Fault Sensitivity Analysis and Reliability Enhancement of Analog-to-Digital Converters

Mandeep Singh, *Member*, IEEE, and Israel Koren, *Fellow*, IEEE

***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 Analog-to-Digital Converter (ADC) to convert the collected data to digital form and a digital unit to process it. Though considerable amount of research has been performed to increase the reliability of digital blocks, the same can not be claimed for mixed signal blocks. The reliability enhancement which we employ starts with fault sensitivity analysis followed by redesign. The data obtained from the sensitivity analysis is used to grade blocks based on their sensitivity to faults. The highly sensitive blocks can then be replaced by more reliable alternatives. The improvement gained by opting for more robust implementations might be limited due to the number of possible implementations. In these cases alternative reliability enhancement techniques such as adding redundancy may provide further improvements. The steps involved in the reliability enhancement of ADCs are illustrated in this paper by first proposing a sensitivity analysis methodology for α-particle induced transients and then suggesting redesign techniques to improve the reliability of the ADC. A novel concept of** *node weights* **specific to α-particle transients is introduced which improves the accuracy of the sensitivity analysis. The fault simulations show that, using techniques such as alternative robust implementations, adding redundancy, pattern detection and transistor sizing, considerable improvements in reliability can be attained.**

***Index Terms*— Fault Tolerance, Fault Sensitivity, Analog-to-Digital Converters, Alpha particles, Reliability, Transient Faults**

I. 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 undergoes a fault sensitivity analysis before and after the redesign to gauge the reliability improvement. 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 early analysis in the design phase thus eliminating costly redesign.

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 various 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 85% 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], [4] and many techniques have been suggested to improve their fault tolerance [4], [5]. In contrast, very little has been done to address the issue of fault tolerance in analog circuits and ADCs which are integral parts of almost all 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 transient faults and identifying *critical* (i.e., most sensitive) blocks.
(ii) Increasing the fault tolerance of the identified *critical blocks*.

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 α-particles have been identified as one of the energetic nuclear particles that can cause a transient fault. However, the techniques developed for these faults can be extended to transient faults caused by other sources. α-particles are found in space [6] and in trace amounts in ICs on the ground due to decay of radioactive elements present in the packaging material or solder [7]. Thus, the applicability of this work is not restricted to systems in outer space but also to other ground based critical systems.

This paper is organized as follows, Section II presents a taxonomy for ADCs and provides a brief functional description of the ADCs addressed in this work. Section III describes the sensitivity analysis methodology used. In Section IV the process of increasing reliability by opting for robust implementations and by introducing redundancy is discussed. Section V summarizes the findings of this work and Section VI discusses future work.

M. Singh is with Advanced Micro Devices in the Computation Products Group, Austin,TX 78704 USA.

I. Koren is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003, USA.

This work has been supported in part by NSF under contract MIP-9710130 and by JPL under contract 961294.

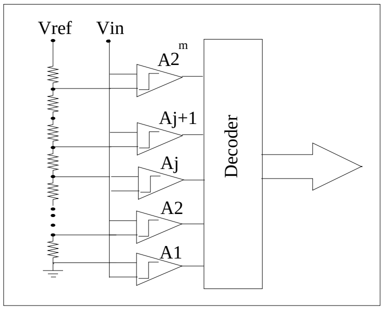

Fig. 1. A flash ADC

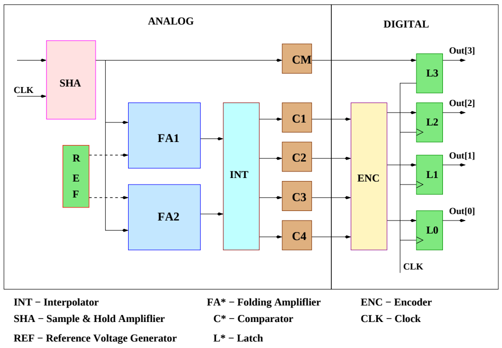

Fig. 2. A 4 -bit Folding and Interpolating ADC

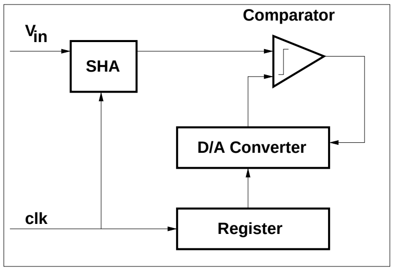

Fig. 3. Successive Approximation architecture

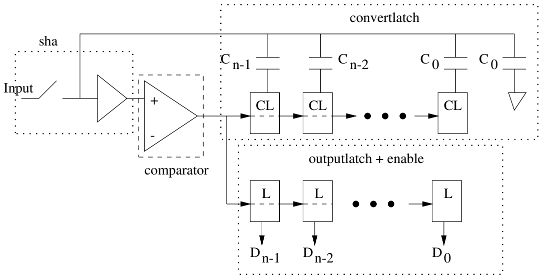

Fig. 4. Successive Approximation ADC

II. 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 and high-accuracy architectures. High-speed architectures include *flash*, *folding and interpolating*, *pipelined*, *multi-step* and *interleaved* ADCs [8]. High-accuracy architectures include *successive approximation*, *delta-sigma* and *integrating* ADCs [8]. 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. The following sections briefly describe the working of the ADCs which have been addressed in this work.

A. Flash ADC

This architecture is conceptually the simplest and potentially the fastest. It employs “parallelism” and “distributed” sampling to achieve high conversion speeds. Figure 1 shows a block diagram of an m-bit flash ADC. The circuit consists of 2^m comparators, a resistor ladder comprising 2^m equal segments and a decoder. The ladder subdivides the main reference into 2^m equally spaced voltages, and the comparators compare the input signal with these voltages. For example, if the analog input is between V_j and V_{j+1}, comparators A_1 through A_j produce 1s at their outputs while the rest generate 0s. Consequently, the comparator outputs constitute a thermometer code which is converted to binary by the decoder.

B. Folding and Interpolating ADC

The large input capacitance posed by the comparators at the input of flash ADCs led to the advent of folding and interpolating (FI) ADCs [9]. FI ADCs fold the information represented by the reference voltages which characterize the quantization levels. Figure 2 shows the block diagram of a 4-bit *folding and interpolating* ADC. The FA blocks in Figure 2 are folding amplifiers, each one of which is a series of cross-coupled differential stages [9]. The sample and hold amplifier (SHA) samples the input and feeds it to two folding amplifiers (FA1 and FA2) and a comparator (CM) which generates the most significant bit. The INT block 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.

C. Successive Approximation ADC

The Successive Approximation (SA) ADCs progress like a binary search algorithm to arrive at the final digital output with an error of no more than half the least significant bit. Figure 3 illustrates the successive approximation architecture, which consists of a front-end SHA, a comparator, a register (shift) and a Digital-to-Analog Converter (DAC). The (shift) register holds the bits that have been converted starting from the most significant bit (MSB). This digital pattern is then converted by the DAC to analog and this value is compared against the held input. The output of the comparator decides the value of the next bit. Thus, the final m-bit digital pattern is generated in such a manner starting from the MSB to the least significant bit taking m cycles to generate an m-bit output. Successive approximation converters that incorporate capacitor DACs are usually based on the charge redistribution principle. Figure 4 shows the block diagram of a charge redistribution implementation of the *successive approximation*(SA) [10] ADC. For this work, an SA ADC based on charge redistribution was implemented. The principle can be illustrated using Figure 4, where the DAC consists of binary-weighted capacitors $C_1 \cdots C_{n-1}$

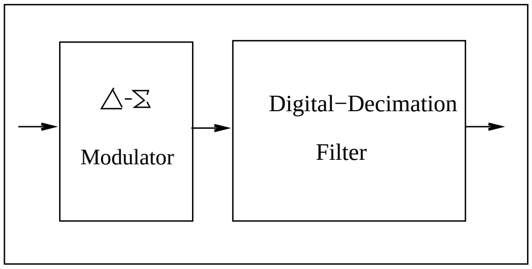

Fig. 5. Δ-Σ ADC

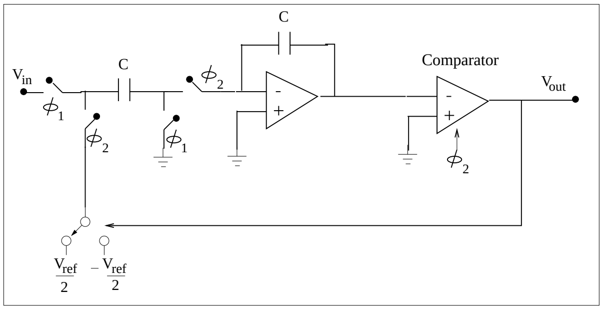

Fig. 6. Δ-Σ Modulator

($C_j = 2 \cdot C_{j-1}, j = 2, \cdots, n-1$ and $C_1 = C_0$). In the sampling mode the bottom-plate is grounded (in CL in Figure 4) and the input value is sampled onto the capacitors. In the hold mode the conversion proceeds by switching the bottom plate of some of the capacitors $C_1 ... C_{n-1}$ to V_{ref} according to a binary search algorithm, such that the top plate eventually becomes 0. The objective during the conversion is to drive to zero the difference between the DAC (convertlatch) output and the sampled input. One bit is converted in each cycle, starting with the most significant bit. A precise capacitor matching is required for this conversion. Current fabrication technologies cater to this requirement quite effectively.

