CARROT – A Tool for Fast and Accurate Soft Error Rate Estimation

Dimitrios Bountas and Georgios I. Stamoulis

Department of Computer and Communications Engineering
University of Thessaly
37 Glavani St.
Volos 38221, Greece
{dibountas, georges}@uth.gr

Abstract. We present a soft error rate (SER) analysis methodology within a simulation and design environment that covers a broad spectrum of design problems and parameters. Our approach includes modeling of the particle hit at the transistor level, fast Monte-Carlo type simulation to obtain the latching probability of a particle hit on all nodes of the circuit, embedded timing analysis to obtain the latching window, and fine-grained accounting of the electrical masking effects to account for both the effects of scaling and of pulse duration versus the period of the system clock to get an estimate of the maximum SER of the circuit. This approach has been implemented in CARROT and placed under a broad design environment to assess design tradeoffs with SER as a parameter.

Keywords: SER, combinational circuits, simulation.

1 Introduction

Particle hits first became an issue for memories but with the continuously shrinking feature sizes and shrinking supply voltages, these issues have become significant in the design of any modern integrated circuit. Soft errors induced in memory structures and latches have been extensively studied and there are empirical models covering their contribution to the overall SER of the circuit [1]. These models work well as the laws of physics along with the electrical behavior of the device suffice for the adequate description of the phenomenon.

The contribution of combinational logic blocks is intrinsically more complicated and not as straight-forward, since the current pulse induced by the particle hit and the localized electrical behavior of the device describe only the generation of a single event upset and not its latching into one of the sequential elements, which would transform it to a soft error. The resulting voltage pulse on the gate output can be eliminated by i) the electrical behavior of the gates it has to cross to reach a latch; the pulse may not be wide enough and it will be filtered out by the inertial delay of the gate, ii) the logic behavior of the circuit; the pulse may be blocked at a gate if the output of the gate does not change logic state due to the pulse, which is a function of

S. Vassiliadis et al. (Eds.): SAMOS 2006, LNCS 4017, pp. 331 – 338, 2006.

the circuit inputs and the gate structure and iii) the timing of the signals, as the pulse may not overlap with the sampling range of the flip-flop it drives.

Several approaches with varying levels of detail have been proposed for the estimation of the SER due to combinational logic that can be categorized as follows:

a. direct simulation of the SEU within a circuit-level framework. Tools such as SEMM [2] use Monte-Carlo simulation for achieving high levels of accuracy but at a high price for simulation time.

b. mixed-mode simulation. The approach suggested by Dynamo [3] attempts to reduce the simulations required by a static approach initially, and processes the remaining ones through a mixed-mode simulator, where the current injection part is simulated at the circuit level, while the rest of the circuit at the timing level.

c. hybrid methods such as SERA[4] and SEUPER_FAST [5]. SERA uses a combination of mathematical and simulation methods to arrive at a SER estimate without the overhead of detailed circuit simulation. It simplifies gates between the charge injection point and the latch input nodes to equivalent inverters and performs and electrical simulation over the simplified circuit. SEUPER_FAST is a simulator at a very high level of abstraction that does not consider either circuit or logic simulation but models SER through a mathematical model.

d. logic-level simulation coupled with glitch and delay analysis. This category includes FAST [6], ASERTA [7], and ASSA [8], and tries to decouple the three factors (electrical, logic, and timing masking) that limit SEU latching. FAST compares the use of a standard event-driven simulator to assess SEU propagation and latching with a zero-delay fault simulator, while using a timing simulator to model the charge injection. ASERTA is using lookup tables for charge insertion, zero-delay simulation for logical masking, a ramp model for electrical masking, and pulse duration for timing masking. ASSA uses and extended timing window for timing masking, probability propagation for logic masking, and a noise rejection curve for electrical masking.

All of the aforementioned approaches have their drawbacks either in terms of accuracy or in terms of runtime. Furthermore, they tend to restrict the control that a designer has over the level of abstraction that is suitable for the design at hand, as well as the side effects of changes to enhance SER. The major source of the error introduced by approaches such as FAST, ASERTA, and ASSA stemmed from the decoupling of the three SEU latching prevention factors. Indeed, this permitted the use of zero-delay simulators for logic masking as well as lookup tables and simplified models for capturing electrical and timing effects.

We are proposing a new SER estimation approach that alleviates the restrictions that limit the accuracy while preserving the significant speedup of a zero-delay fault simulation approach. Our approach uses a zero-delay Monte-Carlo fault simulator that has been augmented by capabilities for handling timing and electrical masking at a fine grain level, within the main flow of fault simulation. This allows the appropriate weighting of the random effects, which is not possible in the flow suggested by ASERTA and ASSA. Our SER simulator, CARROT, also permits the use of charge injection, pulse propagation, and timing analysis methods of varying detail, so that the user can choose between simulation speed or accuracy.

