0			
0.35	1.5	0.2827	3.0187	5.3125	4.9875	4.3438	3.6938	1.6563	1.2375	11.5813	10.1625	3
0.30		1.9460	17.9750	29.7813	28.3562	23.1563	19.8813	10.6563	9.7188	62.0187	50.0875	15
0.25		10.1300	21.6375	14.1938	28.0375	49.1938	43.9937	37.0250	35.1625	90.7062	90.1625	70
Correct circuit		0	0	0	0	0	0	0	0	0	0	0.5

TABLE II.
SIMULATION RESULTS FOR 6-BIT CSEA

Vdd [V]	Delay [ns]	Gate failure probability [%]	Result bits failure probabilities [%]							Circuit failure prob. [%]	Circuit failure prob. obtained in [7] [%]	Run-time [s]
			6	5	4	3	2	1	0			
0.35	1.5	0.2827	3.7437	6.8438	5.2687	4.0938	3.6938	1.6563	1.2375	12.2063	12.1687	3
0.30		1.9460	20.9875	38.1000	30.5125	22.0562	19.8813	10.6563	9.7188	65.0250	55.5938	14
0.25		10.1300	72.5750	81.0750	68.6000	43.7625	43.9937	37.0250	35.1625	93.3875	91.0875	63
Correct circuit		0	0	0	0	0	0	0	0	0	0	0.5

approaches that don't require any code instrumentation (simulator commands and scripts). The second category uses the commands of the simulator at simulation time to alter the value or timing of the signals and variables comprised by the model, without modifying the HDL code.

For this paper, we have used a SFI methodology based on simulator commands and scripts [10]. Like other SFI methodologies, it consists of three main phases: the set-up phase, the actual simulation and the results analysis phase. The block diagram of the proposed methodology is depicted in fig. 1. During the set-up phase, the script is prepared according to the parameters of the faults that are desired to be injected in the design. The probability of occurrence of a fault on the output of a logic gate depends on the Vdd, temperature and delay of the gate, as stated in [7]. The proposed method injects faults at the output of the gates and the transitions that occur at the inputs of the gate are not taken into account. In order to generate the moment when a fault is inserted in the design, according to the desired probabilities, we have used two equivalent methods, which have the same results:

(1) For the first method, we have created a fault injection module in Verilog HDL which asserts a series of control signals in the design whenever a fault must be injected, according to the gate-specific probabilities of failure considered in [7] for the Gate Output Switching (GOS) model. This unit comprises several random number generator modules and probability function modules, which are used to generate the failure condition by asserting one of the control signals. Each control signals generation step is triggered by the presence of a new vector at the inputs of the adder, which announces the beginning of a new run. These signals are parsed by a script and each time a fault injection control signal is asserted, the script alters the logic value of the output of one or more gates in the design.

This method brings an auxiliary overhead to the Verilog code because the random number generation and the probability calculation are performed by an additional Verilog module. A sequence of the actual TCL script that is controlling the simulation campaigns for this case is presented as follows. The time related parameter found in the "force" commands ({10ns}, fig. 1) shows the transient nature of the faults being injected in the design. The script also contains the code lines required to measure the total simulation time of one campaign.

(2) The second method moves the computation required for random number generation and probability calculation from the Verilog code to the TCL script code, so it brings an additional overhead to the commands executed by the simulator.

A part of the TCL script developed for this method is presented on the right side of fig. 1. The random_int procedure is used to generate a random number between 1 and the upper_limit parameter. PTF represents the probability to failure parameter. The beginning of a new step, which comprises the generation of new random numbers, is triggered by the presence of a new vector at the inputs of the adder. This condition is verified by the instruction: when {sim:/RCA_6bits_tb/adder1/x }.

The second phase of the SFI methodology consists in the actual simulation of the circuit under test and the results analysis, which is performed immediately after each run. A Verilog testbench module controls the actual simulation and the results interpretation: it establishes the total number of runs, it applies the input vectors for each run and it compares the faulty trace obtained at the outputs of the circuit with the golden trace obtained for a fault-free run. The equations for the dependability parameters are also implemented in the testbench module.

IV. SIMULATION RESULTS

We have performed multiple simulation campaigns for several types of adders, described in Verilog HDL at gate-level, using only NAND gates. The results of these campaigns are included in tables I and II. The simulations have been carried out using Modelsim 10.05 SE commercial simulator on desktop computer with Intel Core i5 at 3.2 GHz and 4 GB of main memory with Windows 8.1 OS. We have applied a number of 16000

test vectors (16 input vectors * 1000 runs) at the inputs of each adder, in order to maintain the compatibility with the simulation settings applied in [7]. For easily validating the proposed methodology by confronting it with the solution presented in [7], we have analyzed two dependability parameters: the probability of failure for each bit of the result and the probability of failure of the whole output vector representing the result. Moreover, the simulation time required by each campaign was measured.

