Fault Sensitivity and Tolerance of Successive Approximation and Δ-Σ Analog-to-Digital Converters (ADCs) [*]

Mandeep Singh and Israel Koren

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003, USA
E-mail: {msingh,koren}@ecs.umass.edu

Abstract

The reliability of ADCs used in highly critical systems can be increased by applying a two-step procedure starting with sensitivity analysis followed by redesign. The sensitivity analysis is used to identify the most sensitive blocks which could then be redesigned for better reliability by incorporating fault tolerance. This paper illustrates the steps involved in incorporating fault tolerance in an ADC. Two redesign techniques to improve the reliability of a circuit are presented. Novel selective node resizing algorithms for increased tolerance against α-particle induced transients are discussed.

1. Introduction

Critical systems used in space, avionics and biomedical applications have to be highly reliable since the effect of a fault in these systems can be catastrophic. The reliability of these systems can be increased by redesigning them for improved fault tolerance. The system under redesign should undergo a fault sensitivity analysis before and after redesign to gauge the improvement in the reliability of the system. Fault sensitivity analysis involves injection of faults either in the actual hardware or in software through simulation. The latter method is preferable since the former requires a prototype which is expensive. The latter also enables an analysis early in the design process thus eliminating costly redesign.

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 [1, 2] and many techniques have been suggested to improve their fault tolerance [2]. In contrast, very little has been done to address the issue of fault tolerance in analog circuits and ADCs which are integral parts of most 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:

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

(ii) Increasing the fault tolerance of the identified *critical blocks.*

Such a process can yield the most cost-effective way to incorporate fault tolerance into the design. This work addresses both of these steps by first proposing a methodology to analyze the sensitivity of an ADC and then by suggesting techniques to increase the reliability of the ADC. The fault injection experiments, for gauging the sensitivity of the designs addressed in this work, were performed for α-particle induced transients. This is because 85% or more computer system failures are known to be caused by transient faults [3] and among the energetic nuclear particles that can cause a transient fault, α-particles have been identified to be the most damaging [4]. However, the techniques developed for these faults can be extended to transient faults caused by other sources. Though α-particles are mainly found in space, trace amounts of α-particles are also found in ICs on the ground due to decay of radioactive elements present in the packaging material or solder [4]. Extraterrestrial cosmic rays which bombard earth continuously are another source of α-particle radiation. Thus the applicability of this work is not restricted to systems in outer space but also to other ground based critical systems.

The paper is organized as follows. Section 2 provides a functional description of the ADCs addressed in this work. In Section 3, the fault sensitivity analysis methodology is discussed. The results of fault sensi-

[*] A preliminary version of this paper was presented in ISQED 2002 [7].

tivity analysis of the Δ-Σ and successive approximation (SA) ADCs are presented in Section 4. Section 5 illustrates the reliability improvements achieved by incorporating redesign techniques like opting for more robust implementations and selective node resizing. Finally, Section 6 summarizes the findings of the work.

2. 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. In this work two representative high accuracy architectures, namely, *successive approximation* and Δ-Σ ADC are analyzed.

2.1. Successive Approximation ADC

Figure 1 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 sampled 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.

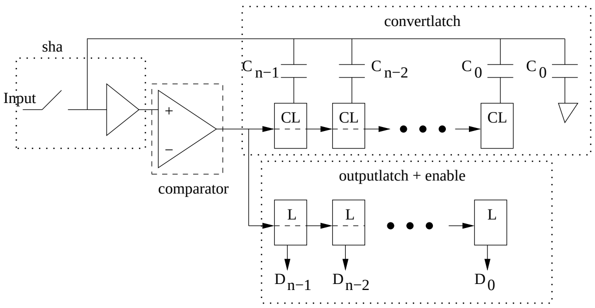

Figure 1: **Successive Approximation architecture**

2.2. Δ-Σ ADC

Figure 2 shows the block diagram of a Δ-Σ ADC. The Δ-Σ modulator is an analog component and the decimation filter is a digital component. The most common implementation of the Δ-Σ modulator provides an oversampled serial output which is a digital representation of the input signal. This serial output thus obtained has high frequency noise in addition to the signal information. The digital decimation filter stage (see Figure 2), following the modulator, filters out this noise and provides a high resolution output.