The aim of this paper is to include SER estimation within a larger design framework and permit trading off SER for timing, power, area, and/or leakage.

The rest of the paper is organized as follows. Section 2 describes the soft-error analysis methodology that we are proposing, Section 3 presents results for all ISCAS85 and ISCAS89 benchmark circuits, and Section 4 contains the concluding remarks.

2 Soft Error Analysis

We assume a standard sequential digital circuit that can be described by an array of flip-flops controlled by one clock and combinational logic driven by flip-flop outputs and driving flip-flop inputs.

We also assume that the Soft Error Rate (SER) of a circuit is defined as the fraction of Single Event Upsets (SEUs) that are captured by the memory elements of the circuit, and that an SEU can occur with uniform probability over the clock period.

2.1 Electrical Modeling

A high-energy particle hit can interact with silicon atoms and generate a number of electron-hole pairs [9], which can flip the state of a logic node. This phenomenon is modeled by a current source with exponential decay as in [10].

$$I(t) = I_0(e^{-t/a} - e^{-t/b})$$

Different particle energy levels will produce a different number of electron-hole pairs which will be reflected into I_0. The circuit node that is affected can be either a gate output or an internal node of a gate. In both cases the particle hit appears as voltage pulse at the gate output. Since the particle energy level can be described as a random variable, the width of the output voltage pulse can also be described as a random variable with a probability density function (pdf) that can be approximated by Monte-Carlo electrical simulations of the charge injection circuits.

Simplifying assumptions can be made on the pdf (uniform or normal) and on the random nature of the pulse width (one can assume that the voltage pulse width is fixed for a specific gate to just one value).

Our approach can work equally well with a fixed value as well as with any pdf for the pulse width. In the latter case, the pdf is propagated through the circuit to the flip-flop inputs. During the propagation, the pdf is altered to account for electrical masking and is then used to calculate the latching probabilities of the SEUs it describes. Our approach requires that the pdf be discretized into a user-defined number of levels to enable its propagation through the circuit. The largest quantization value can be equal to the period of the clock, since for pulse widths larger than that, timing masking has no effect.

The propagation process is intertwined with the logic and timing masking process and will be described in detail in the next subsection.

The probability of a particle hit on a specific node is proportional to the area of the node as the flux of particles through the circuit is considered uniform. Any deviations from this assumption should be quantified by a flux distribution over the area of the circuit, and by the location of the node within the chip. Thus, for non-uniform flux distributions, placement data are required.

2.2 SER Propagation and Electrical/Logic/Timing Masking

This is the main part of this work, which addresses the issues raised by the interaction of a logic level simulator, like the fault simulator that we are using, and the masking phenomena. This is precisely the point where loss of accuracy can occur due to either modeling or systematic difficulties. Instead of decoupling the analysis for the three masking effects, and recombining their effects at the end of the process, we opted to analyze them in every step in order to maintain the required granularity for accurate estimates without resorting to simplifying assumptions.

As in FAST and ASERTA, CARROT also uses a zero-delay fault simulator to estimate the effect of logic masking on the probability of latching the incorrect value due to an SEU. However, unlike the aforementioned approaches, the effects of logic and timing masking are engrained into the zero-delay logic simulation in order to eliminate the error resulting from:

i) an SEU latched in more than one flip-flop
ii) different arrival times and latching probabilities in different flip-flops
iii) differential pulse propagation behavior along different paths from the upset gate to the latch inputs.

After the pdf for the voltage pulse width at the output of the gate being affected by a SEU is calculated and discretized a circuit would be like the one Figure 1, where the pulse widths fall into N levels. The probability of an SEU on the node has been calculated as proportional to the upset node area.

