Transient Fault Sensitivity Analysis of Analog-to-Digital Converters (ADCs)

Mandeep Singh , Ravinder Rachala and Israel Koren
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003
E-mail: {msingh,rrachala,koren}@ecs.umass.edu

Abstract

Reliability of systems used in space, avionic and biomedical applications is highly critical. Such systems consist of an analog front-end to collect data, an ADC to convert the collected data to digital form and a digital unit to process it. It is important to analyze the fault sensitivities of each of these to effectively gauge and improve the reliability of the system. This paper addresses the issue of fault sensitivity of ADCs. A generic methodology for analyzing the fault sensitivity of ADCs is presented. A novel concept of "node weights" specific to α-particle induced transient faults is introduced to increase the accuracy of such an analysis.

1. Introduction

Fault sensitivity analysis allows the testing of the susceptibility of a circuit to different kinds of faults. This kind of study is necessary for space, military, avionics and biomedical applications. The purpose of such an analysis is to identify *critical blocks* in the circuit which are more susceptible to faults so that they can be redesigned for better reliability. Furthermore, different architectures can be weighed in terms of their sensitivities to faults and based on design tradeoffs, a suitable architecture can be chosen.

Two types of faults have been known to affect the proper working of a circuit: *permanent* and *transient*. Whereas permanent faults can be introduced during the fabrication stage and in the field, transient faults are caused in the field due to Electro Magnetic Interference (EMI) such as power transients, crosstalk and α-particle hits in radiation intense environments like space. The effect of transient faults is to temporarily change the behavior of the circuit often resulting in erroneous outputs. This type of faults has been known to account for 80% or more failures in digital systems [1, 2]. Since this might be catastrophic in critical applications, these circuits usually incorporate some measures to increase their fault tolerance.

The reliability of a system is determined by the fault tolerance of its constituent blocks. Systems in space, biomedical and avionics applications consist of an analog front-end to collect data for control and observation purposes and a digital unit which processes the collected data. Digital circuits have been studied extensively for their sensitivity to transient faults [3, 11] and many techniques have been suggested to improve their fault tolerance [4, 11]. In contrast, very little has been done to address the issue of fault tolerance in analog circuits and ADCs which are integral parts of such mixed-signal circuits. Hence, it is necessary to explore techniques to increase the fault tolerance of ADCs.

The process of increasing the tolerance of a circuit to transient faults can be divided into two steps:

1) Grading blocks of the circuit based on their sensitivities to transients and identifying *critical blocks*.

2) Increasing the fault tolerance of the identified *critical blocks*.

This paper addresses the first step. A methodology for ADC fault sensitivity analysis to α-particle induced transient faults is presented. A *folding and interpolating* ADC [5] and a *successive approximation* ADC [6] were analyzed using this methodology. The fault injection experiments were done on transistor level schematics using Hspice.

The paper is organized as follows. Section 2 presents the fault model used to model transient α-particle hits. Section 3 gives a brief description of the two ADCs analyzed. The simulation strategy used to conduct fault injection experiments on the ADCs is discussed in Section 4. The results of the fault injection experiments are presented in Section 5. Section 6 concludes the paper.

2. Transient Fault Model

Several transient fault models have been proposed in [7, 8]. Since this work concentrates on α-particle induced transients, the double exponential α-particle transient model for the injection current, I_{inj}, proposed

in [7] is used.

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

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. Injected charge level is a function of the angle at which the α-particle hits. I_0 can be positive or negative depending on whether the α-particle hits an NMOS drain or a PMOS drain [9]. Figure 1(a) shows the drain of a PMOS transistor and the effect of the injected charge. An α-particle hit generates electron-hole pairs along its trajectory. These charge carriers drift under the influence of the electric field across the junction giving rise to an injection current (I_{inj}) that can be modeled by equation (1). V is the initial voltage on the node (drain of the PMOS), dV is the voltage change due to the α-particle hit and is dependent on I_{inj} and the load connected to the node.