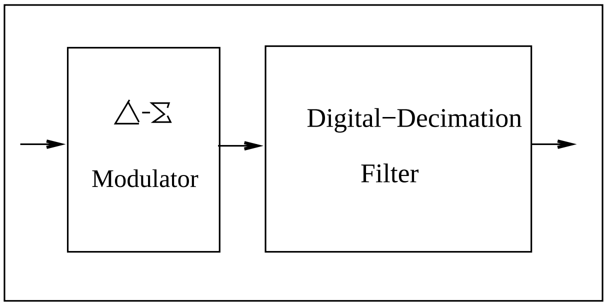

Figure 2: Δ-Σ **converter**

3. Fault Sensitivity Analysis Methodology

Fault sensitivity analysis enables the designer to identify *critical blocks* which can be redesigned for improving the reliability of the system. Our recent work [5, 6] has shown that the fault sensitivity analysis for an α-particle induced transient can be performed at an early stage in the design cycle of mixed-signal circuits.

Two metrics namely, the Probability Of Failure (POF) [5] and the Average Relative Error (ARE) [6] have been proposed in our previous work. To calculate the POF for a circuit with n nodes we first inject a current representing an α-particle hit into each node (this is simulated in Hspice) and observe the output. The outcome of such an experiment is denoted by E_i

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

Since the effect of an α-particle hit depends on the level of injected current, the input voltage and the time of injection we calculate the average outcome, denoted by $\bar{E_i}$, as

$$\bar{E_i} = \frac{1}{k}\sum_{i=1}^{k} E_i \quad , \qquad\qquad k = p \cdot q \cdot r \tag{2}$$

where p is the number of input values for which the simulation was performed, q is the number of injection levels considered and r is the number of time instances at which faults were injected. We then assign each node i a weight, denoted by w_i, which is proportional to the size of its sensitive (to faults) area. This reflects the fact that larger area nodes are more likely to be hit by an α-particle. The sensitive area does not include the area of the interconnects since a particle hit in this area will not cause a fault [5]. Thus, the POF metric is defined as

$$POF = \sum_{i=1}^{n} w_i \bar{E}_i \quad , \qquad w_i = \frac{A_{s,i}}{\sum_{i=1}^{n} A_{s,i}} \tag{3}$$

where $A_{s,i}$ is the area of the fault-sensitive portion of node i. Whereas the POF gives an idea about the frequency of errors, the ARE assigns more weightage to the magnitude of errors. The ARE is given by

$$ARE = \sum_{i=1}^{n} w_i \bar{E}_{rel,i} \quad , \qquad \bar{E}_{rel,i} = \frac{1}{k} \sum_{i=1}^{k} E_{rel,i} \tag{4}$$

where w_i is calculated using (3) and $E_{rel,i}$, the relative error due to an injection at node i is given by

$$E_{rel,i} = \triangle V / V_{exp} \quad , \qquad \triangle V = |V_{err} - V_{exp}| \tag{5}$$

where V_{exp} is the expected correct output and V_{err} is the erroneous output observed when node i is hit by an α-particle.

The choice of a metric for sensitivity analysis should be based on the design objective. A design objective of lowering the frequency of errors would call for the usage of POF for sensitivity analysis. In contrast, if reducing the magnitude of error is the design objective, ARE should be used. Following are the steps involved in the fault sensitivity analysis of an ADC: (i) Calculate *weights* of the nodes (w_i) (ii) Perform transient fault simulations on all nodes (iii) Based on the design objectives, use equation (3) or (4) to calculate the sensitivity of the constituent blocks.

