Techniques for transient fault sensitivity analysis and reduction in VLSI circuits

Atul Maheshwari, Israel Koren and Wayne Burleson

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003, USA

Abstract

Transient faults in VLSI circuits could lead to disastrous consequences. With technology scaling, circuits are becoming increasingly vulnerable to transient faults. This papers presents an accurate and efficient method to estimate fault-sensitivity of VLSI circuits. Using a binary counter and an RC5 encryption implementation as examples, this paper shows that by performing a limited amount of random simulations, fault sensitivity can be estimated accurately at a reasonably low computational cost. This method is then used to show that the combination of two circuit level techniques can make circuits more fault-tolerant than using these techniques individually.

1 Introduction

Reliable operation of VLSI circuits is necessary to avoid catastrophic consequences especially for systems operating under adverse environment conditions. Information in electronic circuits is stored and communicated as a collection of electric charges. Any event which upsets the stored or communicated charge can cause errors in the circuit output. These errors are called transient faults, soft errors (SE) or single event upsets (SEU). The event causing the upset can be an energetic nuclear particle or an electrical source. The nuclear particles which create these upsetting events are either cosmic rays which bombard the earth constantly from space or radioactive atoms which exist in trace amounts in all materials due to atomic decay. Atmospheric nuclear particles include *alpha-particles* [1], protons [2] and neutrons [3]. Electrical sources are power supply noise, electromagnetic interference (EMI) or radiation from lightning [4].

Memories are considered most vulnerable to transients due to their spatial density and the amount of information they store. Recently, it has been demonstrated that it is important to consider memory arrays and core logic when estimating microprocessor soft error rate [5]. Soft errors and single event upsets are considered challenges for high performance and low-power microprocessor design [6].

While permanent faults are mostly related to manufacturing process, transient faults mostly occur due to environmental conditions. Transient faults have been known to account for 80% or more of field failures in digital systems [7, 8], thus making it imperative to estimate and reduce transient fault sensitivity of VLSI circuits. In this paper, a method to estimate the transient fault sensitivity of a circuit is presented. This is followed by an analysis of circuit level fault-tolerance schemes. The paper is organized as follows: Section 2 discusses the fault model used in this study. The fault estimation method is presented in Section 3 and is evaluated using a 4-bit binary counter and a hardware implementation of the RC5 encryption algorithm. Section 4 analyzes the fault tolerance improvement techniques for the two example circuits. Finally, conclusions are presented in Section 5.

2 Transient fault model

Transients can be represented at the device level by a current or a voltage source. These models accurately represent the electrical impact of the transient. Device-level models of cosmic-particle induced transients have been developed [9, 10]. In [9], a SPICE circuit with a current source was used to represent the collected charges generated by $\alpha - particles$. An approximate analytic solution which models a current transient is proposed in [11]. The model includes parameters which represent the maximum current, the collection time constant of the junction, and the time constant for initially establishing the ion track.

At the logic-level, transients can be modeled as a momentary *bit-flip* of the propagating signal. Logic-level approaches are inherently faster than device-level approaches since they do not rely on the evaluation of circuit equations. However, these approaches may not be very accurate. A transient can propagate along multiple paths and cause multiple latch errors. The probability of a faulty pulse propagating to a latch and becoming a latch error is a function of device-level parameters. Moreover, the shape of the pulse may be changed in transit through different gates in the propagation path. It has been shown that a discrete logic-level fault model can result in a 50% error when used to estimate soft errors [12].

Several device-level fault models for transient faults have been proposed [11, 13]. For this study we use the model presented in [11], which models a transient resulting from injection of an active node charge. This model is preferred over others since it can be used to represent other transients by changing its parameters. This work, however, is focusing on $\alpha - particles$ for which the transient is modeled as a double exponential injection current given by:

$$I_{inj}(t) = I_0(e^{-t/\tau_1} - e^{-t/\tau_2}) \tag{1}$$

where I_0 is the maximum current, τ_1 is the collection time constant for a junction and τ_2 is the ion track establishment time constant. τ_1 is dependent on the doping concentration and hence on the process. τ_2 is relatively independent of the technology. I_0 depends on the process and the charge intensity. Figure 1 shows the phenomena of an $\alpha - particle$ hit on a PMOS transistor and the equivalent current injection model.