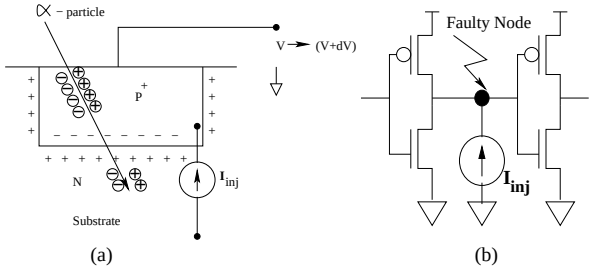

Figure 1: **(a) α-particle hit on the drain of a PMOS transistor (b) The α-particle hit modeled as a current source.**

Figure 1(b) shows the current source equivalent model of the transient fault caused by an α-particle hit.

3. Analog to Digital Converters

Analog to Digital Converters are integral parts of data acquisition systems and act as an interface between analog blocks that acquire the data and digital blocks that process the data. ADCs can be broadly classified into high-speed or high-accuracy architectures. High-speed architectures include *flash*, *folding and interpolating*, *pipelined*, *multi-step* and *interleaved* ADCs [12]. High-accuracy architectures include *successive approximation*, *delta-sigma* and *integrating* ADCs [12]. These two categories tradeoff speed vs accuracy. Based on the demands of the application, one of these ADCs can be chosen after carefully weighing the tradeoffs.

For the purpose of our study we have selected one ADC from each category. A brief description of the working of the selected ADCs follows.

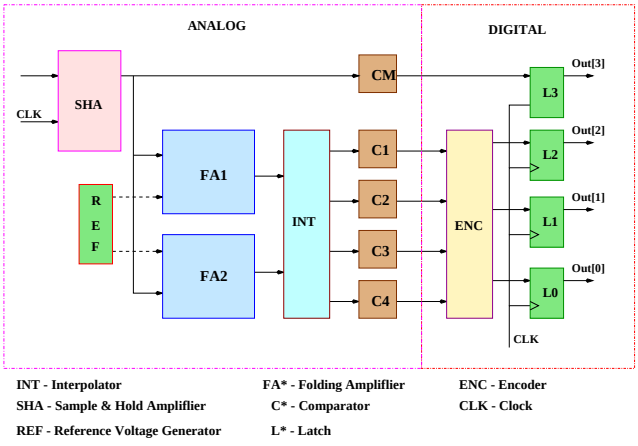

Figure 2: **Block diagram of a 4 -bit Folding and Interpolating ADC.**

1) *Folding and Interpolating (FI) ADC* [5]: Figure 2 shows the block diagram of a *folding and interpolating* ADC. The sample and hold amplifier (SHA) samples the input and the sampled input is fed to two folding amplifiers (FA1 and FA2) and a comparator (CM) which generates the most significant bit. The interpolating block (INT) interpolates between the folding amplifier outputs. The INT block output is fed to the encoder (ENC) which generates the three least significant bits of the final digital output.

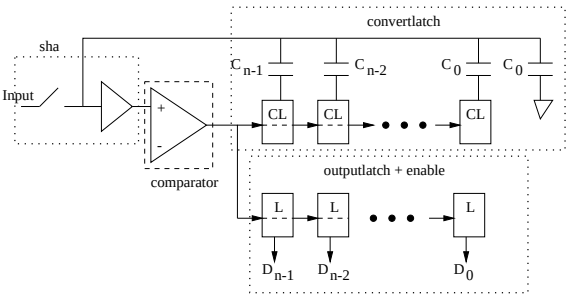

Figure 3: **Successive Approximation ADC.**

2) *Successive Approximation (SA)* [6]: Figure 3 shows the block diagram of a typical charge redistribution implementation of the *successive approximation* architecture. This implementation progresses like a binary search algorithm to arrive at the final digital output with an error of no more than 0.5 V_{LSB}. The objective during this conversion is to drive the difference between the DAC (convertlatch) output and the sam-

pled input to zero. One bit is converted in each cycle starting with the most significant bit. Hence, it takes N cycles in all to produce an N-bit digital output. A precise capacitor matching is required for this conversion. Current fabrication technologies cater to this requirement quite effectively.