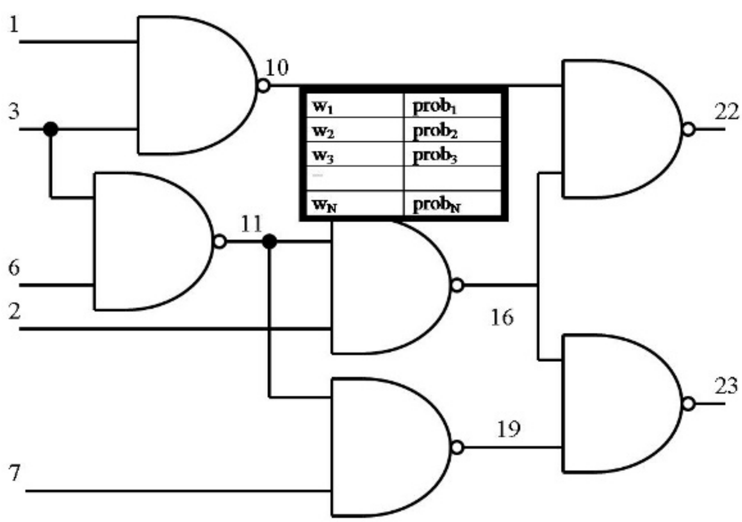

Fig. 1. Modeling of the pdf

We model the SEU as a stuck-at-fault and proceed through a Monte-Carlo fault simulation of the circuit. The number of input vectors is user defined, and the input probability for every input to the combinational block is determined by an RTL simulation in order to capture the effects of the input vectors on SER.

At the end of the Monte-Carlo fault simulation we preserve to flip-flop input vectors for further analysis. It must be noted here that we can now calculate the effect of logic masking on the flip-flop inputs. However, this is result requires further processing as the electric and timing masking effects are not accounted for yet.

In order to model electrical masking we chose the modeling of ASERTA, although the modeling in ASSA or any other could be used without impeding the subsequent analysis. Thus, we estimate the output pulse duration according to the following:

$$\begin{array}{rcl} 0 & \text{if} & w_i<d \\ 2(w_i\text{-}d) & \text{if} & d<w_i<2d \\ w_i & \text{if} & w_i>2d \end{array}$$

where w_i is the input pulse duration and d is the inertial delay of the gate. It must be noted here that we apply this function to all the entries in the table containing the discretized pdf of the pulse width.

For the timing masking we add a basic timing analysis capability by capturing the minimum and maximum pulse arrival times at each node, as shown in Figure 3. At the charge injection node both are 0. The timing at the output of each gate is propagated as follows:

$$\min_o = \min(\min(\text{all inputs to the gate})) + d$$
$$\max_o = \max(\max(\text{all inputs to the gate})) + d$$

where $\min_o$ and $\max_o$ are the min and max for the output node of the gate. For this purpose only the inputs of each gate activated by the upset node are considered.

A significant point is that we do not propagate pulse width information nor min/max timing information to outputs of gates that are not activated by the fault, i.e. their faulty and fault free outputs are identical. This helps eliminate false paths in both electrical and timing analysis.

A second point that needs attention is the handling of the pdf in reconvergent fanout nodes. In such a case we update the pdf as follows:

$$w_i = \max(w_{i,1}, w_{i,2})$$

where w_i is the updated pulse width, and $w_{i,1}$, $w_{i,2}$ the pulse widths being merged.

The above analysis is repeatedly applied to the fanouts of gates being activated until we reach the flip-flop inputs.

The last step is the calculation of the latching probabilities that includes the timing masking effects. In order to achieve that we require the pulse width pdf, the minimum and maximum arrival times and the logic vectors at the end of the fault simulation at every flip-flop input.

For each pulse width entry in the pdf we calculate the probability that it will be latched in a flip-flop. For each experiment in the Monte-Carlo simulation we check which flip-flop inputs are at an erroneous state from logic masking. From these we select the ones whose equivalent pulse width entries are not zero, i.e. the pulse has not been eliminated due to electrical masking, and we merge their minimum/maximum windows as follows:

$$bit\_min = min(of\ all\ selected\ nodes)$$
$$bit\_max = max(of\ all\ selected\ nodes)$$

Then the timing masking probability p_j would be:

$$p_j = \begin{cases} 1 & \text{if } bit\_max - bit\_min + wi + setup + hold > T \\ (bit\_max - bit\_min + wi + setup + hold)/T & \text{if } bit\_max - bit\_min + wi + setup + hold < T \end{cases}$$

where T is the clock period, setup the setup time of the flip-flop, and hold the hold time of the flip-flop.

The total latching probability is the sum of the above probabilities for all Monte-Carlo experiments over the number of experiments. The overall SER due to an upset at a specific node is the sum of the product of the pulse width probabilities by their respective total latching probabilities. Consequently, the overall SER of the circuit is the sum of the products of the SER for each node by the SEU probability calculated by the particle flux.