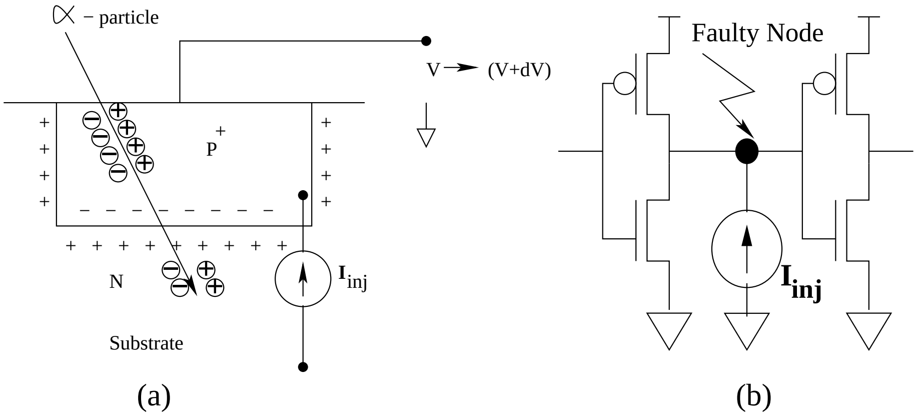


Figure 1. (a) α-particle hit on a PMOS transistor (b) The hit modeled as a current source

3 Fault-sensitivity analysis

Fault simulation and fault-sensitivity quantification are the key components in developing a fault-tolerance methodology. Fault simulation can be performed at several levels of design abstraction (i.e. RTL, gate-level, transistor-level etc.). The fault simulation abstraction level is determined by the fault model that is used.

The core of our fault simulation approach consists of HSPICE [14] based circuit-level simulations. Recent work [15] has shown that the fault sensitivity analysis for an alpha-particle induced transient can be performed at an early stage in the design cycle of VLSI circuits. Layout level designs are not mandated, as a particle hit creates free charge carriers only if hits occur in an active area [16].

A metric which quantifies fault-sensitivity, the Probability of Failure (POF), was proposed in [15]. The POF is given by

$$POF = \sum_{i=1}^{n} w_i \bar{E}_i \quad , \quad where \qquad w_i = \frac{A_i}{\sum_{i=1}^{n} A_i} \tag{2}$$

Here A_i is the area of the node i. $\bar{E}_i$ is given by

$$\bar{E}_i = \frac{1}{k} \sum_{i=1}^{k} E_i \quad , \quad where \qquad k = p \cdot q \cdot r \tag{3}$$

E_i, the outcome of a fault injection experiment is given by

$$E_i = \begin{cases} 1 & \text{if the injection into node } i \text{ results in a fault getting latched} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and
p is the number of input or state combinations,
q is the number of particle injection levels considered,
r is the number of time instances at which faults are injected,
n is the number of nodes in the circuit.

POF is thus a measure of the conditional probability of error given that a particle hits the circuit. Weighing the errors by node area allows us to account for the higher likelihood of larger nodes being hit by a particle. When comparing two different designs, POF fails to account for the higher likelihood of a larger circuit being hit by a particle. Hence, the product of POF and *size of circuit* is used here as a metric to estimate fault sensitivity independent of circuit implementation. This metric is henceforth referred as fault-sensitivity (FS).

$$FS = POF * area \tag{5}$$

In order to obtain an accurate measure of fault-sensitivity, all the conditions affecting the fault-sensitivity should be considered. These include:

- Inputs and/or state of the circuit.
- Size of charge generated by the *alpha-particle* strike.
- Circuit node at which the particle strikes.
- Time instance of the event.