4. Fault Sensitivity Analysis

There are different approaches to investigate the effects of transient faults. Hardware prototyping has been used [10] but it is too time consuming and expensive. Simulation based approaches include *exhaustive* and *Monte-Carlo* methods. *Exhaustive* simulations are accurate but become intractable for large designs. *Monte-Carlo* methods, though tractable for large designs, are not as accurate. Since the ADCs which have been analyzed in this work are relatively small we preferred the exhaustive simulation approach.

Traditionally, fault conditions in these simulation strategies have been varied along three dimensions: space, time and injection level. It is important to consider varying the inputs to the circuit, since this can have a bearing on selecting *critical blocks* for redesign. This is due to the fact that a block identified as a *critical block* for one input may not be as sensitive for another input. Hence, *critical blocks* should be identified based on the distribution of the input values. The circuit should be optimized for input values which are the most probable.

The design flow of ADCs can be broadly classified into three steps: 1) Choosing the architecture based on the requirements and specifications of the application. 2) Schematic entry of the selected architecture and functional verification. 3) Final layout design of the circuit and a re-verification with parasitics. Fault sensitivity analysis should be addressed as early as possible in the design cycle to avoid time consuming iterations. Since fault conditions have to be varied spatially, the physical design step (3) is an ideal point to address this issue. However, the complexity of the layout level database and the design effort needed to create the layout emphasize the need to move the analysis to an earlier stage. As we go up in the design cycle we should expect to pay a penalty in terms of the accuracy of the results.

Fault sensitivity analysis at the transistor level schematic can be done by selecting nodes in the circuit and injecting α-particle transients at these nodes. The fault sensitivity of a block is defined as the probability that an α-particle hitting the block will result in a circuit failure and is denoted by POF (Probability of Failure). For a given input voltage, POF is calculated as follows. We denote by n the number of circuit nodes in a block. An α-particle transient is injected into each node of the block and we denote the outcome of the experiment by E_i:

$$E_i = \begin{cases} 1 & \text{if the injection into node } i \text{ results in a failure} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The POF is now defined as

$$POF = \frac{1}{n} \sum_{i=1}^{n} E_i \tag{3}$$

This calculation assigns equal weights to all nodes, which may cause inaccuracies since the areas of different nodes may vary considerably. A higher accuracy can be achieved by assigning to each node a weight which is proportional to the area that it consumes. However, a circuit node may map onto two types of area in the layout: *fault-insensitive area* (interconnect) and *fault-sensitive area* (terminals of transistors connected to the node) (see Figure 4). It is known that an α-particle hit has a potential of resulting in an error only if it hits the active area (*fault-sensitive* area) of a transistor [11]. A hit at the interconnect (*fault-insensitive* area) will not cause a transient fault because of the lack of a significant electric field in that area. We therefore,

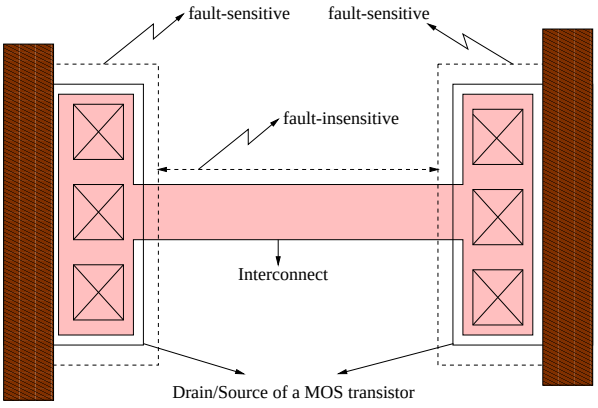

Figure 4: **Constituents of a node in the layout.**

assign to node i a weight, denoted by w_i, given by

$$w_i = \frac{A_{s,i}}{\sum_{i=1}^{n} A_{s,i}} \tag{4}$$

where $A_{s,i}$ the area of the fault-sensitive portion of node i. The sizes of the transistors in the schematics can serve as a good estimate for $A_{s,i}$. We now calculate

the POF as

$$POF = \sum_{i=1}^{n} w_i E_i \quad (5)$$

Following are the steps involved in the fault sensitivity analysis of an ADC:

1) Calculate *weights* of the nodes (w_i).