D. Δ-Σ ADC

The Δ-Σ ADC falls under the category of oversampling converters which have become popular for high-resolution, medium-to-low speed applications such as high-quality digital audio. Figure 5 shows the block diagram of a Δ-Σ ADC. The Δ-Σ modulator is an analog component and the digital decimation filter is a digital component. The most common implementation of the Δ-Σ modulator (shown in Figure 6) 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, following the modulator, filters out this noise and provides a high resolution output. A digital low pass filter realization involves a multiplication of the serial bit pattern with coefficients which represent the *sinc* function. Since an ideal *sinc* function would need an infinite number of coefficients, practical cases implement a windowed *sinc* function. Figure 7 shows a block diagram of an 8-point digital-decimation filter. DFF is a delay element and $x(i)$ is the serial input at the ith time instance. The coefficients are symmetrical around the main lobe, and therefore, only four coefficients (h(0) to h(3)) are required.

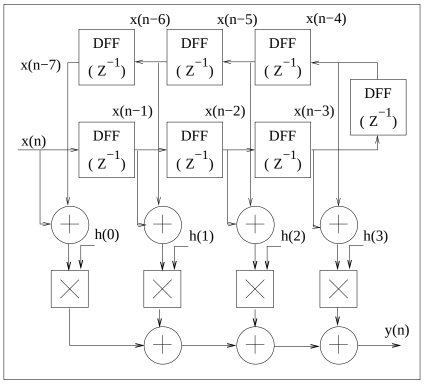

Fig. 7. A Digital Decimation Filter

III. FAULT SENSITIVITY ANALYSIS

There are different approaches to investigate the effects of transient faults. Hardware prototyping has been used [11] but 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 more accurate exhaustive simulation approach and used Hspice [12] for this purpose. The following section discusses the theoretical basis for the fault sensitivity analysis methodology used for this work. We present the transient fault model used for the analysis and the theoretical basis for the fault sensitivity analysis methodology used. We also illustrate the various kinds of analysis that can be performed with the data obtained from a fault simulation run.

A. Transient Fault Model

Several transient fault models have been proposed in [13], [14]. Since this work concentrates on α-particle induced transients, the double exponential α-particle transient model for the injection current proposed in [13] is used. The injection current due to an α-particle strike on a node, denoted by I_{inj}, is given by

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

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. The time constants depend on several process related factors, and in this work, the time constants given in [15] are used: $\tau_1 = 1.63 \times 10^{-10}$ sec and $\tau_2 = 0.5 \times 10^{-10}$ sec. I_0 can be calculated by

$$I_0 = \frac{Q_{inj}}{\tau_1 - \tau_2} \qquad (2)$$

where Q_{inj} is the charge injection level in Coulombs. Charge injection 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 [15]. Figure

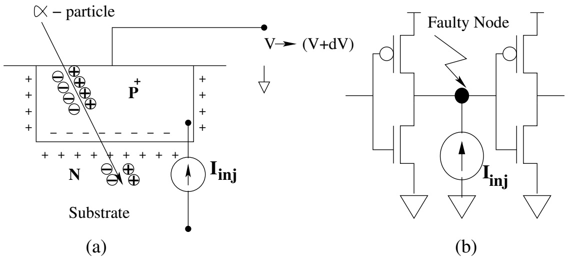

Fig. 8. (a) α-particle hit on the drain of a PMOS transistor (b) The α-particle hit modeled as a current source

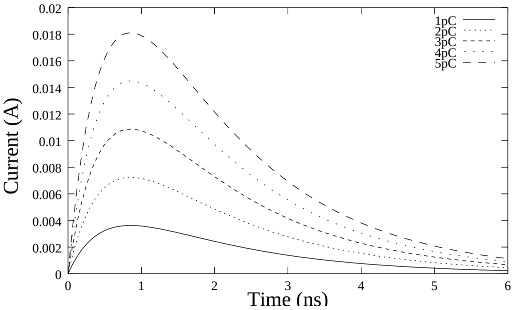

Fig. 9. Current pulse generated as a result of an α-particle strike

8(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 8(b) shows the current source equivalent model of the transient fault caused by an α-particle hit. Figure 9 shows the current pulses that are generated as a result of an α-particle hit for different values of the injection charge. Thus, an α-particle hit on a circuit node can be simulated by connecting a current source injecting a current pulse of I_{inj} at the said node.

B. Theoretical Basis

Traditionally, fault conditions in simulation strategies have been varied along three dimensions: space, time and injection level. It is important to consider varying the inputs to the circuit as well, 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. Since fault conditions have to be varied spatially, the physical design step (3) is the most suitable point to carry out the fault sensitivity analysis. However, the complexity of the design effort needed to create the layout emphasizes the need to move the analysis to an earlier stage. When moving the fault sensitivity analysis up in the design cycle we expect to reduce the number of time consuming design iterations, but should also expect to pay a penalty in terms of accuracy of the results.

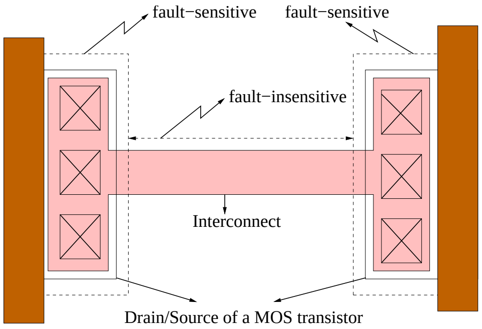

Fig. 10. The sensitive and insensitive parts of a node

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. We define the fault sensitivity of a block as the probability that an α-particle hitting the block will result in a circuit failure and we denote it by *POF* (Probability of Failure). For a given input voltage, *POF* is calculated as follows. An α-particle transient is injected into each node of the block and we denote the outcome of the experiment by E_i:

$$E_i = \begin{cases} 1 & \text{if the injection into node } i \text{ results in} \\ & \text{a failure} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

for $i = 1, 2, \cdots, n$ where n is the number of circuit nodes in a block. The *POF* is now defined as

$$POF = \frac{1}{n}\sum_{i=1}^{n} E_i \tag{4}$$

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 [16]. 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 10). 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 [4]. 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, assign to node i a weight, denoted by w_i, given by

$$w_i = \frac{A_{s,i}}{\sum_{i=1}^{n} A_{s,i}} \tag{5}$$

where $A_{s,i}$ is 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

$$POF = \sum_{i=1}^{n} w_i E_i \tag{6}$$

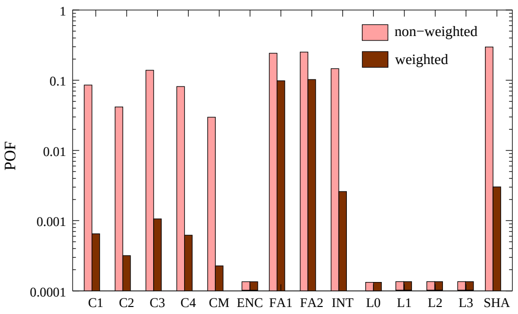

Fig. 11. Weighted vs Non-weighted (log scale) approach for the FI ADC (averaged over all inputs)

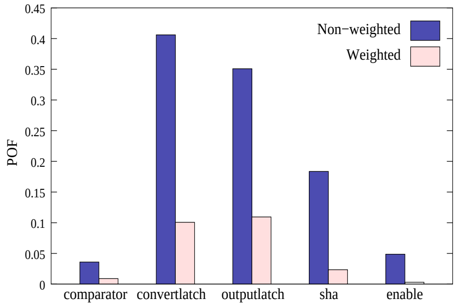

Fig. 12. Weighted vs Non-weighted approach for the SA ADC (averaged over all inputs)

Figures 11 and 12 show the block sensitivities for the FI and SA ADCs, respectively, 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 11 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. Even the more accurate sensitivity metric presented in (6) treats all faults uniformly. However, some faults may result in larger errors at the ADC output than other faults. Hence, another metric, namely, the relative error denoted by E_{rel}, is proposed. E_{rel} is given by

$$E_{rel} = \triangle V / V_{exp} \ , \qquad \triangle V = |V_{err} - V_{exp}| \tag{7}$$

where V_{exp} is the expected correct output and V_{err} is the erroneous output.