The number of simulations to be performed to get an accurate estimate is given in equation 6. This number can be quite large, e.g., for a 4-bit binary counter $N \approx 100,000$.

$$N = p \cdot q \cdot r \cdot n \tag{6}$$

In order to reduce the run-time, simulations are performed for only randomly selected combinations. Figure 2 compares the POF values obtained by performing random simulations with the

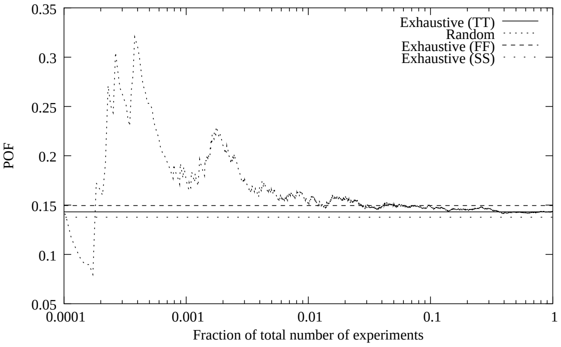


Figure 2. Convergence of random simulations for a 4-bit binary counter

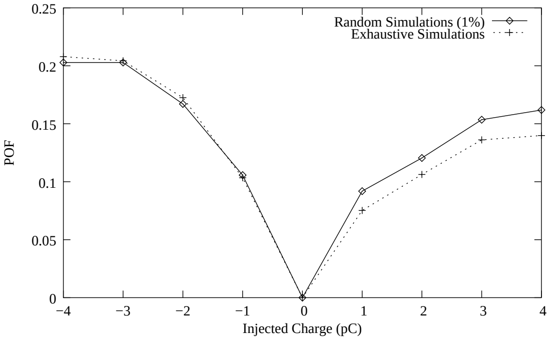


Figure 3. Comparison for various charge levels (4-bit binary counter)

POF value obtained by performing exhaustive simulations. It can be concluded that as the number of simulations is increased the POF of a 4-bit binary counter estimated using random simulations rapidly converges to the POF estimated using exhaustive simulations. Figure 2 also shows the POF estimates obtained, if the "Fast-Fast" or "Slow-Slow" transistor models were used instead of the "Typical-Typical" models. The error in the POF estimates due to process variation is more than the error due to performing limited number of random simulations(1%). Thus, by performing 1% of the total number of experiments a reasonably accurate estimate of POF can be obtained. Figure 3 shows that for different amounts of charge injected, the POF estimated using exhaustive simulation is close to the POF obtained by performing 1% of total experiments. Similar results are obtained when simulations are performed for different input vector or different time instance of transient. These results further validate the claim of using random simulations for estimating transient fault sensitivity.

The proposed fault-sensitivity estimation technique was also evaluated using a hardware implementation of the RC5 encryption algorithm [19]. The pseudo-code of the algorithm is shown below and the hardware implementation is shown in Figure 4. The encryption algorithm parameters are 16/16/4, i.e., 16 bit data, 16 bit key and 4 rounds of encryption.

```
A=A+S[0];
B=B+S[1];
for i=1 to r do
   A = ((A xor B) << B) + S[2*i];
   /* A XOR B shift left B times and add the key */
   B = ((B xor A) << A) + S[2*i];
   /* B XOR A shift left A times and add the key */

   /* A and B are the upper and lower half of the data bits */
   /* S is the Key array */
   /* r is the number of rounds */
```

Figure 5 compares the POF values obtained by performing random simulations with the POF values obtained by performing exhaustive simulations for this implementation of the RC5 encryption algorithm. The results are very similar to the ones obtained for the binary counter and consequently, by performing 1% of the total simulations, reasonable estimates of POF can be obtained. Figure 6 shows that for different amounts of charge injected, the POF estimated using exhaustive simulation is close to the POF obtained by performing 1% of total experiments.

