Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Systems Architecture 50 (2004) 239–246
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

Efficient analysis of single event transients ☆

M. Sonza Reorda, M. Violante *

Dip. Automatica e Informatica, Politecnico di Torino, c.so Duca deglo Abruzzi 24, 10129 Torino, Italy

Abstract

The effects of charged particles striking VLSI circuits and producing single event transients (SETs) are becoming an issue for designers who exploit deep sub-micron technologies; efficient and accurate techniques for assessing their impact on VLSI designs are thus needed. This paper presents a new approach for generating the list of faults to be addressed during fault injection experiments tackling SET effects, which resorts to static timing analysis. Moreover, it proposes a simplified SET fault model, which is suitable for being adopted within a zero-delay fault simulation tool. Experimental results are reported on both standard benchmarks and real-life circuits assessing the effectiveness of the proposed techniques.

1. Introduction

The widespread adoption of very-deep sub-micron technologies is raising the concerns about the effects of soft errors, i.e., temporary circuit misbehaviors that are not the result of design errors or manufacturing defects. Soft errors are indeed the result of the interaction between circuits and the surrounding environment that can let even correctly designed and manufactured circuits to produce wrong results. To cope with these concerns, researchers are developing tools and techniques for accurately forecasting the impact of soft errors like single event upsets (SEUs), originated by the strike on the circuit of highly energized particles [1].

Fault injection [2] is usually exploited to accomplish such a task, since it allows perturbing the system with faults, mimicking the effects of soft errors during the normal circuit operation, e.g., when a workload is applied to the circuit. The goal of the fault injection is twofold: on the one hand, it is exploited to identify the portions of the system that are the most susceptible to soft errors; on the other hand, it allows gathering statistical evidence of the robustness, in terms of ability of detecting and correcting soft errors, of the analyzed system. Among the available techniques, the one known as simulation-based fault injection is very attractive since it is already usable when only a model of the considered system is available. Moreover, it is very flexible since any fault model can potentially be supported and faults can be injected in any component of the system. As a major drawback, the

☆ This work has been partially supported by the Italian Ministry for University through the Center for Multimedia Radio Communications (CERCOM).
* Corresponding author. Tel.: +39-0115647092; fax: +39-0115647099.
E-mail address: massimo.violante@polito.it (M. Violante).

1383-7621/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.sysarc.2003.08.008

approach may require high CPU times for simulating complex circuits.

This paper proposes a new approach to simulation-based fault injection that is suitable to efficiently analyze the effects induced on circuits by a new type of soft errors, known as single event transients (SET), which is rapidly becoming a major issue for designers of VLSI circuits [1]. The method exploits a simplified static timing analysis algorithm to identify among an initial fault list those faults that are likely to produce circuit misbehaviors. Moreover, our approach is able to identify, *without* resorting to simulation, which are the faults that will not produce any modification of the circuit outputs and that can thus be removed from the fault list since their effect is known a priori. By reducing the number of faults that should be simulated, our approach is thus able to significantly reduce the CPU time required to perform fault injection experiments.

The paper also presents a simplified fault model, which can be used to emulate SETs while resorting to a zero-delay simulator for fault effects analysis. The major benefit stemming from the adoption of our new fault model is the possibility to exploit traditional zero-delay fault simulation algorithms to assess effects of SETs instead of more time consuming time-accurate ones. As a result, the CPU time needed for performing fault injection can be reduced by orders of magnitude.

In order to assess the proposed approach, we gathered several experimental results on both well-known benchmarks and on real-life circuits. The recorded figures show that the fault list reduction algorithm is able to collapse the initial fault list by 90% on the average: the CPU time needed for analyzing the effects of a given fault list is thus proportionally reduced. When the fault list reduction algorithm is combined with the simplified SET fault model and a zero-delay fault simulation is exploited, we are able to further reduce CPU time by 4 orders of magnitude. The experiments we performed also focused on the analysis of the estimation accuracy the simplified SET fault model achieved. We observed that, despite our approach introduces some approximations, it always provides conservative results, in the sense that it overestimates the number of SETs modifying the system outputs.