Based on the definition of relative error, a unified metric called the Average Relative Error (*ARE*) which includes the magnitude of error, is proposed. In contrast to the *POF* where all the errors are treated uniformly, *ARE* gives more weight to α-particle hits which cause larger relative errors at the ADC output. The sensitivity, characterized by the ARE, is given by

$$ARE = \sum_{i=1}^{n} w_i \bar{E}_{rel,i} \tag{8}$$

where $\bar{E}_{rel,i}$ is the average relative error due to an injection at node i and is calculated using

$$\bar{E}_{rel,i} = \frac{1}{k} \sum_{i=1}^{k} E_{rel,i} \ , \qquad k = p \cdot q \cdot r \tag{9}$$

Input	Analog	Digital	Full
Representative	0.02600	0.00150	0.02750
Complete	0.02465	0.00140	0.02610

TABLE I

Overall ARE comparison for Representative (1.25v, 1.85v, 2.35v) Vs Complete (1.05v to 2.55v in steps of 0.1v) inputs for the 4-bit Flash ADC

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. $E_{rel,i}$ is calculated using (7) and w_i is calculated using (5). The Maximum Relative Error (*MRE*) can be used as an additional metric which provides the worst case magnitude of the error. The choice of a metric for sensitivity analysis is based on the design objectives and not on the ADC architecture. Our earlier work in [17] has shown that the use of the *ARE* metric may sometimes mask sensitivity improvements. Therefore, the choice of a metric (POF, ARE) 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. Based on the requirements of the system, an appropriate metric can be chosen for the fault sensitivity analysis. 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 (6) or (8) to calculate the sensitivity of the constituent blocks.

In this work, the reduction of frequency of errors for the FI and SA ADCs, and reduction of relative size of errors for the *flash* and Δ-Σ ADCs were the assumed design objectives. Therefore, *POF* has been used as the sensitivity measure for the FI and SA ADCs, and ARE has been used for the *flash* and Δ-Σ ADCs.

C. ADC Sensitivity Analysis

Transient fault injection experiments were performed on 4-bit transistor level implementations of *successive approximation, folding and interpolating, flash* and Δ-Σ ADCs. The results obtained from the simulations have been used to grade the fault sensitivities of the blocks in the ADC. It is essential to vary the inputs in the sensitivity analysis as it can have a bearing on selecting *critical blocks* for redesign. However, performing an exhaustive simulation for all possible inputs is prohibitively expensive. Therefore, a scheme of selecting three representative inputs, one each in the lower, middle and upper input ranges and performing exhaustive simulation only for those inputs, was studied. This study was performed on the analog and digital blocks of a 4-bit flash ADC. The simulations were performed for eight injection levels and four time instances for each node. The *ARE* and *POF* obtained with the representative inputs and with the complete range of inputs (at a quantization step of 0.1v) were compared. Tables I and II show that the *ARE* and *POF* for both cases are reasonably close. Therefore, the simulations in the remaining parts of this work

Input	Analog	Digital	Full
Representative	0.05100	0.00500	0.05600
Complete	0.04750	0.00480	0.05200

TABLE II

OVERALL *POF* COMPARISON FOR REPRESENTATIVE VS COMPLETE INPUTS FOR THE 4-BIT FLASH ADC

Block	Input (V)			ARE	MRE
	1.25	1.85	2.35		
Analog	0.06100	0.01300	0.00530	0.02600	4.0
Digital	0.00260	0.00150	0.00052	0.00157	0.5

TABLE III

FLASH ADC BLOCK SENSITIVITY, q=8, r=4

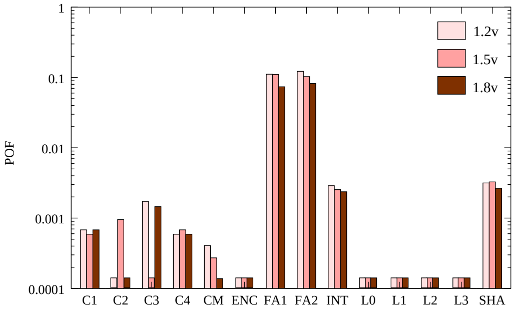

Fig. 13. Block sensitivity (log scale) variation with inputs for the FI ADC, q=14, r=4

have been performed for one representative input in the lower, middle and upper input ranges.

1) Flash ADC: The ARE metric (8) was used for evaluating the sensitivity of the blocks in a flash ADC. Table III shows that the analog block of the flash ADC which comprises of comparators is more sensitive than the digital block. It is also seen that the analog block remains more sensitive than the digital block for all the input subranges considered. Thus, for the flash ADC the analog block is identified as the *critical block*.

2) Folding and Interpolating ADC: The variations of the sensitivity of this ADC to changes in input voltage and level of injection were investigated. The *POF* metric (6) was used for evaluating the sensitivity of the individual blocks. The analog block in this ADC was further partitioned as it comprised of more components as opposed to the *flash* ADC wherein the analog block comprised of comparators only. Figure 13 shows the sensitivities of the blocks for the three input levels. This figure shows that the sensitivities of some blocks to α-particle hits vary from one input value to another (C2 and C3 in Figure 13). The comparators (C1 through C4) 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 1.42-1.52v for the comparator C2. Hence, it can be concluded that C2 is relatively more sensitive in these input ranges (as is shown in the bar graph corresponding to an input of 1.5v in Figure 13). Figure 14 shows that the sensitivities of blocks do not vary considerably if the injection level is more than 6 pico-Coulomb (pC). Thus, there is no need to repeat the sensitivity analysis for injections levels beyond 6pC.

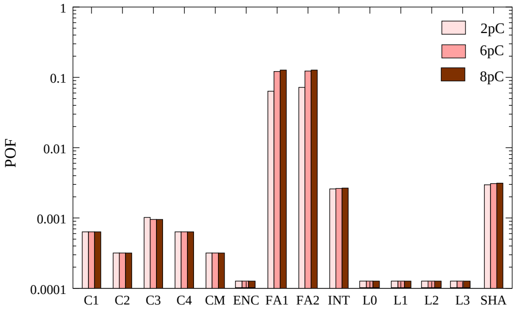

Fig. 14. Block sensitivity (log scale) variation with injection levels (FI), p=3, r=4

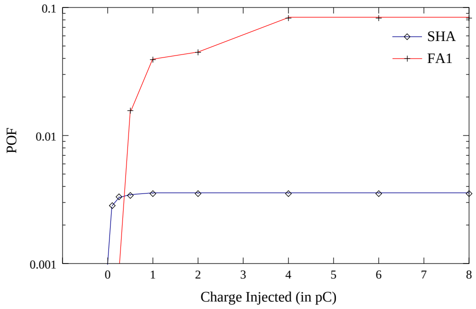

Fig. 15. Block sensitivities (log scale) variation with injections (FI), p=3, r=4

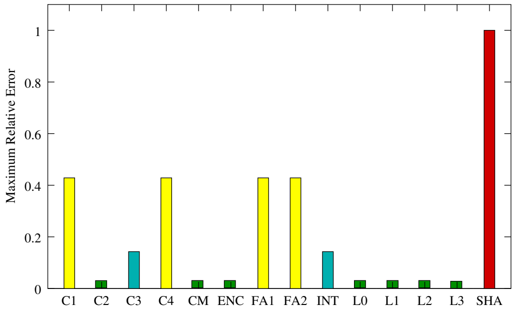

Fig. 16. Maximum relative error (FI), p=3, q=14, r=4

Figure 15 shows that for lower injection levels the ordering of *critical blocks* might change (SHA is more sensitive than FA1 from 0pC to 0.25pC). This figure also shows that beyond a certain injection level there is no further increase in block sensitivity. Figure 16 shows the maximum relative error due to each block. The results again 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 16). Thus, the SHA and FA blocks have been identified as *critical blocks* for the FI ADC.

