Probabilistic Saboteur-based Simulated Fault Injection Techniques for Low Supply Voltage Interconnects

Sergiu Nimara Alexandru Amaricai Oana Boncalo Mircea Popa
"Politehnica" University of Timisoara
Timisoara, Romania
sergiu.nimara@gmail.com, alexandru.amaricai@cs.upt.ro, oana.boncalo@cs.upt.ro, mircea.popa@rectorat.upt.ro

Abstract— **Probabilistic behavior of logic gates represents one of the main reliability problems associated to CMOS circuits supplied at very low supply voltages. This paper aims to analyze the impact of probabilistic faults in interconnects, by means of HDL saboteur-based simulated fault injection (SFI). We propose four types of saboteurs: the simplistic probabilistic type, a switching type - aware and two data dependent types. We have analyzed the behavior of the Wishbone bus in the presence of probabilistic errors. Several sets of simulations have been performed, by injecting probabilistic faults on address, control signals and data components of the bus. The performed simulations indicate that the simulation time for a SFI campaign is 1.7x higher with respect to the gold circuit.**

Keywords—simulated fault injection, saboteurs, interconnects, probabilistic circuits;

I. Introduction

The quest for lower power consumption has led to dramatic down scaling of the supply voltage to sub and near threshold regimes. Coupled with the scaling of transistor sizes to nanometer levels and with process and temperature variations, this has led to important reliability issues in logic devices. These logic circuits exhibit a probabilistic behavior. The probabilistic behavior may become more acute in interconnects; this is due to systematic and random process variations, including metal, dielectric barriers and low-*k* dielectrics, combined with crosstalk noise [1]. Therefore, in addition to the effects caused by transient faults at gate-level, the probabilistic error occurrence in interconnects must be taken into account.

The reliability attributes of digital systems can be efficiently determined using fault injection techniques, which are classified in three main categories: hardware-based, software-based and simulated fault injection [2],[3],[4]. The simulation-based fault injection technique is preferred over the others because it offers the possibility of an early diagnosis of the circuit under test (CUT), during the design phase [4]. Simulated fault injection can be performed using techniques which do not require source code intervention (based on simulator commands and scripts) and techniques which alter the CUT's source code (saboteurs and mutants). Regarding the former, these rely heavily on the employed HDL simulator's capability and limited fault modeling capability. Regarding the latter, the overhead given by the source code modification is compensated by the increased fault modeling capability.

This paper proposes probabilistic saboteur based techniques for reliability analysis of interconnects. We focus on reliability issues of signals transmitted on interconnects and not on logic gates and memory elements. We propose several types of saboteurs. The most simple relies on performing a probabilistic bit-flip on a signal of the interconnect. The most accurate takes into consideration data dependency: the probabilities for each signal depend on the data values transmitted on the interconnect. This type of saboteur captures in the most accurate way due to the importance of the crosstalk in interconnects' behavior (crosstalk is data dependent). We have performed our analysis on the open-source Wishbone bus. We have analyzed the impact of probabilistic faults on different types of signals of the bus: data signals, address lines, control and handshaking signals.

This paper is organized as follows: Section II is dedicated to reliability issues of interconnects; the probabilistic saboteur-based simulated fault injection technique is described in Section III, while Section IV presents the simulation campaigns. Some concluding observations are stated in the last section of this paper.

II. Reliability Issues In Interconnects

The main factors that lead to reliability issues in interconnects are process variation and crosstalk induced faults. Regarding the process variations, the most frequent forms of it are represented by: device geometry variations, device material and electrical parameter variations, interconnect geometry and material parameter variations [5]. These variations will have an effect on the metal thickness or length, dielectric thickness, contact and via size, metal resistivity or dielectric constant. Thus, the resistance,