2) Perform transient fault simulations on all nodes.

3) Use equation (5) to calculate the sensitivity of the constituent blocks.

For a comprehensive evaluation of fault susceptibility it is necessary to perform a full transient simulation of the system in the presence of the appropriate transient faults. Hspice was used for the simulations. The following section shows the results of the simulations done using the *node weights* described above.

5. Results

Transient fault injection experiments were performed on 4-bit transistor level implementations of *successive approximation* and *folding and interpolating* ADCs. The results obtained from the simulations have been used to grade the fault sensitivities of the blocks in the ADC. The simulations were done for one representative input each in the lower, upper and middle ranges.

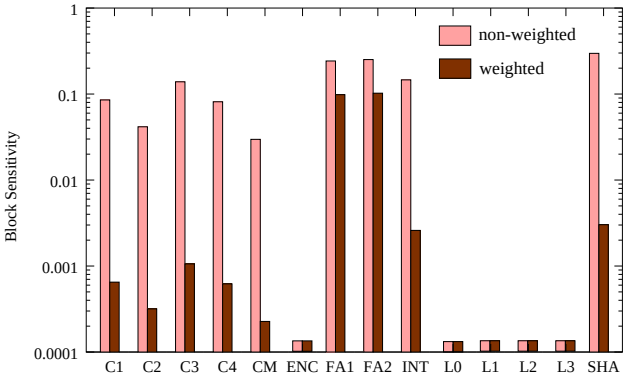

Figure 5: **Weighted vs Non-weighted (log scale) approach (FI)(averaged over all inputs).**

Figures 5 and 6 show the block sensitivities with the *non-weighted* and the *weighted* approach. These figures show that the less accurate non-weighted analysis may lead to incorrect conclusions. For example, SHA in Figure 5 has the highest block sensitivity according to the non-weighted analysis but has a considerably lower sensitivity than FA1 and FA2 according to the weighted analysis.