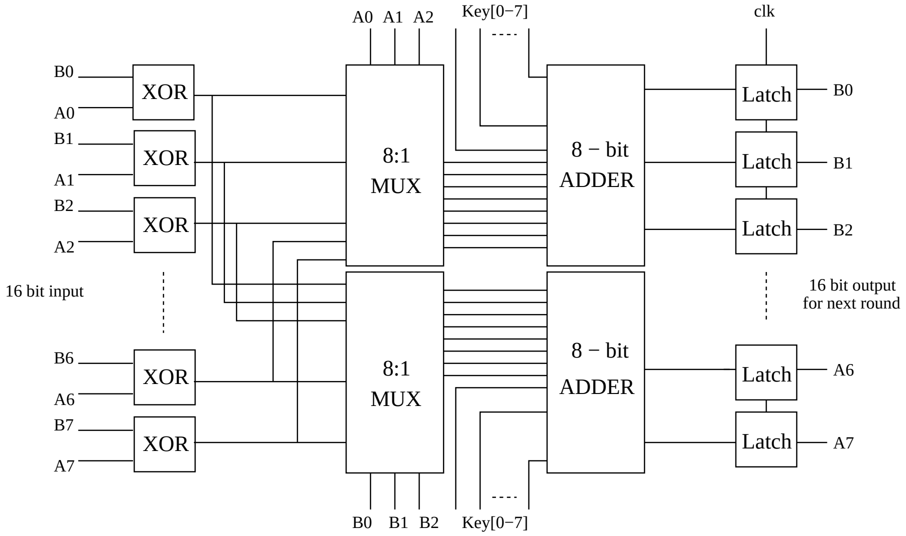
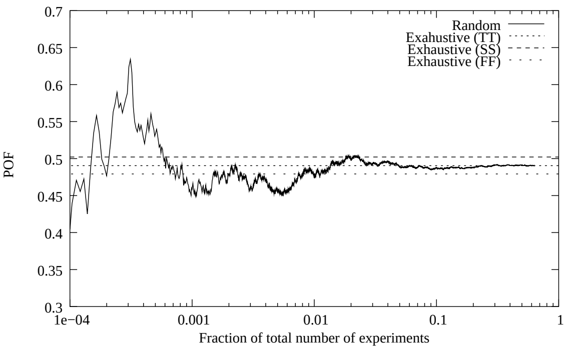


Figure 4. Hardware implementation of the RC5 Encryption Algorithm



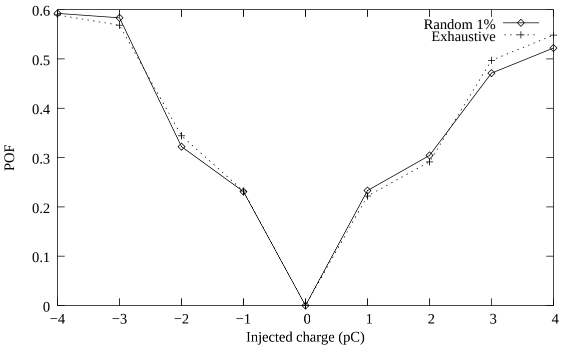


Figure 5. Convergence of random simulations (RC5 encryption algorithm)

Figure 6. Comparison for various charge levels (RC5 encryption algorithm)

4 Fault-tolerance enhancement techniques

Several circuit-level fault tolerance techniques have been proposed. These include, the use of a Transient Pulse Tolerant Latch (TPTL)[17] as shown in Figure 7 and sizing of transistors in the circuit [18]. By using a RC low-pass filter, TPTL tries to filter out the high frequency transients. Sizing of transistor on the other hand reduces the offset voltage caused by the transient (Figure 8).

4.1 Results for binary counter

As shown in Figure 9, using a combination of TPTL and transistor sizing technique provides a much higher fault-tolerance than using any of these techniques alone. Sizing the transistors of the circuit by a factor of 2.4 and inserting a filter with an RC product of 14000 results in a reduction of fault-sensitivity (FS) by a factor of 1.39. Fault-sensitivity improves at most by a factor of 1.27 by using the transistor sizing alone. If the transient pulse tolerant latch alone is used, the fault-sensitivity improves by a factor of 1.19 at most. Thus, the combination of the two techniques gives a much better result as compared to using the techniques individually.