3) Successive Approximation ADC: The *POF* metric (6) was used for evaluating the sensitivity of the blocks in the Successive Approximation (SA) ADC. Figure 17 shows the sensitivities of the blocks for the three representative inputs. It is evident from this figure that the sensitivities of some blocks to α-particle hits vary from one input value to another (outputlatch, convertlatch in Figure 17). Figure 18 shows that for lower injec-

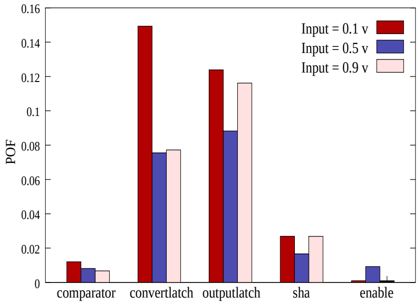

Fig. 17. Block sensitivity variation with inputs for the SA ADC, q=14, r=32

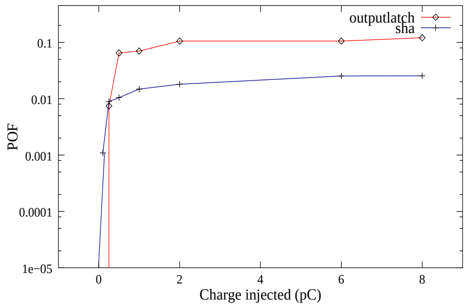

Fig. 18. Block sensitivities (log scale) variation with injections (SA), p=3, r=32

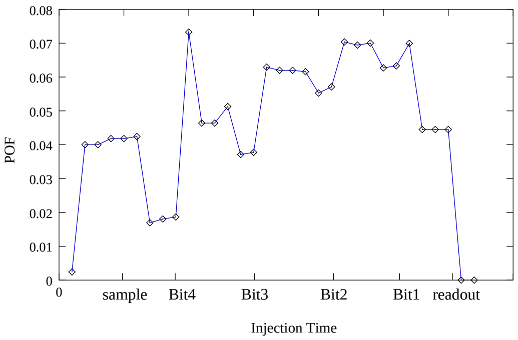

Fig. 19. Variation of *POF* with fault injection times (SA), p=3, q=14

tion levels the ordering of *critical blocks* might change (SHA is more sensitive than the outputlatch from 0pC to 0.25pC). This figure also shows that, as for the FI ADC, beyond a certain injection level there is no further increase in the sensitivity of the blocks in the SA ADC. Figure 19 shows the variation in sensitivity with 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 20 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 20). Another analysis performed on the convertlatch revealed that out of the four latches in the convertlatch, the latch containing the most significant bit is the most sensitive (Figure 21). The outputlatch, convertlatch and sha have been thus identified as *critical blocks* that should be redesigned to improve the reliability of the circuit.

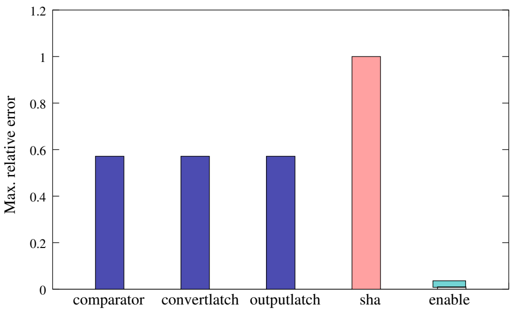

Fig. 20. Maximum relative error (SA), p=3, q=14, r=32

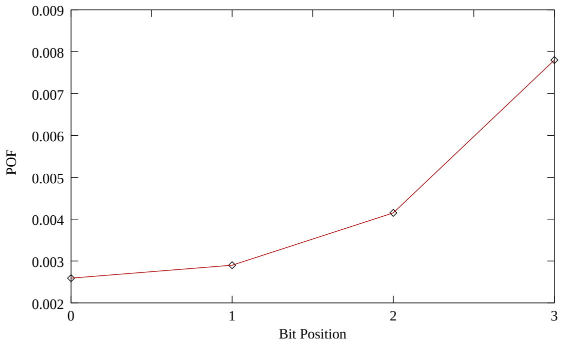

Fig. 21. Sensitivity of the four latches in convertlatch (SA), p=3, q=14, r=32

Block	Input (V)		ARE
	2	3	
Analog	0.00002	0.00006	0.00004
Digital	0.00426	0.00573	0.00500

TABLE IV

Δ-Σ ADC SENSITIVITY ANALYSIS, p=2, q=8, r=16

4) Δ-Σ ADC: The Δ-Σ ADC like the SA 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. For the 4-bit Δ-Σ ADC implemented for this work, the Δ-Σ modulator generates eight bits in 8 clock cycles. These eight bits are then digitally filtered and the final 4-bit ADC output is generated at a decimated frequency. The middle bits (bit 3 to 6) among the eight bits generated by the Δ-Σ modulator contribute more towards the final ADC output as the larger filter coefficients are multiplied by these bits. However, these bits are not necessarily the most sensitive ones. Therefore, it is of interest to analyze the number of errors in each of the eight bits resulting from injections in the Δ-Σ modulator. Figure 22 shows the variation of the number of errors with bits generated in the first to the eighth clock cycles. It is observed that the bits generated in the later cycles are more prone to faults. This can be attributed to the fact that the bit generated in the nth cycle is dependent on the offset stored in the integrator in the $(n-1)$th cycle. Therefore, the bit generated in the last cycle (in this case the 8th cycle) will be sensitive to faults injected in all preceding cycles in addition to those injected in the current cycle. A sensitivity analysis of

Design	Input (V)			ARE	MRE	Fault Sens. Area	Delay
	1.25	1.85	2.35				
Tabatabaei[18]	0.06100	0.01300	0.00530	0.02650	4	1795.46	1.10ns
Yee[19]	0.06500	0.01270	0.00450	0.02700	4	2712.58	0.57ns
Differential[8]	0.00630	0.00140	0.00036	0.00290	2	4933.11	0.90ns
Hester[20]	0.00660	0.00190	0.00065	0.00300	4	3325.11	0.47ns

TABLE V
SENSITIVITIES OF FOUR COMPARATORS WITH VARYING INPUTS FOR 4-BIT FLASH ADC, p=3, q=8, r=4

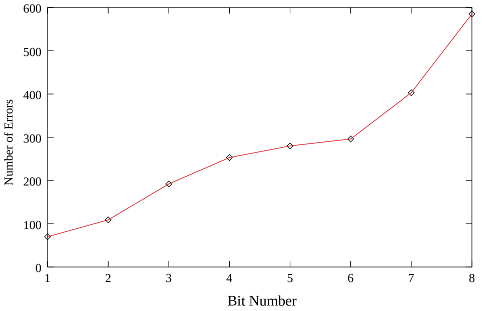

Fig. 22. Variation in number of errors with time for the Δ-Σ ADC, p=3, q=8, r=16

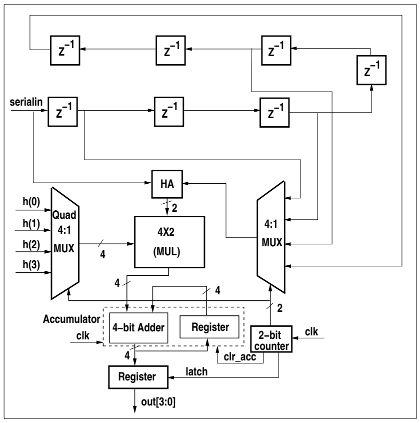

Fig. 23. Digital Decimation Filter in the Δ-Σ ADC

the Δ-Σ ADC revealed that the digital decimation filter (depicted in Figure 23) is the *critical block* (see Table IV). The results shown above illustrate the different kinds of analysis that can be performed to aid the designer in arriving at a more reliable implementation. 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.