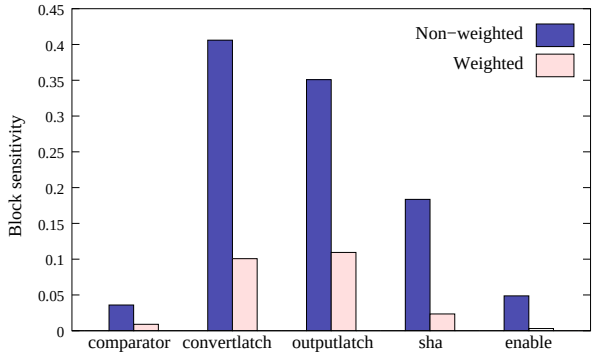

Figure 6: **Weighted vs Non-weighted approach (SA)(averaged over all inputs).**

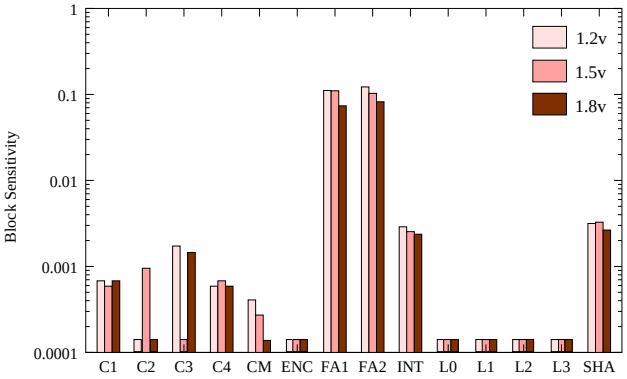

Figure 7: **Block sensitivity (log scale) variation with inputs (FI).**

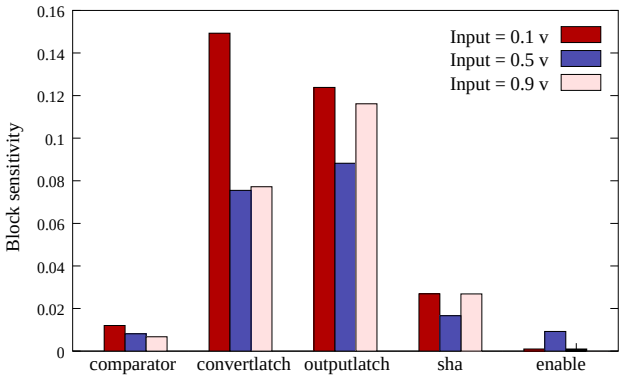

Figure 8: **Block sensitivity variation with inputs (SA).**

Figures 7 and 8 show bar-graphs with sensitivities of the blocks for the three inputs. It is evident from these figures that the sensitivities of some blocks to α-particle hits vary from one input value to another (C2, C3 in Figure 7 and outputlatch, convertlatch in

Figure 8). Comparators (C*) in FI were found to be more susceptible when their output is a logic 0. This corresponds to an ADC input in the range of 1-1.1v and 1.42-1.52v for the comparator C2. Hence, it can be concluded that C2 is relatively more sensitive in these input ranges.

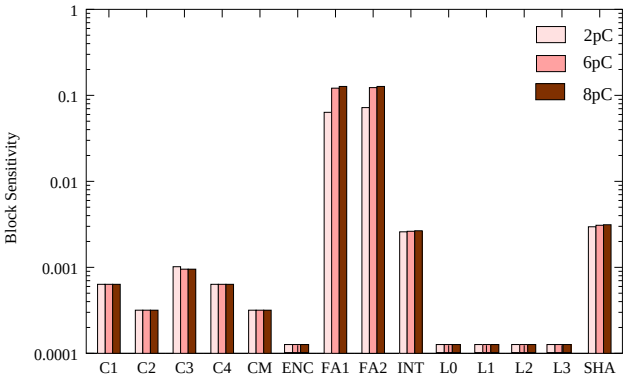

Figure 9: **Block sensitivity (log scale) variation with injection levels (FI).**

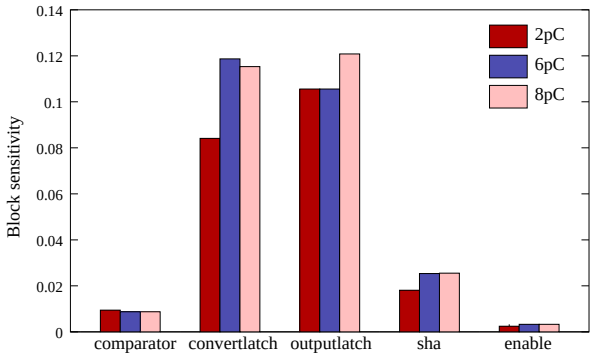

Figure 10: **Block sensitivity variation with injection levels (SA).**

Figures 9 and 10 show that the sensitivities of blocks do not vary considerably if the injection level is more than 6 pico-Coulumbs (pC). Figures 11 and 12 show that for lower injection levels the ordering of *critical blocks* might change (SHA is more sensitive than FA1 from 0pC-0.25pC in Fig. 11). These figures also show that beyond a certain injection level there is no further increase in block sensitivity.

Figure 13 shows the number of faults resulting in errors for faults injected at different time instances for a successive approximation ADC. The results indicate that the ADC is more susceptible to α-particle hits during the early part of each bit conversion cycle.