The main novelties of this paper with respect to that we presented in [3] are a more detailed discussion of the considered fault type as well as new and extended experimental results, which show the applicability of the proposed techniques to real-life circuits.

The paper is organized as follows: Section 2 revises the previous works concerning fault injection techniques. Section 3 reports details about the fault model we considered, while Section 4 describes the proposed fault list reduction algorithm. Section 5 presents the simplified SET fault model we developed, while Section 6 reports and comments the experimental results we gathered. Finally, Section 7 draws some conclusions.

2. Previous works

In the last years, several approaches to simulation-based fault injection have been proposed. Earlier works, such as [4], proposed the exploitation of switch-level simulators for analyzing error propagation through a circuit. In order to match the constantly increasing complexity of the target circuits, a probabilistic and a deterministic approach have then been proposed. The authors of [5] exploited device-level simulation to estimate the probability that ion-device interactions produce erroneous signals capable of propagating to memory elements, moreover logic-simulation was exploited to analyze how wrong information propagates among the circuit. More recently, a further method has been proposed, based on the probabilistic description of error propagation in VLSI circuits, formulated and solved as a set of linear equations [6].

The alternative approach is based on deterministic simulation. Switch-level simulation tools, such as the one described in [7] can be adapted to the execution of fault injection experiments targeting soft errors. More recently, inspired by the widespread adoption of HDL simulators in VLSI design centers and the high level of efficiency of modern simulation algorithms, several authors proposed the use of HDL simulators to perform

fault injection campaigns, and many approaches have been presented (e.g., [8]) for speeding-up the process. The approach presented in [9] is based on probabilistic and deterministic models allowing the identification of the most sensible elements among the gates in a combinational circuit. As a result, fault injection is performed on a potentially small subset of the circuit gates, thus saving simulation time.

3. Single event transients

Today, the fault model that is normally used during fault injection experiments is the (single/multiple) bit-flip in the circuit storage elements, i.e., registers and embedded memories, which is used to model the effects of single event upsets. With the adoption of deep sub-micron technologies, a new fault type is becoming of interest: the *single event transient*.

A single event transient is originated when highly energized particles strike a sensible area within a combinational circuit. In deep sub-micron CMOS devices, the most sensible areas are depletion regions at transistor drains [10]. The particle strike produces there several hole–electron pairs that start to drift under the effect of the electric field. As a result, the injected charge tends to change the state of the struck node with a short voltage pulse. As the depletion region is reformed, the charge-drift process decays, and the expected voltage level at the struck node is restored.

Highly energized particles are particularly common in the space environment, while in ground-level applications they are typically originated by radioactive decay of the packaging of integrated circuits or by the interaction between cosmic neutrons and atoms in the atmosphere [10]. In deep sub-micron circuits the capacitance associated to circuit nodes is very small, therefore non-negligible disturbances can be originated even by small amounts of deposited charge, i.e., when energized particles strike the circuit. Considering a typical deposited charge of 3 pC and a node capacitance of 4 pF, we have that the largest possible voltage disturbance is 0.75 V [10]. In old 5 V CMOS technologies, the magnitude of the voltage swing associated to SETs is about 15% of the normal voltage swing of the node and thus its impact is quite limited, in terms of both duration and magnitude. Conversely, if the technology is scaled to a 3.3 V one, the disturbance becomes 22% of a normal swing and thus the transistor that must restore the correct value of the struck node will employ more time to suppress the charge-drift process. Given the considered figures of deposited charge and node capacitance, SET effects on a 1.8 V technology will be certainly critical [10]. In very deep sub-micron technologies SET effects may become a critical issue since the duration of the SET-induced voltage pulse may become comparable to the gate propagation delay and thus the voltage pulse may spread throughout the circuit, possibly reaching its outputs. Two consequences may be produced: the affected outputs control the clock or the asynchronous reset/preset signals of a number of flip-flops, or they are sampled by memory elements thus provoking effects similar to those of SEUs.