The proposed methodology has been applied for 6-bit ripple carry adders (RCA) and carry select adders (CSeA). We have kept the same 6-bit configuration as the one described in [7], in order to perform the results comparison between the two methods on the same circuits.

The first simulation campaign has been carried out for a 6-bit ripple carry adder (RCA) configuration with the same delay of 1.5 ns for each gate of the design. The supply voltage was also considered the same for each gate of the design, taking the following values: 0.25, 0.30 and 0.35 V respectively. The results are depicted in Table I.

Regarding the results obtained for the 6-bit RCA, we can observe that the bit-independent failure probabilities and the overall probability of failure of the result are approximately the same, only slightly higher than the ones obtained in [7] for the Gate Output Switching (GOS) model. The differences in the probabilities are justified by the fact that the GOS model injects a fault for a gate output, with a given probability, only when the gate performed a switch (from logic 0 to logic 1 or vice-versa). Our new methodology injects more faults in the considered time window because it employs the same probabilities but it doesn't account for gate switching.

The second simulation campaign has been carried out for a 6-bit carry select adder (CSeA) configuration with the same delay of 1.5 ns for each NAND gate. The results for the CSeA are presented in Table II.

The simulation of the correct RCA and CSeA adders both require 0.5 s. According to tables I and II, a simulation campaign of 1000 runs, based on simulator commands and scripts requires between 3 and 70 seconds, depending on the number of faults that must be injected in the design. For usual gate probabilities of failure, found in practice, the simulation overhead is 6x – 30x with respect to the gold circuit. The authors in [9] expect that the techniques based on simulator commands will provide the lowest temporal cost associated with the simulation. However, our measurements suggest that the temporal cost of this method is higher with respect to the one required by the mutant-based method, but it still represents an affordable value for simulations running on modern computers.

V. CONCLUSIONS

This paper addresses the reliability issues of sub-threshold CMOS circuits and proposes a novel approach for the reliability assessment of small and medium digital systems. The proposed methodology consists in simulated fault injection techniques based on simulator commands and scripts. The methodology is validated by confronting the results with a similar technique, based on code alteration, which was developed in [7].

The main advantages of the proposed technique are the easiness in the simulation set-up process and the flexibility. The technique was implemented by two different means and the one based entirely on simulator commands has the supplementary advantage of producing no overhead for the Verilog code of the CUT. The methodology has been applied for several types of adders in order to prove that the simulation time is reasonable for small and medium gate-level netlists and the obtained probabilities of failure are similar to the results obtained in [7].

ACKNOWLEDGMENT

This work has been supported by the Seventh Framework Programme of European Union under Grant Agreement 309129, project i-RISC – "Innovative Reliable Chip Design from Low Power Unreliable Components".

REFERENCES

[1] B. H. Calhoun, A. Wang, N. Verma and A. Chandrakasan, "Sub-threshold design: the challenges of minimizing circuit energy," Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED), Germany, 2006.

[2] Alice Wang, Anantha Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," IEEE Journal of Solid State Circuits, vol. 40, no. 1, 2005.

[3] U. R. Karpuzcu, K. B. Kolluru, N. S. Kim, J. Torrellas, „VARIUS-NTV: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages," 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2012.

[4] J. Gracia, J. C. Baraza, D. Gil, P.J. Gil, "Comparison and application of different VHDL-based fault injection techniques," Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), 2001.

[5] M. M. Ibrahim, K. Asami, M. Cho, "Evaluation of SRAM based FPGA performance by simulating SEU through fault injection," 6th International Conference on Recent Advances on Space Technologies (RAST), 2013.

[6] Oana Boncalo, Mihai Udrescu, Lucian Prodan, Mircea Vladutiu, Alexandru Amaricai, "Simulated fault injection for quantum circuits based on simulator commands," 4th International Symposium on Applied Computational Intelligence and Informatics (SACI), 2007.

[7] Alexandru Amaricai, Sergiu Nimara, Oana Boncalo, Jiaoyan Chen, Emanuel Popovici, "Probabilistic gate level fault modeling for near and sub-threshold CMOS circuits," Proceedings of the 17th Euromicro Conference on Digital System Design, August 2014.

[8] Jiaoyan Chen, Christian Spagnol, Satish Grandhi, Emanuel Popovici, Sorin Cotofana, Alexandru Amaricai, "Linear compositional delay model for the timing analysis of sub-powered combinational circuits," Proc. International Symposium on VLSI (ISVLSI), 2014.

[9] J. C. Baraza, J. Gracia, D. Gil, P.J. Gil, "Improvement of fault injection techniques based on VHDL code modification," Tenth IEEE International High-Level Design Validation and Test Workshop, 2005

[10] Mentor Graphics ModelSim User's Manual, available at: http://www.microsemi.com/document-portal/doc_view/131619-modelsim-user

[11] G. S. Choi, R. K. Iyer, V. A. Carreno, "Simulated fault injection: A methodology to evaluate fault tolerant microprocessor architectures," IEEE Transactions on Reliability, vol. 39, no. 4, 1990