3 Results

The methodology described in Section 2 has been implemented in C++ tool (CARROT) and run on most of the ISCAS85 and ISCAS89 benchmark circuits. The combinational examples were assumed to have flip-flops at both primary inputs and outputs. The pdf was discretized into 10 levels, while 10000 vectors were used for the Monte-Carlo simulations, which is more than adequate to ensure high-confidence estimation of the probabilistic quantities required for the SER analysis (confidence level of more than 95%). The results are shown in Table 1. The reported execution times are on a Pentium4™ with 512MB of RAM. It should be noted that memory usage did not exceed 200MB.

4 Conclusion

This paper presents a new approach to SER estimation in combinational circuits. The proposed approach allows the accurate assessment of the effects of electrical, logic, and timing masking while still permitting a fast logic-level simulation for their evaluation. Both the pulse width modeling and the glitch absorption algorithm can be user defined for further flexibility and accuracy.

The method described above has been implemented in CARROT, which has become part of a broader simulation environment to assess the tradeoffs between SER and other design considerations such as timing, power, and leakage. Further research will be towards assessing circuit design solutions for SER enhancement.

Table 1. CARROT execution time for ISCAS85/89 benchmark circuits

Circuit Name	Nodes	Inputs	Gates	DFFs	Exec. time
S27	17	4	13	3	<1sec
S208_1	125	10	115	8	<1sec
S208	149	10	139	8	<1sec
S298	169	3	166	14	1sec
S386	284	7	277	6	2sec
S382	196	3	193	21	2sec
S344	240	9	231	15	2sec
S349	224	9	215	15	2sec
S400	203	3	200	21	2sec
S444	211	3	208	21	2sec
S526	280	3	277	21	4sec
S526N	280	3	277	21	4sec
S420	252	19	233	16	1sec
S510	293	19	274	6	3sec
S420_1	313	18	295	16	2sec
S832	457	28	429	5	7sec
S820	443	18	425	5	7sec
S641	517	35	482	19	8sec
S713	539	35	504	19	9sec
S953	496	16	480	29	11sec
S838_1	641	34	607	32	8sec
S838	641	34	607	32	8sec
S1238	768	14	754	18	17sec
S1196	762	14	748	18	16sec
S1494	1213	8	1205	6	39sec
S1488	1211	8	1203	6	33sec
S1423	1008	17	991	74	51sec
S5378	3053	35	3018	179	5.9min
S9234	7002	19	6983	228	35.3min
S9234_1	7019	36	6983	211	34.5min
S13207	9608	31	9577	669	1.1h
S13207_1	9609	32	9577	638	1.1h
S15850	12115	14	12101	597	1.7h
S15850_1	12178	77	12101	534	1.6h
S35932	21278	35	21243	1728	7.8h

Acknowledgements. The authors would like to acknowledge the help of Nestor Evmorfopoulos, Dimitrios Karampatzakis, Maria Kozyri, and Antonios Dadaliaris in support of this work.

References

1. International technology roadmap for semiconductors, http://public.itrs.net/, 2002.
2. Murley, P.C., and Srinivasan, G.R.: Soft-error Monte Carlo modeling program, SEMM, IBM Journal of Research and Development, vol. 40, no. 1 (1996) 109–118.
3. Yang F.L., and Saleh R.A.: Simulation and analysis of transient faults in digital circuits, IEEE Journal of Solid-State Circuits, vol. 27, no. 3 (1992) 258–264.
4. Zhang, M., and Shanbhag, N.R.: A soft error rate analysis (SERA) methodology," in Proceedings of International Conference on Computer Aided Design (2004) 111–118.
5. Baze, M.P., Buchner, S.P., Bartholet, W.G., and Dao, T.A.: An SEU analysis approach for error propagation in digital VLSI CMOS ASICs, IEEE Transactions on Nuclear Science, vol. 42, no. 6 (1995) 1863–1869.
6. Cha, H., Rudnick, E.M., Patel, J.H., Iyer, R.K., and Choi, G.S.: A gate-level simulation environment for alpha-particle-induced transient faults, IEEE Transactions on Computers, vol. 45, no. 11 (1996) 1248–1256.
7. Dhillon , Y.S., Diril, A.U., Chatterjee, A.: Soft-error tolerance analysis and optimization of nanometer circuits, in Proceedings of Design, Automation, and Test in Europe (2005) 288–293.
8. Zhao, C., Bai, X., Dey, S.:A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits, in Proceedings of Design Automation Conference (2004) 894-899.
9. Lantz, L.: Soft errors induced by alpha particles, in IEEE Transaction on Reliability, vol. 45, no. 2 (1996) 174-179.
10. Messenger, G.C: Collection of charge on junction nodes from ion tracks, IEEE Trans. Nucl. Sci., vol. NS-29, no. 6 (1982) 2024-2031.