4. ADC Sensitivity Analysis

Transient fault injection experiments were performed on 4-bit transistor level implementations of the *successive approximation* (SA) and the Δ-Σ ADCs. The results obtained from the simulations have been used to grade the fault sensitivities of the blocks in the ADCs. Since the choice of a metric (POF or ARE) for sensitivity analysis is not dependent on the ADC architecture, any metric could have been used. In this work, POF has been used as the sensitivity measure for the SA ADC and ARE has been used for the Δ-Σ ADC.

4.1. Successive Approximation

Sensitivity analysis results for the 4-bit implementation of the SA ADC in our earlier work [5] identified the convertlatch and the outputlatch (Figure 1) as the *critical blocks* (see Figure 3). Further analysis performed on the convertlatch revealed that out of the four latches in convertlatch, the latch containing the most significant bit (MSB) is the most sensitive (Figure 4). This is because the MSB is used for generating the rest of the bits of the ADC. Therefore, an injection at the MSB has a higher probability of turning into an error. Figure 5 shows that for lower injection levels the ordering of *critical blocks* might change (the sample and hold (sha) is more sensitive than the outputlatch from 0pC to 0.25pC). This figure also shows that beyond a certain injection level there is no further increase in the sensitivity of the blocks in the SA ADC. Figure 6 shows the maximum relative error due to each block. Our results indicate 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 1).

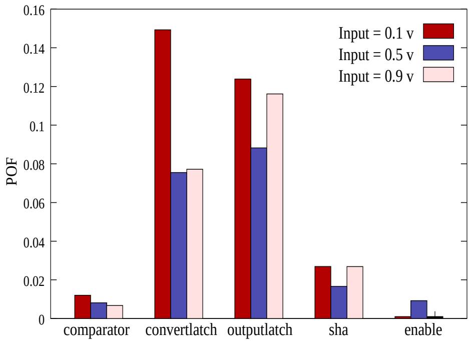

Figure 3: **Block sensitivity variation with inputs (SA), q=14, r=32**

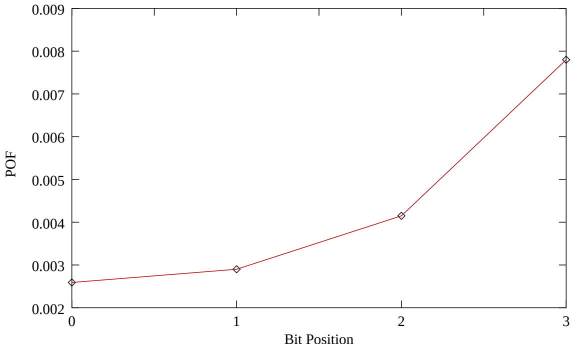

Figure 4: **Sensitivity of the four latches in convertlatch (SA), p=3, q=14, r=32**

4.2. Δ-Σ ADC

Like the SA ADC, the Δ-Σ ADC takes several cycles to generate the final ADC output. The number of cycles required to generate the final output is governed by the oversampling ratio of the Δ-Σ converter.

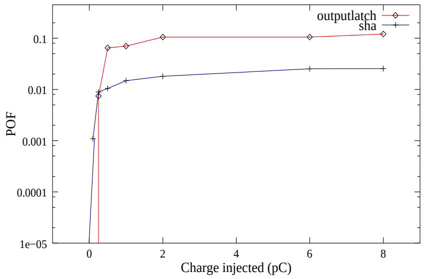

Figure 5: **Block Sensitivities (log scale) variation with injections (SA), p=3, r=32**

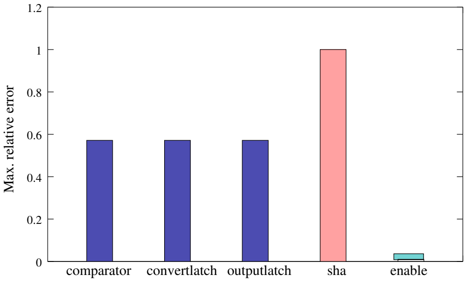

Figure 6: **Maximum Relative Error (SA), p=3, q=14, r=32**

For the 4-bit Δ-Σ ADC implemented for this work the Δ-Σ modulator generates eight bits in eight clock cycles. These eight bits are then digitally filtered and the final 4-bit ADC output is generated at a decimated frequency.