As measurements reported in [10] show, SET can be conveniently modeled at the gate level as erroneous transitions (either from 0 to 1 or from 1 to 0) on the output of combinational gates.

4. Fault list reduction algorithm

We assumed that the considered system is either a combinational circuit or the combinational portion of a sequential circuit. SET effects can spread through the fan-out cone of the affected gate, the *faulty gate*, if, and only if, the duration of the spurious transition is equal to or longer than the gate propagation delay and if the magnitude of the transition is compatible with the device logic levels. In the following, we will concentrate our attention only on those particles that when hitting the circuit produce SETs that satisfy the above conditions.

Let T_H be the time when the SET is originated by a particle strike, δ be the worst-case SET duration for the considered type of particles, T_S the time when the outputs of the circuit are sampled (determined by the system clock cycle) and Π is the set of the propagation delays associated to the

sensitized paths from the faulty gate to the circuit outputs, e.g., all those paths that, due to the input configuration on the circuit inputs, let a change on the output of the faulty gate to spread the circuit outputs.

Any SET is effect-less, i.e., its effects cannot reach the circuit outputs, if the following condition is met:

$$T_H + \delta + t < T_S \quad \forall t \in \Pi \tag{1}$$

If Eq. (1) holds, it means that as soon as the SET expires and the expected value is restored on the faulty gate, the correct value has enough time to reach the circuit outputs, and thus the expected output values are sampled. The values T_H and δ are known since they are used to characterize the SET in the initial, complete, fault list: T_H is usually randomly generated with a time resolution equal to that of smallest propagation delay of the considered technology, while δ is selected on the basis of the worst-case energy of the particles expected to strike the circuit. Furthermore, T_S is known a priori, and it is normally selected according to the system clock cycle.

In order to identify the SETs possibly affecting the circuit output values, we need to compute Π. To develop a simple but efficient method for this task, we made the assumption that every path stemming from the faulty gate is sensitized. As a result, we overestimate the number of paths through which the SET may spread: indeed, given an input vector, only a subset of all the possible paths stemming from the faulty gate is normally sensitized. Nevertheless, this assumption allows us to compute the set Π in polynomial time *without* resorting to simulation. The algorithm we exploit for computing Π is reported in Fig. 1, where τ_k is the propagation delay of gate k, and Π_i is the list of propagation delays associated to paths stemming from gate i. In this analysis the delay introduced by the interconnections is neglected, but it can be easily taken into account provided that the circuit layout is known.

After the computation of the values of propagation delays has been performed, the algorithm reported in Fig. 2 is used to reduce the fault list, where TFL is the initial fault list, RFL is the reduced fault list, f is a fault extracted from the initial fault list and i is the faulty gate.

```
∏i = {TS}   ∀i | i is a circuit output
l = max circuit level – 1
while( l > 0 )
{
  foreach( gate i at level l )
    foreach( gate k on the fan out of gate i )
      ∏i = ∏i ∪ (∏k-τk)
  l = l - 1
}
```

Fig. 1. Static timing analysis algorithm.

```
RFL = ∅
foreach( fault f ∈ TFL on gate i originated at T^f_H)
{
   if not(∏i + T^f_H + δ < TS )
      RFL = RFL ∪ f
}
```

Fig. 2. Fault list reduction algorithm.

At the end of the fault list reduction algorithm, simulation-based fault injection experiments are required to assess the effects of the faults in RFL. As far as combination circuits are considered, the fault injection experiments are intended for identifying the faults in RFL that actually propagate to the circuit outputs. Conversely, when sequential circuits are considered, fault injection is used also to identify which faults in RFL propagate up to the circuit memory elements. In the latter case, the simulation of several clock cycles may be required in order to understand if the fault is able to propagate from the circuit memory elements to its outputs.