The above mentioned fault tolerance techniques adversely affect the performance of the circuit.

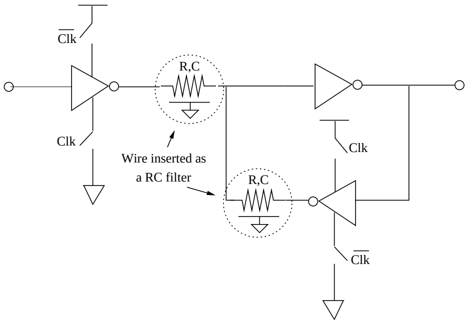


Figure 7. Transient Pulse Tolerant Latch (TPTL)

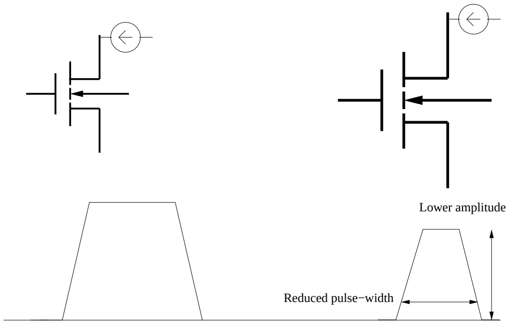


Figure 8. Impact of transistor sizing on the transient pulse

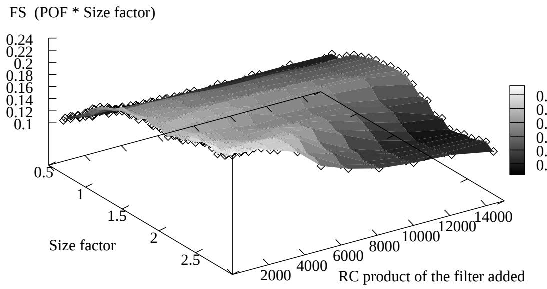


Figure 9. Fault Sensitivity as a function of two techniques

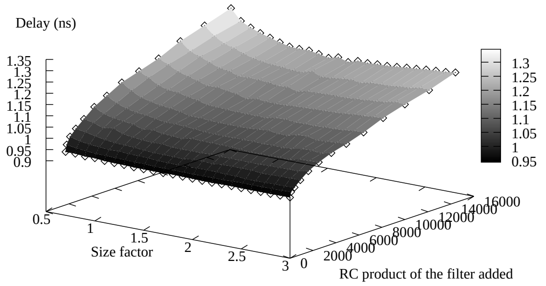


Figure 10. Delay as a function of two techniques

As shown in Figure 10 the delay of the circuit with these techniques increases. Objective functions like normalized product of delay and fault-sensitivity (FS) can be used to determine a design solution as a trade-off between delay and fault-sensitivity. Figure 11 is the plot of above mentioned objective function. As seen in this figure, the best design of the counter for this objective function is the one with transistors sized by a factor 0.7 and a RC filter with RC product of 900 inserted in the latch.

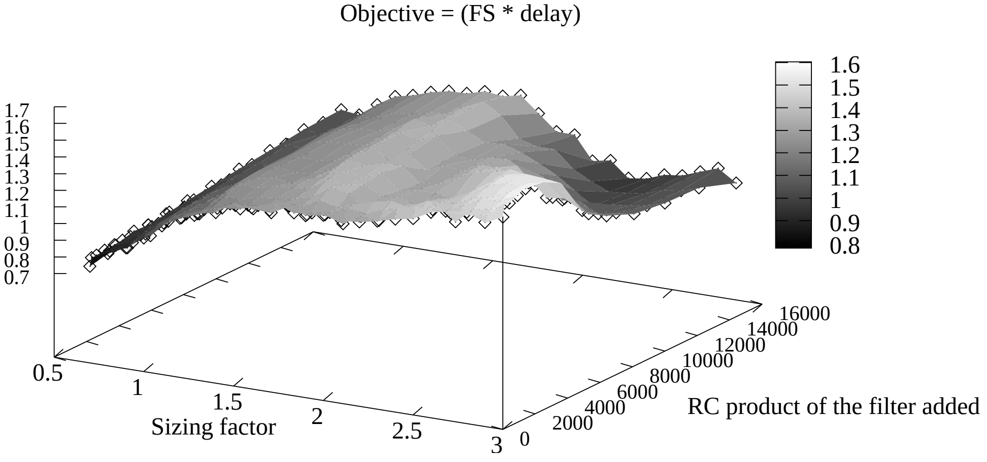
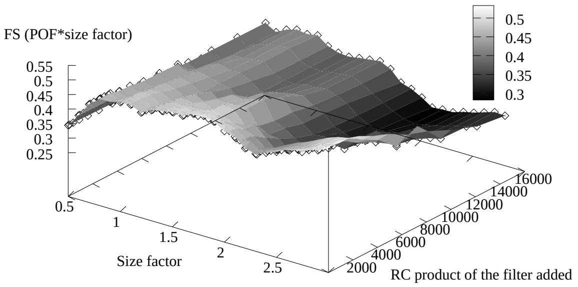