A sensitivity analysis of the Δ-Σ ADC reveals that the digital decimation filter is the most *critical block* (see Table 1).

Block	Input (V)		ARE
	2	3	
Δ-Σ Modulator	0.00002	0.00006	0.00004
Decimation Filter	0.00426	0.00573	0.00500

Table 1: **Δ-Σ ADC sensitivity analysis, p=2, q=8, r=16**

5. Redesign Techniques

A sensitivity analysis identifies *critical blocks* that the designer can concentrate on to improve the reliability of the system. Fault tolerance of a block can be improved in one of two ways: (1) Evaluating the sensitivities of alternative implementations of a block and selecting the most robust implementation. (2) Affecting design changes in the existing implementation.

It is essential to gauge the improvement that each of these techniques offers as this would help the designer to decide on the most cost-effective fault tolerance design strategy. The following sections describe two such techniques and also illustrate the amount of sensitivity improvement that can be gained by employing them. In [6] the technique of opting for alternative robust implementations has been used for the Flash and Folding and Interpolating ADC. A different approach is required for incorporating fault tolerance in the two ADCs analyzed here.

5.1. Successive Approximation

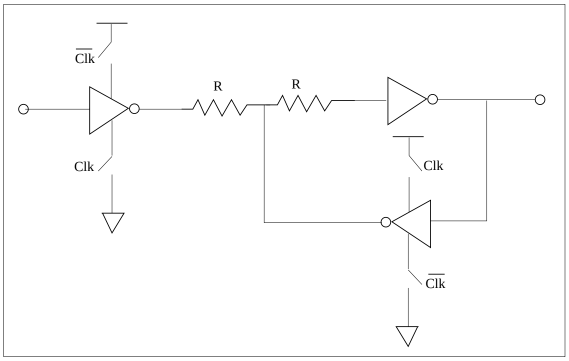

Figure 7: **Transient Pulse Tolerant Latch (TPTL)**

The sensitivity analysis of the 4-bit SA ADC had identified the convertlatch and the outputlatch as the most *critical blocks* (Section 4.1). The Transient Pulse Tolerant Latch (TPTL) proposed in [8] was considered for sensitivity reduction (see Figure 7). The resistors (R) added in TPTL filter out the transients arriving at the input of the latch thus hardening it against transients. With increasing values of the resistors the latch becomes more fault tolerant but at the same time a performance penalty has to be expected. Figures 8 and 9 show that the ARE reduces by 50% and the POF reduces by as much as 45% for a resistance value of 15K. These figures also show that the ARE may sometimes mask sensitivity improvements. Though the ARE (Figure 8) remains almost constant for an input value of 0.9v, the POF (Figure 9) reduces by as much as 50%. Therefore, the choice of a metric to gauge the sensitivity should be based on whether the candidate application requires a reduction in the frequency of errors or the magnitude of error. The improvement in sensitivity is achieved at a cost of performance. Figure 10 shows that the delay of a latch increases with an increasing resistance in a superlinear fashion. Therefore,