5. Simplified SET fault model

No matter how the list of faults has been selected, simulations are still required for assessing fault effects and when very complex designs are considered, the CPU time for simulation execution may be prohibitive. The authors of [11] already proposed an approach that, by exploiting a mixed-level simulator, is able to significantly reduce simulation time. The approach is specifically

crafted to deal with sequential circuits, and thus it cannot be applied when combinational circuits are addressed. In this section, we propose a new and approximate SET fault model suitable to be adopted within zero-delay fault simulators. As a result, fault simulation tools exploiting fault-parallel simulation algorithms can be exploited for assessing the effects of SETs, and CPU-intensive timed simulation can be avoided.

Let T_V be the time when a given vector is applied to the circuit inputs, T_H be the time when a SET originates, δ be the SET duration, T_S the time when the outputs of the circuit are sampled. The proposed fault model, called *vector-bounded stuck-at*, consists in approximating the considered SET with a stuck-at fault on the output of the faulty gate which originates at T_V and lasts up to T_S. As a result the stuck-at fault only affects the faulty gate during the evaluation of one vector (i.e., the single clock cycle during which the SET we are modeling appears). Moreover, the stuck-at has no effects on the faulty gate before and after the evaluation of the vector during which the SET we are modeling appears. In other terms, we propose to set $T_H = T_V$ and that $\delta = T_S - T_V$. Hence, timed-simulation is no longer needed, since the vector-bounded stuck-at appears as soon as a vector is applied to the circuit input and its effects last for exactly one vector.

Given an initial fault list containing the SETs whose effects we intend to analyze, by adopting the vector-bounded stuck-at we may obtain results with un-acceptable accuracy. We indeed neglect that in the fault list many faults may exist that satisfy Eq. (1). Conversely, if only the faults in the reduced fault list coming from the algorithm of Fig. 2 are simulated, we are able to greatly increase the accuracy zero-delay fault simulators may provide when evaluating SETs. The faults in the reduced fault list are indeed those whose effects, provided that at least one sensitized path exists, are able to spread to the circuit outputs in time to be sampled. Moreover, being the fault duration equal to the circuit clock pulse width, its effects are not filtered out by combinational re-convergence. This assumption neglects that, due to the presence of re-convergent fan-out stems from the faulty gate or due to the configuration on the circuit inputs, the SET effects may be masked before reaching the circuit outputs. Therefore, the results coming from this approach overestimate the number of actual errors produced by SETs. Nevertheless, the estimation accuracy is much higher than that obtained through a straightforward approach where all the SETs are modeled as vector-bounded stuck-at faults, and at a much lower CPU time cost than that timed fault simulation requires.

6. Experimental results

In this section we report experimental results we gathered by exploiting the techniques described in the previous sections. The effectiveness of the fault list reduction algorithm is evaluated in Section 6.1, where results gathered on large benchmarks and real-life circuits are reported. Moreover, Section 6.2 reports the figures we measured while evaluating the accuracy and effectiveness of our vector-bound stuck-at fault model.

All the experiments have been performed on a Sun UltraSparc 250 running at 400 MHz and equipped with 2 GB of RAM.

6.1. Analysis of the fault list reduction algorithm

The purpose of these experiments is to assess the effectiveness of the approach described in Section 4 in terms of compaction efficiency, i.e., the capability of reducing the cardinality of the list of SETs we are interested in analyzing, when large circuits and complex workloads are considered.

For this purpose we considered a workload composed of 1,000 randomly generated input stimuli and we analyzed several circuits coming from different sources: the five largest circuits belonging to the ISCAS'89 benchmark set, a floating-point coprocessor (FPU) able to perform addition, subtraction and comparison operations in compliance with the IEEE 754 standard and the integer unit (IU) of a Sparc v8 compatible processor core. Being the considered benchmarks sequential circuits, we first extracted their combinational part. Then, we generated an initial fault list composed of randomly selected faults and we applied our fault list reduction algorithm. In

generating the initial fault list we set the number of faults equal to $N_{vect} \cdot N_{gate}$, where N_{vect} is the number of input stimuli in the workload and N_{gate} is the number of gates in the circuit.