978-1-4799-4994-6/14/$31.00 ©2014 IEEE

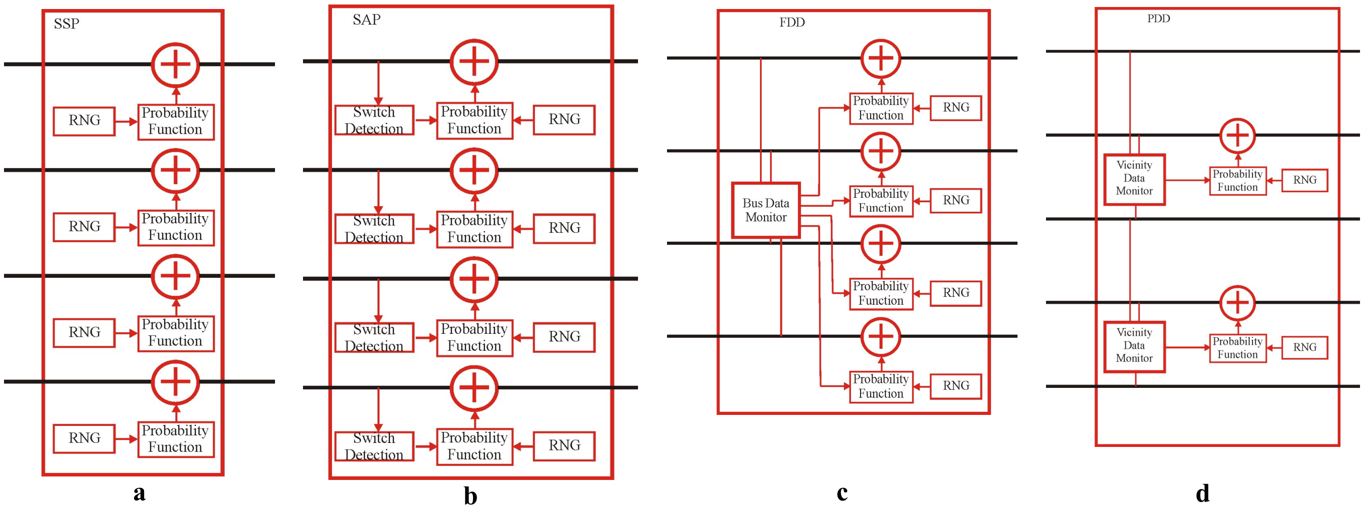

Fig. 1 - The saboteurs' architectures according to four fault models (a – SSP, b – SAP, c – FDD, d – PDD)

capacitance or inductance parameters of a wire are affected. Process variation in interconnects may alter the timing characteristics of the signals. Thus, an erroneous result at the moment when a certain signal is sampled may appear due to increased resistance or ground capacitance of the wire.

Crosstalk faults are most probably the result of an inappropriate interconnect routing scheme, rather than manufacturing defects [6], and they are strongly data-dependent. For the interconnection lines, cross-talk induced faults result from an undesired inductive or capacitive coupling between two or more signal lines, producing both timing alterations and / or noise (like glitches) on those signals [7]. These parasitic couplings determine an energy transfer from one wire to another, depending on the driver strength and they result in crosstalk faults [6]. The authors in [6] realize a classification of crosstalk faults into crosstalk induced glitches and delays. Crosstalk induced glitches appear on a static victim (affected) line when one or more aggressor lines switch their logic value, while crosstalk induced delays occur when aggressor and victim signals change their logic state simultaneously [6]. The most dominant effect is represented by the capacitive crosstalk: this affect only the neighboring line [8]. The inductive crosstalk has a smaller influence with respect to the capacitive one; however, the inductive effects may span across multiple lines [6],[11].

In this paper, we address the probabilistic occurrence of faults caused by either process variation or crosstalk effect, affecting the interconnects of a digital system. The circuit under test is described in the next section, along with the fault injection method employed.

III. Probabilistic Saboteurs for Interconnects

Simulated fault injection can be achieved without source code intervention (such as the ones based on simulator commands) or by utilization of techniques which alter the CUT’s HDL description. The techniques which rely on source code modification present two advantages: (i) they are independent of the HDL simulator (ii) they present high fault modeling capability. Two techniques are widely used for simulated fault injection: mutants and saboteurs. The mutant represents a component description which replaces the correct one. In VHDL, mutants are implemented using multiple architectures for the same entity. In Verilog, the mutant represents another module description. The saboteur is a special component (entity or module) which alters the value or timing characteristics of a signal [4]. The goal of our analysis is to perform reliability evaluation (by modifying the values) for different groups of signals contained by the interconnect. This makes the saboteur the natural candidate for our analysis. Our methodology has been implemented in Verilog; however, it can be easily adapted to VHDL. Several types of saboteurs have been proposed, such as [4], [9]: serial simple unidirectional saboteur, serial simple bidirectional, serial complex saboteur, serial complex bidirectional saboteur, *n*-bit unidirectional serial saboteur, *n*-bit bidirectional serial saboteur, parallel saboteur. According to this classification, the saboteurs employed in this paper can be considered *n*-bit unidirectional serial saboteurs.