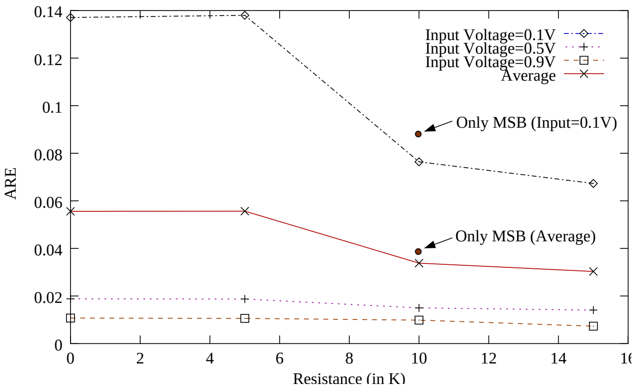

Figure 8: **Variation in ARE with increasing resistance (R),** p**=3,** q**=8,** r**=32**

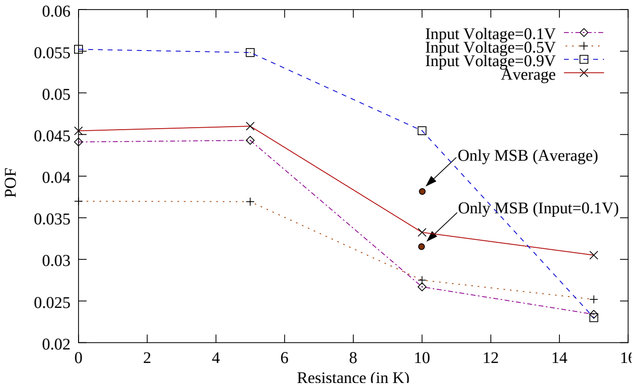

Figure 9: **Variation in POF with increasing resistance (R),** p**=3,** q**=8,** r**=32**

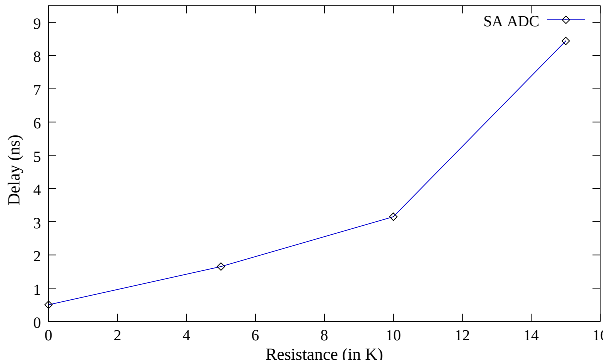

Figure 10: **Variation in performance with increasing resistance (R),** p**=3,** q**=8,** r**=32**

the final resistance value should be chosen by taking the performance degradation into account. In our experiments, a resistance of 10K seems to provide a good compromise between sensitivity improvement and performance degradation. The overhead can be further reduced by replacing only the most sensitive latch in convertlatch by TPTL. Figure 4 shows that the latch holding the MSB is the most sensitive. Sensitivity analysis performed after replacing only the MSB latch in convertlatch by TPTL (instead of all the latches in this block) revealed that almost identical improvements in ARE are obtained (see Figure 8). However, the improvement in POF is not comparable (see Figure 9). This further fortifies the argument of the selection of a metric (POF or ARE) being design objective dependent.

5.2. Δ-Σ ADC

Our earlier work [6] presented an implementation of the Δ-Σ modulator with redundancy which achieved sensitivity gains of 65.8% with a 75% overall area overhead. Two techniques for reliability enhancement with a lesser overall area overhead are presented here. The first technique uses TPTL as a more robust alternative for the latches. The second technique uses transistor sizing which can lead to substantial reliability improvements [6]. This improvement however, is achieved at the cost of considerable area overhead. An alternative strategy is proposed where nodes are selectively resized thus reducing the area overhead. Two algorithms for resizing selective nodes for reliability improvement have been experimented with. These algorithms were applied to a 2-bit counter (with 73 circuit nodes) used in the digital decimation filter in the Δ-Σ ADC and the variation of the sensitivity with sizing and by bounding the maximum injection level was analyzed and is presented below.