IV. RELIABILITY IMPROVEMENT 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

Design	Input (V)			ARE
	1.25	1.85	2.35	
Tabatabaei[18]	0.06370	0.01469	0.00590	0.02810
Yee[19]	0.06719	0.01355	0.00480	0.02850
Differential[8]	0.00677	0.00162	0.00042	0.00323
Hester[20]	0.00735	0.00219	0.00073	0.00342

TABLE VI
SENSITIVITIES OF THE 4-BIT FLASH ADC WITH DIFFERENT COMPARATORS, p=3, q=8, r=4

It is essential to gauge the improvement that each of these techniques offers as this would help the designer to decide on an effective fault tolerance design strategy. The following sections describe several redesign techniques [21] and also illustrate the amount of sensitivity improvement that can be gained by employing them.

A. Alternative Robust Implementations

Most of the ADC building blocks like the sample and hold amplifier and comparators have several possible implementations which trade-off area, speed and susceptibility to noise and parametric variations. These implementations inherently have different sensitivities to α-particle transients. When deciding on an implementation for the ADC in question, the sensitivity of feasible implementations should be compared and an appropriate implementation should be chosen.

1) Flash ADC: Our sensitivity analysis of the 4-bit Flash ADC has identified the analog block primarily comprising of comparators as the *critical block*. Four comparators were then considered for sensitivity evaluation to identify the most robust implementation.

Tabatabaei's Comparator [18] is a recent implementation of a single stage comparator. Such single stage comparators provide the desired gain in most cases but their delay may be too high. To alleviate the problem of high delay, comparators with multiple preamplification stages have been proposed. One such multistage comparator implementation (Hester's comparator [20]) incorporates positive feedback to achieve the desired gain. Another comparator (Yee's comparator [19]) uses inverters biased in the high gain region as preamplifiers. The simplicity of this multistage comparator has made it quite popular in the resolution range from 8 to 10 bits. Table V shows the results of sensitivity analysis of the four alternative comparator implementations. The initial version of the 4-bit flash ADC (discussed

Design	Input (V)			ARE	MRE	Fault Sens. Area	Delay
	1.2	1.5	1.8				
Conventional[8]	0.3200	0.1210	0.0580	0.2120	1.8	20	6.60ns
McCreary[22]	0.3370	0.0970	0.0600	0.1700	1.6	25	7.80ns
Lim[23]	0.0110	0.0017	0.0019	0.0048	1.6	41869	4.22ns

TABLE VII
SENSITIVITIES OF SHAS WITH VARYING INPUTS FOR 4-BIT FI ADC, *p*=3, *q*=10, *r*=4

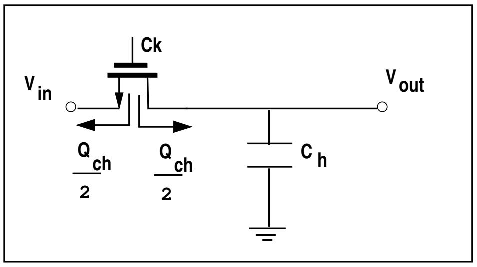

Fig. 24. Conventional Sample and Hold Amplifier

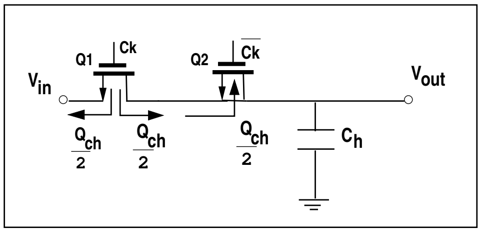

Fig. 25. Clock Feedthrough Cancellation SHA

in subsection II-A) incorporated the comparator proposed by Tabatabaei [18]. Either one of the above four implementations can be chosen based on the requirements of the application. We have found that the comparator proposed by Hester [20] and the differential one [8] are the least sensitive among the implementations considered. Sensitivity gains of as much as 89% were observed. The differential implementation also showed an improvement of 50% in the MRE. The improvement in sensitivity is achieved with a penalty in terms of area (see Table V). However, the less sensitive comparators have also a lower delay. A lesser area overhead is observed in Hester's implementation with almost comparable sensitivity improvement. Table VI shows the effect on the sensitivity of the whole ADC. Since the comparators are the *critical blocks* in the flash ADC, a similar sensitivity improvement is observed for the whole ADC.

2) Folding and Interpolating ADC: The sensitivity analysis of the 4-bit FI ADC has identified the sample and hold amplifier (SHA) as a *critical block*. Three implementations were considered to identify the most robust SHA. Figure 24 shows the conventional implementation [8] of the SHA. This implementation is susceptible to clock feedthrough which causes an extra charge of $Q_{ch}/2$ (Q_{ch} is the channel charge) on the hold capacitor (C_h) whenever Ck turns the sampling switch off. An implementation [22] which alleviates the problem of clock feedthrough (see Figure 25) uses a dummy switch ($Q2$) with $(W/L)_{Q2} = 0.5 \cdot (W/L)_{Q1}$. The dummy switch turns on whenever the sampling switch ($Q1$) turns off and absorbs the channel charge ($Q_{ch}/2$) released by $Q1$, leaving the hold capacitor unaffected.

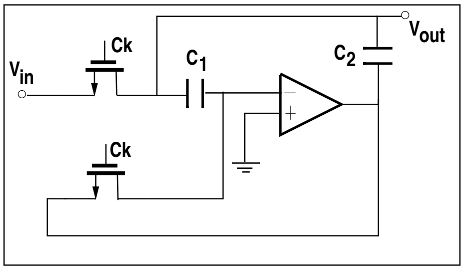

Fig. 26. Miller Hold Capacitor SHA

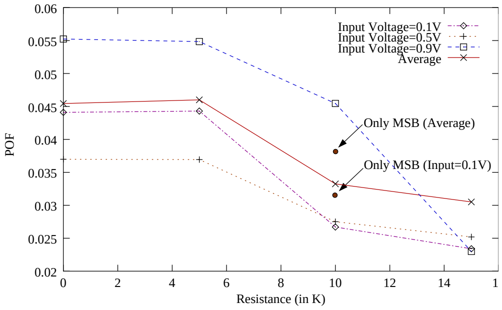

Fig. 27. Variation in sensitivity with increasing resistance (R), *p*=3, *q*=8, *r*=32

Since the holding capacitor has to hold the sampled value for some time, its value is usually large. However, a larger value of the holding capacitor also implies that the acquisition time for sampling the input will increase. An alternate implementation (Figure 26) [23] changes the configuration in the sample and hold modes so that the input sees a small capacitance in the sampling mode without sacrificing in terms of the hold time. The following expressions show the values of the acquisition (C_{acq}) and the hold (C_{hold}) capacitances in this configuration:

$$C_{acq} = C_1 + C_2 \ , \qquad C_{hold} = (1 + A) \cdot \frac{C_1 C_2}{C_1 + C_2} \tag{10}$$

where A is the gain of the operational amplifier.