For probabilistic interconnects, we have developed four types of saboteurs:

1. *Standard Signal Probabilistic* (SSP) saboteur. It represents the simplest one because it only flips the logic value of a certain signal with a given bit-independent probability. It doesn’t take into account the last type of transition that took place on that line, nor the data pattern. The SSP model-based saboteur contains a fault insertion module, which is triggered according to the desired probability of failure and to the output provided by a random number generator.
2. *Switching-Aware Probabilistic* (SAP) saboteur. This model considers probabilistic behavior for a signal only when switching is taking place. It models accurately timing faults: the switching for a line does not respect a given timing constraint. The most simplistic type of SAP considers the same probability for both types of

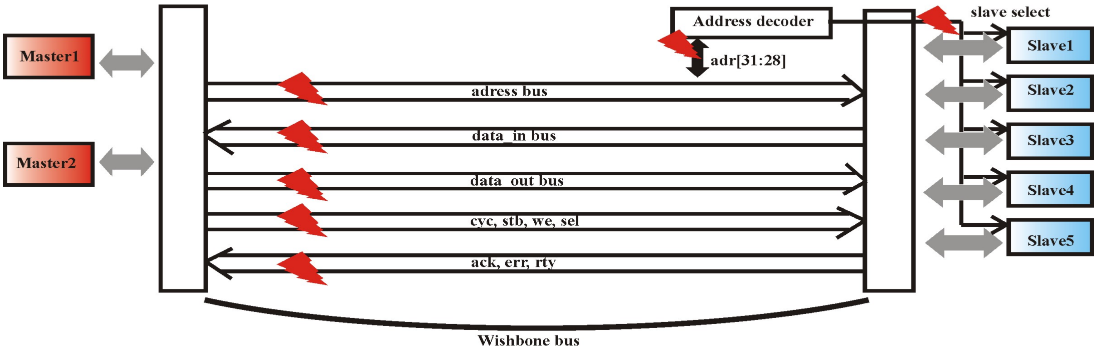

Fig. 2 - The wishbone's signal groups that are the subject of fault injection campaigns

switching; a more accurate considers different probabilities for charging and discharging processes. The architecture for this saboteur contains a switching detector (or switching type).

3. *Full Data Dependent* (FDD) saboteur. For this model, the probabilities for a line are dependent on the data configuration on the entire bus. This represents the most accurate model, as the timing and value characteristics for a wire are affected by crosstalk (which is data dependent). Although this model is the most accurate, it has very poor scalability: for an n-bit bus, 2^{2n} probabilities for a single line are derived (the crosstalk noise manifests when the bus switches, therefore).

4. *Partial Data Dependent* (PDD) saboteur. This model represents a simplification of the previous one. The probabilities for a line are dependent on the data configuration on vicinity (1-wire vicinity or 2-wire vicinity). The 1-wire vicinity model is based on the fact that the capacitance effect (which is dominant) manifests only on the neighbor line.

Fig. 1 presents the architectures for four types of saboteurs. All saboteurs consist of a random number generator (which is used to compute the probability of an error). The SAP incorporates a switch detection module, while the PDD and FDD monitor the data on the lines.

IV. The Fault Injection Campaigns

We performed several simulation campaigns, each of them consisting of 1000 runs and data transmitted was chosen randomly for each run. The simulations have been carried out using Modelsim 10.3 commercial HDL simulator on desktop computer with Intel Core 2 Duo at 2.4 GHz and 2 GB of main memory, with Windows XP OS.

The circuit under test has been the open-source Wishbone bus, designed in Verilog HDL and available on the OpenCores site [10]. The system was simulated in the particular case of 2 master units and 5 slave units, with 32-bit data and address buses. We have simulated conventional read and write cycles. The sabotaged signals have been grouped into the following:

- Data write signals (the 32-bit unidirectional data bus from master to slave)
- Data read signals (the 32-bit unidirectional data bus from slave to master)
- Address signals – a distinction between the first 4 address bits (the ones used to select the slave) and the rest of the address bits (which are used to address within the slave)
- Master control and handshaking signals (*we, cyc, stb* and *sel*)
- Slave handshaking signals (*ack, rty, err*)

The analyzed system is depicted in Fig. 2.

The simulation campaigns and simulation times are presented in Table I. Regarding the simulation times, a simulation set consisting of 1000 executions requires less than 2 s. The gold circuit simulation requires about 1 s.

Regarding the reliability analysis of the Wishbone bus, the following conclusions can be drawn:

1. Faults affecting the most significant signals of the address line have a dramatic effect on the overall signal reliability, as these signals are used for slave selection. Therefore, an error on these signals will result in selecting a wrong slave.
2. Faults affecting master to slave control and handshaking signals (*cyc, stb* and *we*) have the following effects: wrong type of transaction (read instead of write or vice-versa), no transaction is performed (because the bus arbiter cannot grant the bus to the master which had asserted the *cyc* signal or the slave to take into consideration the request from a master), prematurely terminated transactions (due to errors on an ongoing transaction on *cyc* and *stb* signals – these signals are activated throughout an entire transaction);
3. Faults affecting the slave to master handshaking have the following effects: the bus may enter into a stand-still, as the master does not de-asserts the *cyc* signals because he has not received any *ack, rty* or *err*; a transaction may be terminated before, as the master receives a wrong ack, err, or *rty* – in case an error affects *ack*, the master may read the wrong data; longer transaction when errors appear on the *rty* signal (usually a master restarts the transaction for a *rty*).
4. Faults that affect *sel* lines and data signals affect only the data transmitted on the bus. They do not affect the transaction timing or flow.