A. *Alternative Robust Implementation*

Since the digital decimation filter in the Δ-Σ ADC uses latches extensively, using the TPTL described in the previous section can lower the sensitivity of the decimation filter. Simulations were performed by replacing all latches with TPTL and the sensitivity of the Non-Fault Tolerant (NFT) and the Fault Tolerant (FT) versions were compared. A resistance of 10K was used in TPTL as the simulations performed for the SA ADC revealed that this value was the most cost-effective. Improvement in sensitivities of as much as 21% (Table 2) was observed. This improvement is achieved at a cost of performance (see Figure 10).

B. *Selective node resizing*

Nodes which will result in maximum sensitivity gains should be chosen as candidates for sizing. One of the schemes that can be followed sorts the nodes in decreasing order of their contribution to the overall sensitivity of the block. Out of this sorted list the first n nodes can be selected as candidates for resizing. For our experiments the twelve most sensitive nodes (n=12) in the counter (out of 73) were considered for

Input(V)	NFT (ARE)	FT (ARE)	% imp
2	0.00428	0.00341	20.3
3	0.00579	0.00497	14
Avg.	0.00504	0.00398	21

Table 2: Δ-Σ **ADC (NFT Vs FT) sensitivity,** p**=2,** q**=8,** r**=16**

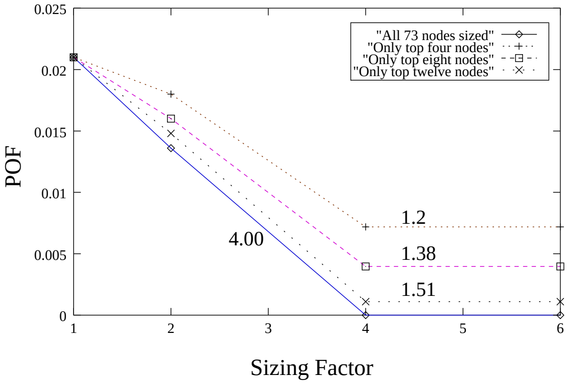

Figure 11: **Sensitivity variation with selective node resizing with injection level bounded by 1pC (the number on the curves indicate the corresponding fault-sensitive area increase factor)**

resizing. Figure 11 shows that for lower injection level bounds ($1pC$) the *most-sensitive-node-first* algorithm shows sensitivity improvement of as much as 60% with an area overhead of only 20%. The numbers on the curves indicate the area increase factor corresponding to the sizing factor which results in the lowest sensitivity among the sizing factors considered (sizing factor of 4 in Figure 11 and 6 in Figures 12,13,14). The above node selection scheme works well for low injection levels bounded by 1pC. However, the improvement is not so sizeable for higher injection level bounds (see Figure 12 where a sensitivity improvement of about 39% incurs an area overhead of 86%). This motivates a search for a better node selection scheme and an insight into why the improvement is not so sizeable for higher injection levels. The error offset at a node caused by an injection is dependent among other factors on the injection level and the transistor driving strength. Denote by ΔV the erroneous voltage offset $\Delta V = I_{inj} R_{on}$, where I_{inj} is the magnitude of the injected current and R_{on} is the resistance posed by the transistor connected to the node. ΔV can be reduced by lowering R_{on}, which can be achieved by sizing up the transistor. But if R_{on} is large then the transistor will have to be considerably sized before any gain in sensitivity can be seen. It is very likely that these kind of nodes will show up at the top of the sorted list of nodes. Thus, in such cases it is possible to achieve considerable sensitivity gains with a small area overhead by opting for an alternate scheme. In this scheme only the m nodes at the bot-

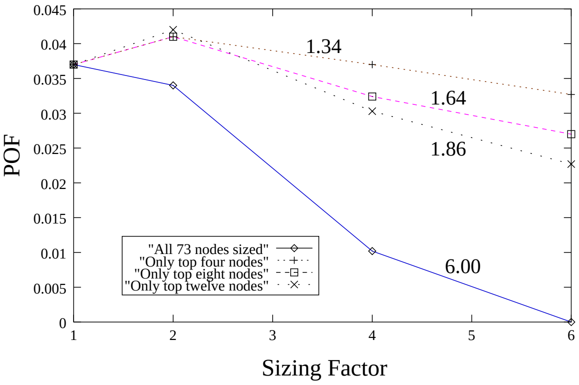

Figure 12: **Sensitivity variation with selective node resizing (bounded by 2pC)**