Table VII shows the results of the sensitivity analysis of the above three possible implementations of the SHA. McCreary's [22] implementation shows a 19.8% sensitivity improvement over the conventional [8] implementation. In addition, it also shows an improvement of 11% in the MRE. It however, consumes more area and incurs a higher delay. Although Lim's [23] implementation shows a higher improvement in sensitivity, it consumes much more area than McCreary's implemen-

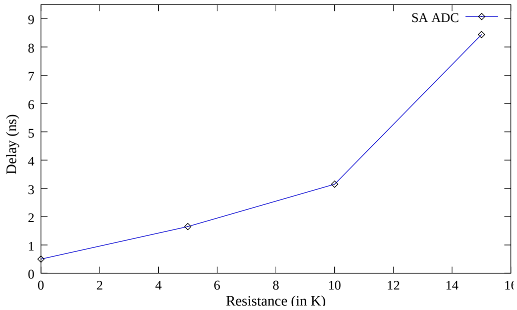

Fig. 28. Variation in performance with increasing resistance (R), p=3, q=8, r=32

Input(V)	NFT (ARE)	FT (ARE)	% impr
2	0.00426	0.00339	20.30
3	0.00573	0.00491	14.22
Avg.	0.00500	0.00394	21.10

TABLE VIII
DIGITAL DECIMATION FILTER SENSITIVITY OF THE NON FAULT TOLERANT (NFT) AND THE FAULT TOLERANT (FT) VERSIONS , p=2, q=8, r=4

Input(V)	NFT (ARE)	FT (ARE)	% impr
2	0.00428	0.00341	20.3
3	0.00579	0.00497	14.0
Avg.	0.00504	0.00398	21.0

TABLE IX
Δ-Σ ADC (NFT VS FT) SENSITIVITY, p=2, q=8, r=16

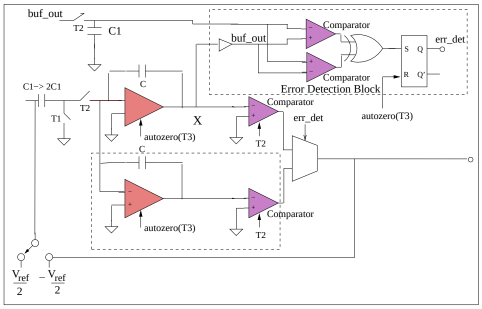

Fig. 29. Δ-Σ modulator with redundancy

Input(V)	NFT (ARE)	FT (ARE)	% impr
2	0.300	0.164	45.3
2.5	1.370	0.080	95.0
3	0.660	0.563	14.7
Avg.	0.790	0.270	65.8
MRE	0.667	0.500	25.0

TABLE X
SENSITIVITY($\times 10^{-4}$) AND MRE OF Δ-Σ MODULATOR, p=3, q=8, r=16

tation and hence may not be an effective replacement for the conventional implementation.

3) Successive Approximation ADC: The sensitivity analysis of the 4-bit SA ADC has identified the convertlatch and the outputlatch as *critical blocks*. The Transient Pulse Tolerant Latch (TPTL) proposed in [24] was considered for reliability improvement. The resistors in the 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 is incurred [17](Figure 27). The overhead can be reduced by replacing only the most sensitive latch in convertlatch by TPTL.

Figure 28 shows that the delay due to higher resistance increases exponentially. Therefore, the final resistance value should be chosen by taking the performance degradation into account.

4) Δ-Σ ADC: Since the digital decimation filter in the Δ-Σ ADC uses latches extensively, using the TPTL described in the previous subsection can lower the sensitivity of the decimation filter. We therefore, replaced all latches with TPTL and observed improvements in sensitivities of as much as 21% (Tables VIII and IX). This improvement is achieved however, at a cost of reduced performance. Figure 28 shows the performance degradation due to introduction of the resistance (R).

B. Adding Redundancy

Whereas the previous technique tends towards *fault resilience*, this technique attempts to mask the effect of a fault. One of the ways fault tolerance can be achieved through redundancy is to first detect the fault and then recover from the fault. This involves duplication of the block, and design of an error detection scheme which can activate the redundant block when a fault is detected. This technique has been implemented for the modulator in the Δ-Σ ADC. The Δ-Σ modulator is an ideal candidate for applying this technique because the technique addresses the integrator which is auto-zeroed on the decimated clock ($T3$ in Figure 29). If the error is not corrected it will effect the subsequent serial bit stream generated and it will generate erroneous bits till the next time the integrator is auto-zeroed. Though this technique can be used for other ADCs, it will have the maximum impact on ADCs like the Δ-Σ, part of which retains some information from the previous cycle (like the integrator).

While the input is being sampled onto the sampling capacitor, the rest of the nodes in the ADC are maintained at the value evaluated in the previous cycle. This characteristic can be used to detect an error and protect the circuit from faults injected during the sampling time. Since recent Δ-Σ ADC implementations show that almost 50% [18] of the cycle time is spent in sampling, this scheme would address a sizeable number of faults. This fact further fortifies the argument that the proposed technique is better suited to the Δ-Σ ADC as compared to other ADCs. Figure 29 shows the modified first-order Δ-Σ modulator with the redundancy incorporated in it. The capacitor C1

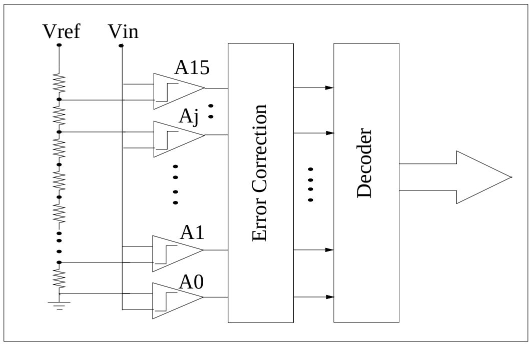

Fig. 30. Fault Tolerant 4-bit Flash ADC

stores a copy of the value in the integrator. When the input is being sampled (T1 is high and T2 is low), the integrator output (marked by X in Figure 29) should not change. In the event that a fault causes it to change, the error detection block flags an error which activates the redundant block (when T2 goes high). Table X shows the result of the sensitivity analysis run on the Non-Fault Tolerant (NFT) and the Fault Tolerant (FT) versions of the Δ-Σ Modulator. The results show a 65.8% improvement in sensitivity and 25% improvement in *MRE* with approximately 75% area overhead.

C. Pattern Detection

In some ADCs (e.g., Flash and FI) the signal lines at the boundary between the analog and the digital blocks exhibit a specific pattern. If the expected pattern is not detected, either a flag can be asserted or, if possible, a correction can be attempted. This technique has been used for improving the reliability of a 4-bit flash ADC. The output of the comparators in the flash ADC exhibit a thermometer code pattern. Therefore, a 0 detected within a string of 1s or vice-versa, indicates an error. This error can be corrected by selecting the majority value from within a neighborhood of x bits on either side of the bit to be corrected, where $x \geq 1$. For our implementation, x was taken as 1. Figure 30 shows the modified block diagram of the fault tolerant 4-bit flash ADC. The Error Correction block contains three types of error correcting subblocks, two for the boundary signals (A_{15} and A_0) and one for the rest of the signals. The following Boolean expressions illustrate the logic used for the correction.

$$\tilde{A_{15}} = A_{15} \cdot (A_{13} + A_{14}) \tag{11}$$

$$\tilde{A_0} = A_0 + (A_1 \cdot A_2) \tag{12}$$

$$\tilde{A_j} = A_j \cdot A_{j-1} + A_j \cdot A_{j+1} + A_{j+1} \cdot A_{j-1} \tag{13}$$

Table XI shows the results of the sensitivity analysis on the flash ADC. It was observed earlier (Subsection III-C.1) that the analog portion of the ADC which is primarily comprised of comparators was more sensitive (Table III) than the digital part. Since this technique addresses the errors due to faults injected in the analog block, it will provide considerable overall sensitivity improvement. An improvement of around 67.8% in sensitivity at the cost of 55% area overhead was observed.

Input(V)	NFT (ARE)	FT (ARE)	% impr
1.25	0.0637	0.0185	70.9
1.85	0.0146	0.0076	47.9
2.35	0.0059	0.0019	67.7
Avg.	0.0280	0.0090	67.8

TABLE XI

FLASH ADC (NFT VS FT) SENSITIVITY, p=3, q=8, r=4

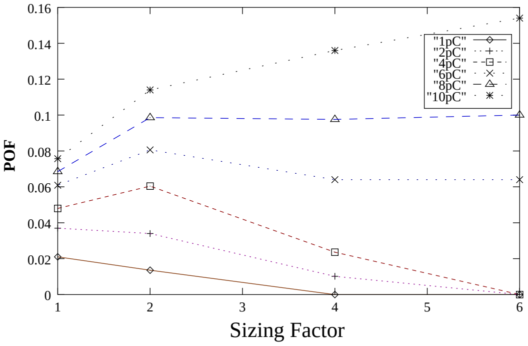

Fig. 31. Sensitivity variation with sizing and injection levels, q=14, r=4

D. Transistor Sizing

An α-particle injection results in a current spike at the faulty node. This current translates to a voltage fluctuation whose magnitude depends on the driving strength of the transistor, the capacitance at the node and the injection current [24]. One of the primary factors influencing the magnitude of the fluctuation is the resistance posed by the transistors connected to that node. Therefore, one would expect an improvement in the reliability by sizing up the transistor and thus reducing the resistance. On the other hand, sizing up a transistor also increases its *fault-sensitive* area. The benefits of transistor sizing were analyzed for both digital and analog circuits in [17] and are further discussed in the following subsections.