During these experiments, we recorded the time for fault list compaction as well as the size of the initial fault list and the compaction ratio (the number of faults removed from the initial fault list over that of the initial fault list). The obtained figures are reported in Table 1. As one can observe from these figures, the approach we propose is able to significantly reduce the number of faults to be simulated: the average compaction ratio is indeed about 90%. As a result, significant savings in terms of time needed for performing injection experiments can be achieved.

As far as the accuracy of the attained results in terms of fault effect classification is considered, the fault injection experiments we performed confirmed to observation we already reported in [3]: by simulating the reduced fault list we recorded numbers of SETs leading the circuits to wrong results which are exactly equal to those measured while simulating to the initial fault list.

6.2. Analysis of the vector-bounded fault model

The aim of the following experiments is two-fold: to measure the speed-up that we attain by simulating with a zero-delay fault simulator instead of using a timed simulator, and to measure the loss of accuracy stemming from the simplified fault model described in Section 5.

In the left half of Table 2 we reported the fault injection results obtained by simulating the initial fault list and the reduced one with an in-house developed zero-delay gate-level fault simulator supporting the vector-bounded stuck-at. Results are reported in terms of *failure rate*, i.e., the ratio between the number of faults producing wrong

Table 1
Results on large circuits

Circ.	Initial fault list [#]	Compaction ratio [%]	Fault list reduction time [s]
s13207c	7,951,000	93.47	86.76
s15850c	9,772,000	93.75	68.77
s35932c	16,065,000	91.61	127.62
s38584c	19,253,000	95.61	142.28
s38417c	22,179,000	94.01	166.73
FPU	6,333,000	83.75	41.89
IU	13,050,000	82.93	130.51

Table 2
Zero-delay vs. timed simulation

Circ.	Fault effect classification			Simulation execution time	
	Zero-delay simulation		Timed simulation [%]	Zero-delay simulation [s]	Timed simulation [s]
	Initial fault list [%]	Reduced fault list [%]			
c17	82.24	24.52	24.52	0.1	669.7
c432	30.52	8.40	3.28	0.2	2261.6
c499	31.16	1.78	1.78	0.1	2865.7
c880	52.84	4.52	2.18	0.2	3980.3
c1355	40.02	4.34	1.56	0.3	3496.9
c1908	43.62	11.12	1.12	0.4	3606.4
c2670	43.18	7.62	1.14	0.5	15,037.5
c3540	27.28	7.96	1.06	0.6	3177.6
c5315	40.24	5.22	1.06	0.9	6532.9
c6288	89.82	38.68	11.82	1.3	3360.3
c7552	41.62	8.48	1.44	1.2	8254.6

answers over the total number of fault in the fault list. For comparison sake, we reported the results of timed simulations [12] performed on the initial fault list, too. For the sake of this experiments we considered the circuits in the ISCAS'85 benchmark set.

As anticipated in Section 5, the results coming from fault simulating the initial fault list with a zero-delay fault simulator and the vector-bounded stuck-at are quite different than those coming from timed simulation. However, error rate figures coming from adopting our new fault model and exploiting a zero-delay fault simulator are close to, and in some cases coincident to, those timed simulations produce. The vector-bound stuck-at fault model is indeed able to provide an estimation of the exact circuit failure rate: the reader should note that this estimation is always *conservative* (i.e., the measured failure rate is always greater or equal to the actual failure rate), so that designers can always rely on our method for identifying a superset of the SETs producing effects on the circuit outputs. If a more detailed analysis is required, exact timed simulation can be performed on the faults marked as possible failures by the zero-delay fault simulation; in this way significant savings in the overall required CPU time can still be obtained.

The right half of Table 2 compares the CPU time required for timed simulation with that required by zero-delay one. From this table, the advantages stemming from the adoption of the vector-bounded stuck-at are evident: if accuracy is not a major concern, SET effects can be studied with a 4-magnitude speed-up over timed simulations.

7. Conclusions

This paper first presented an approach that exploits simple static timing analysis of combinational circuits for effectively reducing the list of faults to be considered during SET effects analysis. As experimental results showed, the proposed method is able to reduce the fault list size by 90% on the average, without reducing the accuracy of the obtained results.