Figure 11. Delay and Fault-sensitivity product



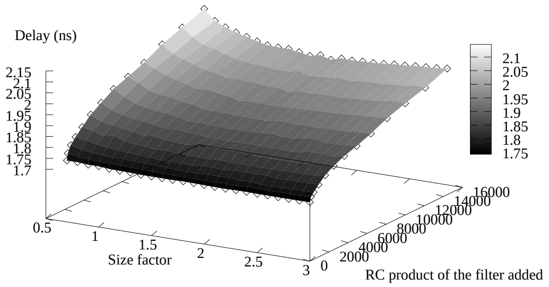


Figure 12. Fault Sensitivity as a function of two techniques

Figure 13. Delay as a function of two techniques

4.2 Results for the RC5 encryption algorithm

As shown in Figure 12, using a combination of TPTL and transistor sizing technique provides a much higher fault-tolerance than using any of these techniques alone. Sizing the transistors of the circuit by a factor of 2.5 and inserting a filter with an RC product of 14000 results in a reduction of fault-sensitivity (FS) by a factor of 1.64. Fault-sensitivity improves at most by a factor of 1.12 by using the transistor sizing alone. If the transient pulse tolerant latch alone is used, the fault-sensitivity improves by a factor of 1.16 at most. Thus, similar to the binary counter, the combination of the two techniques gives a much better result as compared to using the techniques individually.

Figure 13 shows the delay of the circuit incorporating these techniques and Figure 14 shows the objective function for various combinations of the transient fault-tolerant circuit techniques. As seen in this figure, the best design of the encryption circuit is the one with transistors sized by a factor 2.3 and a RC filter with RC product of 10000 inserted in the latch.

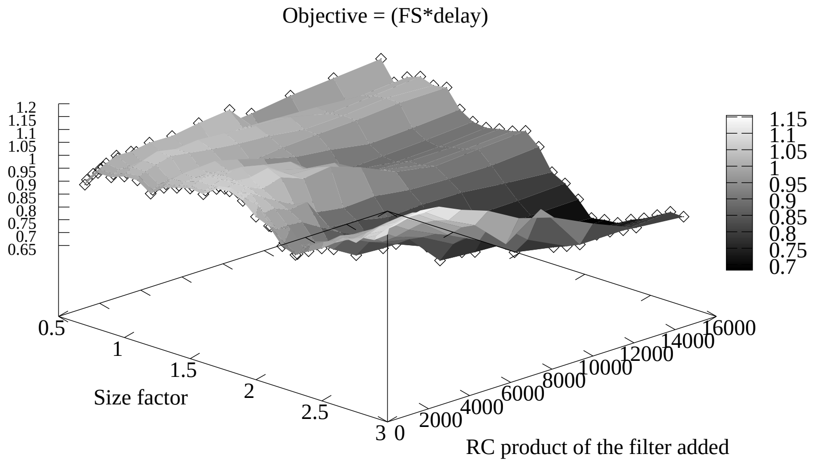


Figure 14. Delay and Fault-sensitivity product

5 Conclusion

In this paper a transient fault-sensitivity estimation method based on random simulations is presented. By performing 1% of total experiments, accurate estimates of fault-sensitivity can be obtained. It was also established that using a combination of circuit level fault-tolerance techniques (e.g., transient pulse tolerant latch and transistor sizing) provides a significant improvement over using these techniques individually. Using the delay and fault-sensitivity product as a design objective, the best combination of the circuit techniques was explored for a binary counter and an RC5 encryption circuit.