1) Digital Circuits: This technique was implemented in 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. Figure 31 shows that the sensitivity increases and then decreases with sizing for bounded injections. For injection levels bounded by 1pC an improvement in reliability of 33% is observed when sizing the circuit by twice its original size. Furthermore, the maximum value of the *POF* for a higher injection bound occurs at a higher sizing ratio (1 for 1pC and 2 for 4pC in Figure 31). For the above simulations the whole circuit was sized. The benefits of selective node sizing on the sensitivity of the 2-bit counter were also studied and are discussed next. The 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. Figure 32 shows that for lower injection level bounds ($1pC$) sensitivity improvement of as much as 60% is observed with an

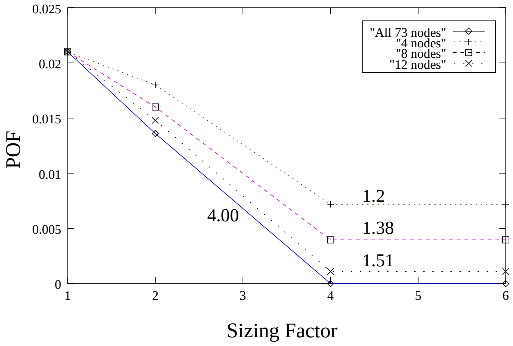

Fig. 32. Sensitivity variation with selective node resizing with injection level bounded by 1pC (the number on the curves indicate the corresponding *fault-sensitive* area increase factor) *q*=14,*r*=4

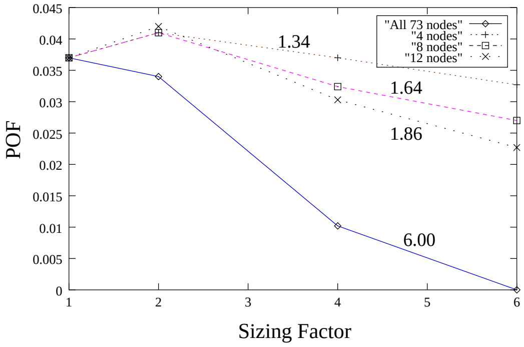

Fig. 33. Sensitivity variation with selective node resizing (injection level bounded by 2pC) *q*=14,*r*=4

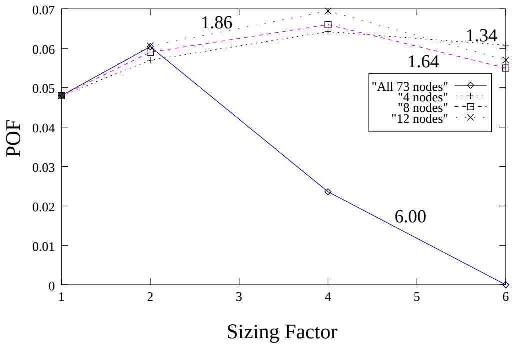

Fig. 34. Sensitivity variation with selective node resizing (injection level bounded by 4pC) *q*=14,*r*=4

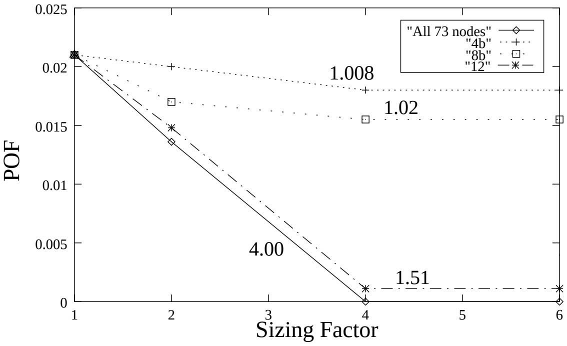

Fig. 35. Sensitivity variation with selective node resizing (injection level bounded by 1pC) *q*=14,*r*=4

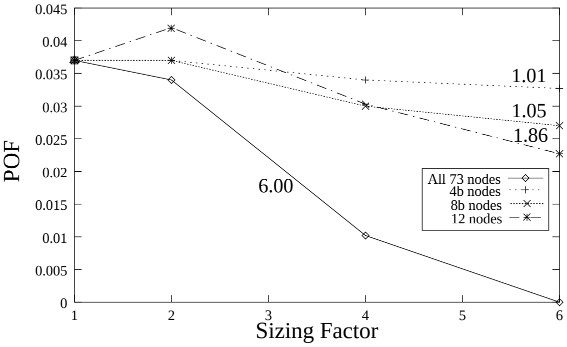

Fig. 36. Sensitivity variation with selective node resizing (injection level bounded by 2pC) *q*=14, *r*=4

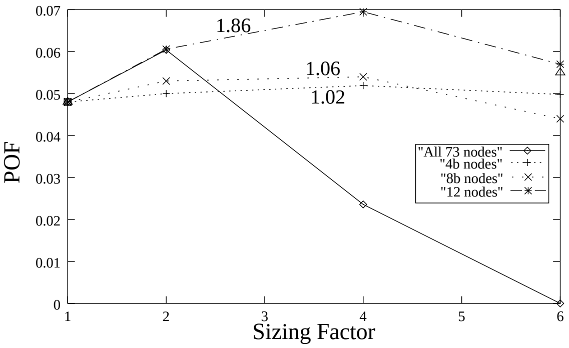

Fig. 37. Sensitivity variation with selective node resizing (injection level bounded by 4pC) *q*=14,*r*=4

area overhead of only 20%. The numbers on the curves indicate the area increase factor for the sizing factor which results in the lowest sensitivity. The above node selection scheme works well for injection levels bounded by 1pC. However, the improvement is not so sizeable for higher injection level bounds (see Figures 33 and 34). This motivates a search for a better node selection scheme and an insight into why the improvement is limited 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, which is equal to $\Delta V = I_{inj}R_{on}$, 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 achieved. It is very likely that this kind of nodes will show up at the top of the sorted list of nodes. Thus, in such cases it is possible to achieve higher sensitivity gains with a smaller area overhead by opting for an alternate scheme. In this scheme only the m nodes at the bottom 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 35, 36 and 37 show that for higher injection bounds, a sizeable improvement in sensitivity can be attained by opting for the alternate scheme (see Figure 37, the 4b and 8b curves show the sensitivity variation when m is 4 and 8, respectively, for n=12). In summary, for lower injection levels ($\leq 1pC$) resizing the n most sensi-

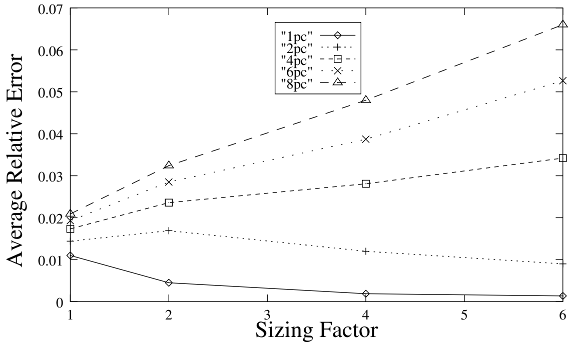

Fig. 38. Analog Block *ARE* variation with sizing and injection levels, *p*=3, *q*=14, *r*=4

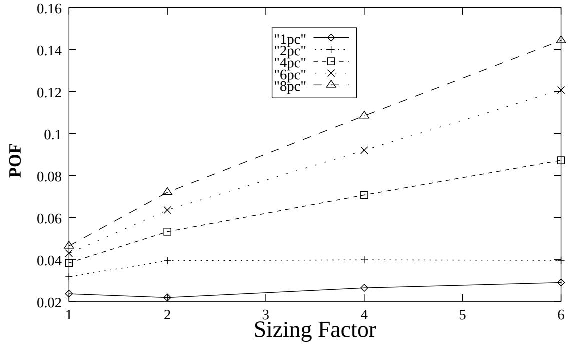

Fig. 39. Analog Block *POF* variation with sizing and injection levels, *p*=3, *q*=14, *r*=4

tive nodes proved beneficial while for higher injection levels ($\geq 2pC$) resizing the least sensitive m nodes in the sorted list of n proved beneficial. 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.