A new and simplified fault model was also proposed, whose aim is to allow the analysis of SET effects by exploiting zero-delay fault simulation instead of time-accurate simulation. Experimental results are reported, assessing the effectiveness of the approach in reducing the CPU time for SET effects analysis. As far as the accuracy of the analysis is concerned, the experiments showed that the obtained results are affected by a relatively small estimation error: the proposed method always identifies a superset of the SETs producing some effects on the circuit outputs, and thus it is useful to preliminary (and very quickly) identify the set of faults to be later analyzed with accurate (and time consuming) timed simulations.

References

[1] L. Anghel, M. Nicolaidis, Cost reduction of a temporary faults detecting technique, in: DATE'2000: ACM/IEEE Design, Automation and Test in Europe Conference, pp. 591–598.
[2] M.-C. Hsueh, T.K Tsai, R.K. Iyer, Fault injection techniques and tools, IEEE Computer 30 (4) (1997) 75–82.
[3] M. Sonza Reorda, M. Violante, Fault list compaction through static timing analysis for efficient fault injection experiments, in: IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2002, pp. 263–271.
[4] B.L. Bhuva, J.J. Paulos, R.S. Gyurcsik, S.E. Kerns, Switch-level simulation of total dose effects on CMOS VLSI circuits, IEEE Transactions on Nuclear Science 8 (9) (1989) 933–938.
[5] N. Kaul, B.L. Bhuva, S.E. Kerns, Simulation of SEU transients in CMOS IC, IEEE Transactions on Nuclear Science 38 (6) (1991) 1514–1520.
[6] M.P. Baze, S. Buchner, W.G. Bartholet, T.A. Dao, An SEU analysis approach for error propagation in digital VLSI CMOS ASICs, IEEE Transactions on Nuclear Science 42 (6) (1995) 1863–1869.
[7] P. Dahlgren, P. Liden, A switch-level algorithm for simulation of transients in combination logic, in: Proc. Fault Tolerant Computing, FTCS-25, 1995, pp. 207–216.
[8] E. Jenn, J. Arlat, M. Rimen, J. Ohlsson, J. Karlsson, Fault injection into VHDL models: the MEFISTO tool, in: Proc. Fault Tolerant Computing, FTCS-24, 1994, pp. 66–75.
[9] L.W. Massengill, A.E. Baranski, D.O. Van Nort, J. Meng, B.L. Bhuva, Analysis of single-event effects in combinational logic-simulation of the AM2901 Bitslice processor, IEEE Transactions on Nuclear Science 47 (6) (2000) 2609–2615.
[10] K.J. Hass, J.W. Gambles, Single event transients in deep submicron CMOS, in: IEEE 42nd Midwest Symposium on Circuits and Systems, 1999, pp. 122–125.
[11] H. Cha, E.M. Rudnick, J. Patel, R.K. Iyer, G.S. Choi, A gate-level simulation environment for alpha-particle-induced transient faults, IEEE Transaction on Computers 45 (11) (1996) 1248–1256.

[12] B. Parrotta, M. Rebaudengo, M. Sonza Reorda, M. Violante, New techniques for accelerating fault injection in VHDL descriptions, in: IEEE International On-Line Testing Workshop, July 2000, pp. 61–66.

Matteo Sonza Reorda took his Master and PhD degrees in Electronics (1986) and Computer Science (1990) from Politecnico di Torino, Italy. Since 1990 he is with the Department of Computer Science and Automation of the same Institution, where he is now a Full Professor. He serves in the PC of several international events, and has been the General and Program Chair of the IEEE International On-line Test Symposium. His research interests include test of Integrated Circuits and design techniques for Fault Tolerant systems.

Massimo Violante received the Ms and PhD degrees from the Department of Computer Science and Automation of Politecnico di Torino, Italy, in 1996 and 2001 respectively, and he is now an Assistant Professor with the same institution. His main research interests are testing of digital systems, design and evaluation of fault tolerant systems.