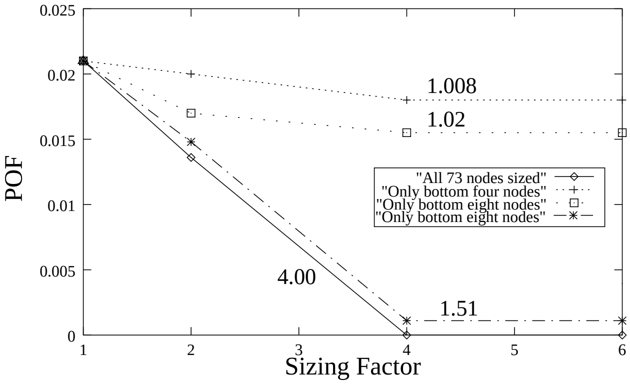

Figure 13: **Sensitivity variation with selective node resizing,** n**=12 (bounded by 1pC)**

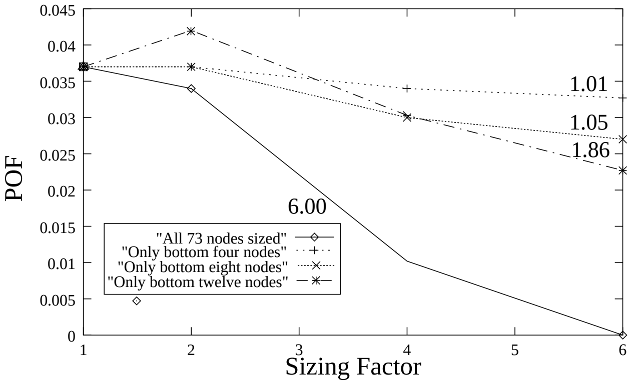

Figure 14: **Sensitivity variation with selective node resizing,** n**=12 (bounded by 2pC)**

tom of the sorted list of the n most sensitive nodes are sized. Note that since the m nodes are selected from a list of the n most sensitive nodes they are still quite sensitive. Figures 13 and 14 show that for higher injection bounds sizeable improvement in sensitivity, with a smaller area overhead (5% in Figure 14 Vs 64% in Figure 12 for 8 nodes), can be attained by opting for the alternate scheme (the "bottom four" and "bottom eight" curves in Figure 14 show the sensitivity variation when m is 4 and 8, respectively).

Therefore, for lower injection levels ($\leq 1pC$) resizing the n most sensitive nodes will prove beneficial and for higher injection levels ($\geq 2pC$) resizing the bottom m nodes in the sorted list of the n most sensitive nodes will prove more cost-effective. The selective node selection strategy provides sensitivity gains with minimum area overhead among the two strategies considered. This is an example of a strategy for effective node selection for reliability enhancement.

6. Effect of Parametric Variations on Sensitivity

Mixed-signal circuits like ADCs are typically designed for a certain range of device parameters to account for parametric variations which happen during fabrication. Thus, it is important to study the variation of the sensitivity of these circuits with parametric variations. Fault simulations were performed on Tabatabaei's comparator which was used in the successive approximation and Δ-Σ ADCs, by varying the width (W) and threshold voltage (V_{th}). A notation of $\pm par$ (in Tables 3, 4 and 5) implies that the parameter being varied, denoted by par, has been increased for one of the matched transistors and has been decreased for the other transistor.