2) Analog Circuits: This technique was implemented for the comparators which were identified as the *critical blocks* in the 4-bit Flash ADC. Figure 38 shows that an improvement of around 50% in *ARE* can be achieved by sizing for injection levels bounded by 1pC. However, the *ARE* increases with sizing for injection levels above 4pC. An interesting point to note here is that the *POF* metric (see Figure 39) indicates that the sensitivity does not change by much for a sizing factor of 2 for an injection bounded by 1pC. This implies that the number of injected faults translating to errors still remains about the same but the magnitude of error due to each of those faults has reduced, thus resulting in an overall reduction of 50% in the *ARE* (see Figure 38). Further reduction in sensitivity for a higher sizing ratio is smaller because increasing the width of the transistor in an analog circuit entails increasing its length also, since the W/L ratios must be maintained. This implies that the resistance posed by the transistor will not change, but the capacitance seen by the node will increase, thus causing a reduction in the sensitivity in some cases as shown in Figure 38.

Therefore, sensitivity gains can be attained for lower injection levels (≤2pC) but as the injection level bound increases this technique does not prove beneficial.

V. Conclusions

A generic methodology for the reliability enhancement of ADCs has been presented. Fault sensitivity analysis followed by circuit redesign was identified as the fault tolerance strategy to be applied. The use of *node weights*, specific to α-particle transients, was proposed to increase the accuracy of the sensitivity analysis. Two metrics, namely the POF and ARE which characterize the sensitivity of a block, were presented. The following steps have been identified for α-particle induced fault sensitivity analysis:

(i) Calculate *weights* of the nodes.
(ii) Perform transient fault simulations on all nodes.
(iii) Use equation (6) or (8) to calculate the sensitivity of the constituent blocks.

This methodology was used to first identify *critical blocks* in the FI, SA, flash and Δ-Σ ADCs and then increase their reliability by circuit redesign. Several redesign techniques were presented including the selection of more robust implementations, adding redundancy, pattern detection and transistor sizing, using which, sensitivity gains of as much as 89%, 65.8%, 67.8% and 60%, respectively, were observed. Each of the proposed circuit redesign techniques can be used for other ADCs. For example, the SHA and comparator are some of the most ubiquitous blocks in ADCs and numerous circuit implementations have been proposed. Thus, several alternative robust implementations of these can be evaluated for most ADCs. The proposed redundancy technique will be useful in circuits where the node voltages have to be held to a constant value for a substantial amount of time (like in the Δ-Σ modulator). Pattern detection can be used in ADCs where the signal lines at the boundary between the analog and digital blocks are limited to certain patterns (like the flash and FI ADCs). Lastly, transistor sizing was found to be beneficial for both digital and analog circuits under certain circumstances.

VI. Future Work

ADC architectures like the pipelined, multi-step and integrating ADCs can be similarly analyzed for their sensitivities to α-particle transients. It is also necessary to study the impact of parametric variations in mixed-signal circuits on the sensitivity to α-particle transients. Currently, this work assumes that the circuit under test is properly centered in the process envelope. Initial simulations by varying the width and the threshold voltage of the transistors in a comparator have shown that sometimes an “uncentered” design can have a lower rather than higher sensitivity to α-particle transients. In most of the cases however, the variation was small for the type of parametric variations considered. All in all, there is a need to study the sensitivity of so called “uncentered” designs to α-particle induced transients. Finally, this work can be extended for other types of transient faults by developing appropriate models for the candidate faults.

References

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

[2] R. Iyer and D. Rosetti, "A statistical load dependency of CPU errors at SLAC," in *Proc. FTCS-12*, 1982, pp. 363–372.

[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, pp. 258–264, 1992.

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

[5] A. Antola, R. Negrini, M. Sami and N. Scarabottolo, "Tolerance to transient faults in microprogrammed control units," *IEEE Transactions on Reliability*, vol. 27, pp. 258–264, 1992.

[6] J. F. Ziegler , "Terrestrial Cosmic Rays," *IBM Journal of Research and Development*, vol. 40, pp. 19–40, 1996.

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

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

[9] R. E. J. Van De Grift, I. W. J. M. Rutten and M. Van Der Veen, "An 8-bit video ADC incorporating folding and interpolating techniques," *IEEE Journal of Solid-State Circuits*, pp. 944–953, 1987.

[10] 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, pp. 1759–1763, 1997.

[11] H. Amer, M. Cortes and E. McCluskey, "Robust dynamic recovery mechanisms," in *Proceedings of IEEE International Conference on Computer Design*, Oct 1987, pp. 310–313.

[12] Avant! Corporation, *Star-Hspice manual*, Dec 1999.

[13] G. C. Messenger, "Collection of charge on junction nodes from ion tracks," *IEEE Transactions on Nuclear Science*, pp. 2024–2031, 1982.

[14] G. Laguna and R. Treece, "VLSI modeling and design for radiation environments," in *Proceedings of IEEE International Conference on Computer Design*, Oct 1986, pp. 380–384.

[15] 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, pp. 1248–1256, 1996.

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

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

[18] A. Tabatabaei and B. A. Wooley, "A Two-Path Bandpass Sigma-Delta Modulator with Extended Noise Shaping," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1799–1809, 2000.

[19] Y. S. Yee, L. M. Terman and L. G. Heller , "A 1 mV MOS Comparator," *IEEE Journal of Solid-State Circuits*, vol. 13, pp. 294–297, 1978.

[20] R. K. Hester et al., "Fully Differential ADC with Rail-to-Rail Common Mode Range and Nonlinear Capacitor Compensation," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 173–183, 1990.

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

[22] J. McCreary and P. R. Gray, "All-MOS Charge Redistribution Analog-to-Digital Conversion techniques-Part I," *IEEE Journal of Solid State Circuits*, pp. 371–379, 1975.

[23] P. Lim and B. Wooley, "A High Speed Sample-and-Hold Technique Using a Miller Holding Capacitor," *IEEE Journal of Solid State Circuits*, vol. 26, pp. 643–651, 1991.

[24] H. Cha and J. H. Patel, "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.

PLACE
PHOTO
HERE

Mandeep Singh (M'01) received the B.E(Hons.) degree in electrical engineering from the Birla Institute of Technology and Sciences, Rajasthan, India in 1997 and the M.S. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 2001. While at the University of Massachusetts, he completed his thesis on the Reliability Enhancement of Analog-to-Digital Converters. He worked for Wipro Global Research and Development, Bangalore, India from 1997 to 1999, where he was involved in the design and development of high speed communication ASICs. He is currently working for Advanced Micro Devices in the Computation Products Group, Austin, TX, where he is involved in high performance microprocessor design.

PLACE
PHOTO
HERE

Israel Koren (S'72 - M'76 - SM'87 - F'91) received the B.Sc., M.Sc. and D.Sc. degrees from the Technion - Israel Institute of Technology, Haifa, in 1967, 1970, and 1975, respectively, all in Electrical Engineering. He is currently a Professor of Electrical and Computer Engineering at the University of Massachusetts, Amherst. Previously he was with the Technion - Israel Institute of Technology. He also held visiting positions with the University of California at Berkeley, University of Southern California, Los Angeles and University of California, Santa Barbara. He has been a consultant to several companies including IBM, Intel, Analog Devices, AMD, Digital Equipment Corp., National Semiconductor and Tolerant Systems.

Dr. Koren's current research interests include Techniques for Yield and Reliability Enhancement, Fault-Tolerant Architectures, Real-time systems and Computer Arithmetic. He published extensively in several IEEE Transactions and has over 150 publications in refereed journals and conferences. He currently serves on the Editorial Board of the IEEE Transactions on VLSI Systems. He was a Co-Guest Editor for the IEEE Transactions on Computers, special issue on High Yield VLSI Systems, April 1989 and the special issue on Computer Arithmetic, July 2000, and served on the Editorial Board of these Transactions between 1992 and 1997. He also served as General Chair, Program Chair and Program Committee member for numerous conferences. He has edited and co-authored the book, *Defect and Fault-Tolerance in VLSI Systems*, Vol. 1, Plenum, 1989. He is the author of the textbook *Computer Arithmetic Algorithms, 2nd Edition*, A.K. Peters, Ltd., 2002.