Thus, regarding the reliability of the bus, the most critical signals are the most significant bits in the address line and the control and handshaking signals.

V. Conclusions

This paper proposes SFI based reliability methods for interconnects affected by probabilistic faults. The SFI techniques used are based on saboteurs. We have developed four types of saboteurs: the simplistic probabilistic type, the switching-aware probabilistic saboteur, the full data-dependent saboteur and the partial data-dependent saboteur. The full data dependent and the partial data-dependent saboteurs are the most accurate ones, as the crosstalk noise which affects the interconnects is data dependent. We have performed several simulation campaigns analyzing the effects of probabilistic faults affecting interconnects on a Wishbone bus system, consisting of 2 masters and 5 slaves. The simulations have indicated the most critical signals in the overall reliability.

Acknowledgment

This work has been supported by the project "*i-RISC: Innovative Reliable Chip Designs from Low-Powered Unreliable Components*" – FP7 Grant Number 309129.

References

[1] N. S. Nagaraj, "Interconnect process variations: theory and practice", Proceedings of the 19th International Conference on VLSI Design (VLSID), 2006

[2] H. R. Zarandi, S. G. Miremadi, A. Ejlali, "Dependability Analysis using a Fault Injection Tool Based on Synthesizability of HDL Models", Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), 2003

[3] 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

[4] 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

[5] D. Boning, S. Nassif, "Models of process variations in device and interconnect", Design of High-Performance Microprocessor Circuits, chapter 06, pp. 98-116

[6] S. Hasan, A. K. Palit, W. Anheier, "Fault Diagnosis of Crosstalk Induced Glitches and Delay Faults", 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010

[7] M. Favalli, C. Metra, "TMR Voting in the Presence of Crosstalk Faults at the Voter Inputs", IEEE Transactions on Reliability, vol. 53, no. 3, september 2004

[8] A. Sanyal, A. Pan, S. Kundu, "A study on impact of loading effect on capacitive crosstalk noise" Proc. Int. Symp. On Quality Electronic Design (ISQED), 2009

[9] E. Jenn, J. Arlat, M. Rimen, J. Ohlsson, J. Karlsson, "Fault Injection into VHDL Models: The MEFISTO Tool", Proceedings 24th Annual International Symposium on Fault Tolerant Computing Systems (FTCS-24), 1994, pp 66-75

[10] OpenCores website: http://www.opencores.org

[11] K. Agarwal, D. Sylvester, D. Blaauw, "Modeling and analysis of crosstalk noise in coupled RLC interconnects", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 5, 2006

TABLE I. Simulation Results For All Campaigns

Fault model type	Victim signal	Probability of failure	Runtime [ms]
SSP during WRITE cycle	sel	3%	1828
	sel and data	3%	1765
	adr[31:28]	3%	1812
	adr[31:28]	5%	1750
	adr[31:28]	10%	1750
	cyc, stb, we, sel	3%	1750
SSP during READ cycle	ack, err, rty	3%	1703
SAP during WRITE cycle	adr[31:28]	5% for 0->1 3% for 1->0	1766
	adr[31:28]	10% for 0->1 5% for 1->0	1766
	cyc, stb, we, sel	5% for 0->1 3% for 1->0	1750
	cyc, stb, we, sel	10% for 0->1 5% for 1->0	1781
	data	5% for 0->1 3% for 1->0	1766
	data	10% for 0->1 5% for 1->0	1782
SAP during READ cycle	ack, err, rty	5% for 0->1 3% for 1->0	1703
	ack, err, rty	10% for 0->1 5% for 1->0	1703
PDD during WRITE cycle (1-wire vicinity)	adr[31:28]	[3% ÷ 20%], depending on the transition pattern	1797
	adr[27:0]		1797
	slave select signals		1766
	cyc, stb, we, sel		1766
	ack, err, rty		1765
	data		1782
PDD during READ cycle (1-wire vicinity)	ack, err, rty	[3% ÷ 20%], depending on the transition pattern	1782
	data		1906
Gold circuit – WRITE cycle	NO fault injection	0%	1078
Gold circuit – READ cycle	NO fault injection	0%	1046