%Var(W)	POF
0	0.26
±2	0.258
±4	0.252

Table 3: Parametric Variation W (transistor Width), p=3, q=8, r=18

Table 3 shows that width variation, which causes high mismatch, results in lower sensitivity. This is due to the fact that sizing up a transistor causes the sensitivity of the node connected to the transistor to decrease. The detrimental effect due to decreasing the width of the other matched transistor is subdued because the *fault-sensitive* area corresponding to that node also decreases. V_{th} variation (Table 4) also translates to an increase/decrease in the drive of the transistor based on whether the V_{th} is decreased or increased. However, this increase/decrease is achieved at no extra *fault-sensitive* area cost as opposed to the width variation. Therefore, a larger improvement in sensitivity can be seen. Note that for the last case in Table 4 the NMOS transistors and the PMOS transistors were varied by different amounts unlike the first three cases. This is because the % variation numbers were picked from a commercial process and the worst possible variation for the NMOS and PMOS was ±8% and ±13%, respectively. Table 5 shows the results of both W and V_{th} variation. The sensitivity increases for the first case (±2, ±4) because both V_{th} and W were increased for the same transistor. Therefore, although increasing the width resulted in an increase in the driving strength of the transistor, an increase in the V_{th} caused a reduction in the drive. The *fault-sensitive* area associated with the drain/source of the transistor increased without providing any driving strength benefits, thus yielding a higher sensitivity. The results show that the sensitivity variation, for the type of parametric variations considered, was limited.

%Var(V_{th})	POF
0	0.26
±4	0.2319
±8	0.235
±8,NMOS ±13,PMOS	0.238

Table 4: Parametric Variation V_{th} (threshold voltage), p=3, q=8, r=18

%Var(W,V_{th})	POF
0	0.26
±2,±4	0.351
±4,±4	0.276
±4,±8	0.2479

Table 5: Parametric Variation (W and V_{th}), p=3, q=8, r=18

7. Conclusions

The steps involved in incorporating fault tolerance in ADCs have been illustrated. *Critical blocks* in the Δ-Σ and SA ADC were identified and the benefits of redesign techniques like opting for more robust implementations and selective node resizing have been analyzed. Sensitivity gains of as much as 50% for the SA ADC and 21% for the Δ-Σ ADC were observed. It was also observed that selective node resizing provides sensitivity gains of as much as 60% for the 2-bit counter used in the Δ-Σ ADC. This improvement is achieved at a considerably less area overhead compared to the full circuit sizing solution. Lastly, the impact of certain kind of parameter variations on the sensitivity of the circuit was illustrated. For the kind of parametric variations considered the sensitivity numbers showed minor excursions.

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

8. References

[1] Yang F.L., and Saleh R.A., "Simulation and Analysis of Transient Faults in Digital Circuits," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 258–264, 1992.

[2] Kang S., and Chu D., "CMOS circuit design for the prevention of single event upset," in *Proceedings of IEEE International Conference on Computer Design*, Oct 1986, pp. 385–388.

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

[4] Ziegler J.F., et. al., "IBM Experiments in Soft Fails in Computer Electronics," *IBM Journal of Research and Development*, vol. 40, pp. 3–17, 1996.

[5] Singh M., Rachala R., and Koren I., "Transient Fault Sensitivity Analysis of Analog-to-Digital Converters (ADCs)," in *IEEE Annual Workshop on VLSI*, Apr. 2001, pp. 140–145.

[6] Singh M., and Koren I., "Reliability Enhancement of Analog-to-Digital Converters (ADCs)," in *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Oct. 2001, pp. 347–353.

[7] Singh M., and Koren I., "Incorporating Fault Tolerance in Analog-to-Digital Converters (ADCs)," in *International Symposium on Quality of Electronic Design (ISQED'02)*, March 2002, pp. 286–291.

[8] Cha H., and Patel J.H., "A Logic-Level Model for α-Particle Hits in CMOS Circuits," in *Proceedings of IEEE International Conference on Computer Design*, Oct 1993, pp. 538–542.