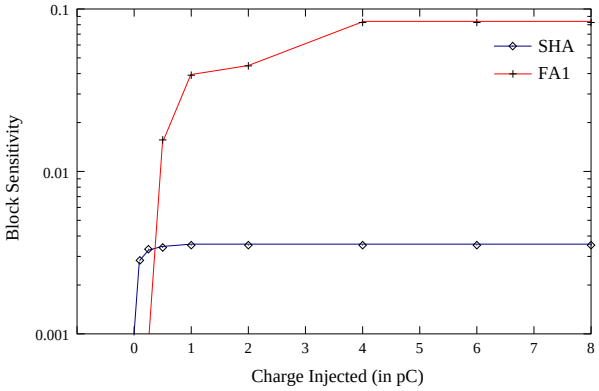

Figure 11: **Block sensitivities (log scale) variation with injections (FI).**

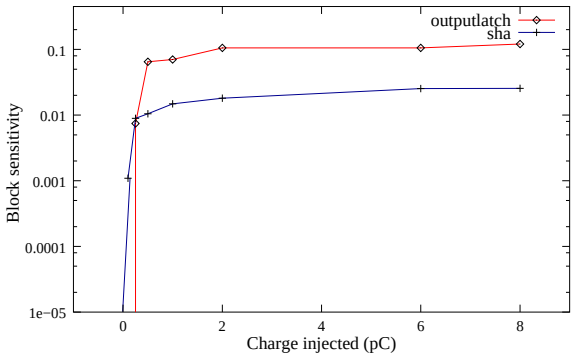

Figure 12: **Block sensitivities (log scale) variation with injections (SA).**

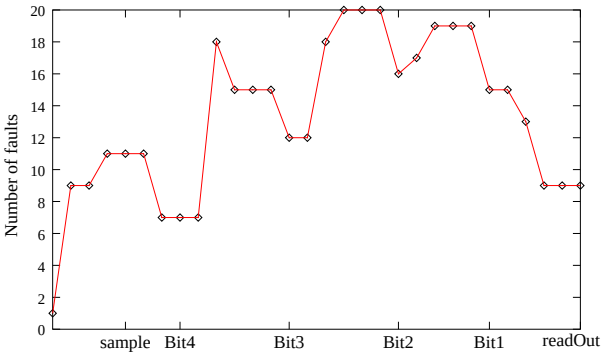

Figure 13: **Number of errors at different fault injection times (SA).**

The block sensitivities which have been presented so far treat all faults uniformly. However, some faults may result in larger errors than other faults, at the outputs of the ADC. We define the relative error, denoted by