References

[1] T. C. May and M. H. Woods, “Alpha-particle-induced soft errors in dynamic memories”, IEEE Transactions Electronic Devices, Jan. 1979, vol 26, no 1, pp. 2-9.

[2] E. Normand et. al., “Single event upset and charge collection measurements using high energy protons and neutrons”, IEEE Transactions on Nuclear Science, Dec. 1998, vol 45, pp. 2904-2914.

[3] C. A. Gossett et. al., “Single event phenomena in atmospheric neutron environment”, IEEE Transactions on Nuclear Science, Dec. 1993, vol 40, pp. 1845-1852.

[4] J. F. Ziegler et. al., ”IBM experiments in soft fails in computer electronics (1978-1994)”, IBM Journal of Research and Development, 1996, Vol 40, pp. 3-17.

[5] N. Cohen et. al., “Soft error considerations for deep-submicron CMOS circuit applications”, Technical Digest of International Electronic Devices Meeting (IEDM), 1999, pp.315-318.

[6] V. De and S. Borkar, “Technology and design challenges for low power and high performance”, Proceedings of IEEE Symposium on Low Power Electronics and Design, 1999, pp. 163-168.

[7] H. Ball and F. Hardy, “Effects and detection of intermittent failures in digital systems,” Proceedings of FLCC, AFIPS conference, 1969, pp. 329-335.

[8] R. Iyer and D. Rosetti, “A statistical load dependency of CPU errors at SLAC,” Proceedings of FTCS, 1982.

[9] R. McPartland, “Circuit simulations of alpha-particle-induced soft errors in dynamic RAM’s”, IEEE Journal of Solid State Circuit, Feb. 1981, vol 16, pp. 31-34.

[10] R. Johnson, S. Diehl-Nagle and J. Hauser, “Simulation approach for modeling single event upsets on advanced CMOS SRAMS”, IEEE Transactions on Nuclear Science, Dec. 1985, vol 32, pp. 4122-4127.

[11] G. C. Messenger, “Collection of charge on junction nodes from ion tracks,” IEEE Transactions on Nuclear Science, 1982, pp. 2024-2031.

[12] G. L. Ries, G. S. Choi and R. K. Iyer, “Device-level transient fault modeling,” Digest of Papers, International Symposium on Fault-Tolerant Computing, 1994, pp. 86-94.

[13] G. Laguna and R. Treece, “VLSI modeling and design for radiation environments,” Proceedings of IEEE International Conference on Computer Design, 1986, pp. 380-384.

[14] HSPICE, “www.synopsys.com”.

[15] M. Singh, R. Rachala, and I. Koren, “Transient fault sensitivity analysis of analog-to-digital converters (ADCs),” Proceedings of IEEE Annual Workshop on VLSI, Apr. 2001, pp. 140-145.

[16] S. Kang and D. Chu, “CMOS circuit design for the prevention of single event upset,” Proceedings of IEEE International Conference on Computer Design, 1986, pp. 385-388.

[17] H. Cha, and J. H. Patel, “Latch design for transient pulse tolerance,” Proceedings of IEEE International Conference on Computer Design, 1994, pp. 385–388.

[18] M. Singh, and I. Koren, “Reliability enhancement of analog-to-digital converters (ADCs),” IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Oct. 2001, pp. 347-353.

[19] R. Rivest, “The RC5 encryption algorithm,” Proceedings of the 1994 Leuven Workshop on Fast Software Encryption (Springer 1995), pp. 86-96.