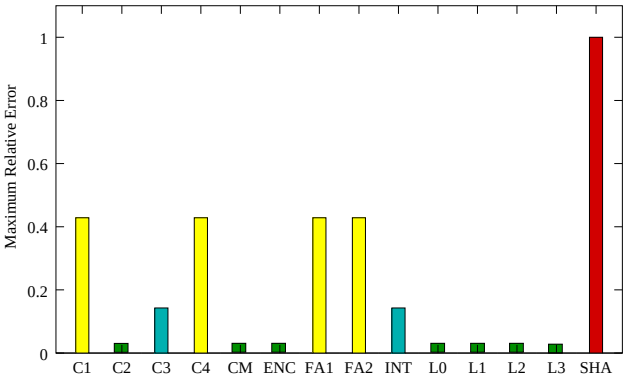

Figure 14: **Maximum relative error (FI).**

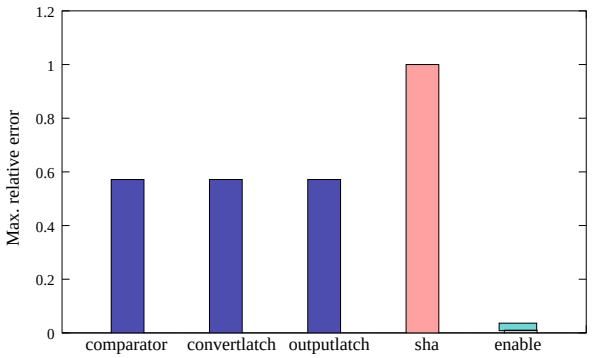

Figure 15: **Maximum relative error (SA).**

E_{rel}, as

$$E_{rel} = \triangle V / V_{exp} \tag{6}$$

where V_{exp} is the expected correct output and $\triangle$V is given by

$$\triangle V = |V_{err} - V_{exp}| \tag{7}$$

where V_{err} is the erroneous output. Figures 14, 15 show the maximum relative errors due to each block. Our results show that as we get to blocks closer to the input the maximum relative error increases reaching a peak for the sample and hold amplifiers (SHA in Figure 14, sha in Figure 15).

6. Conclusions

A generic methodology for the transient fault sensitivity analysis of ADCs has been presented. This methodology was used to determine the sensitivities of different blocks in the *successive approximation* and *folding and interpolating* ADCs and identify *critical blocks*. Issues in grading the blocks according to their fault sensitivities were pointed out. A novel concept of *node weights* specific to α-particle induced transient faults was introduced to improve the accuracy of such an analysis.

This methodology can be used to analyze the fault sensitivities of the constituent blocks in any ADC architecture at an early stage in the design cycle thus reducing concept-to-silicon time.

Acknowledgment: This work was supported in part by JPL under contract 961294 and by NSF under contract MIP–9710130.

7. References

[1] H. Ball and F. Hardy, "Effects and detection of intermittent failures in digital systems," *Proc. of 1969 FJCC,AFIPS Conf.,* Vol. 35, pp. 329-335.

[2] R. Iyer and D. Rosetti, "A statistical load dependency of CPU errors at SLAC," *Proc. FTCS-12 (Santa Monica, CA),* 1982.

[3] F. L. Yang and R. A. Saleh, "Simulation and Analysis of Transient Faults in Digital Circuits," *IEEE Journal of Solid-State Circuits,* Vol. 27, No. 3, pp. 258-264, Mar. 1992.

[4] A. Antola, R. Negrini, M. Sami and N. Scarabottolo, "Tolerance to transient faults in microprogrammed control units," *IEEE Transactions on Reliability,* Vol. 27, No. 3, pp. 258-264, Mar. 1992.

[5] P. Vorenkamp and R. Roovers, "A 12-b, 60-MSample/s cascaded folding and interpolating ADC," *IEEE Journal of Solid-State Circuits,* Vol. 32, No. 12, pp. 1876-1886, Dec. 1997.

[6] Z. Zhou, B. Pain and E. R. Fossum, "CMOS active pixel sensor with on-chip successive approximation analog-to-digital converter," *IEEE Transactions on Electron Devices,* Vol. 44, No. 10, pp. 1759-1763, October 1997.

[7] G. C. Messenger, "Collection of charge on junction nodes from ion tracks," *IEEE Transactions on Nuclear Science,* Vol. NS-29, No. 6, pp. 2024-2031, Dec. 1982.

[8] G. Laguna and R. Treece, "VLSI modeling and design for radiation environments," *Proc. IEEE Int. Conf. Comput. Design,* pp. 380-384, Oct. 1986.

[9] H. Cha, E. M. Rudnick, J. H. Patel, R. K. Iyer and G. S. Choi, "A gate level simulation environment for alpha-particle-induced transient faults," *IEEE Transactions on Computers,* Vol. 45, No. 11, pp. 1248-1256, Nov. 1996.

[10] H. Amer, M. Cortes and E. McCluskey, "Robust dynamic recovery mechanisms," *Proc. IEEE Int. Conf. Comput. Design,* pp. 310-313, 1987.

[11] S. Kang and D. Chu, "CMOS circuit design for the prevention of single event upset," *Proc. Int. Conf. Comput. Design,* pp. 385-388, Oct. 1986.

[12] B. Razavi, *Principles of data conversion system design,* IEEE press, 1995.