Reliability Enhancement of Analog-to-Digital Converters (ADCs) *

Mandeep Singh and Israel Koren

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

Abstract

Reliability of systems used in space, avionic and biomedical applications is highly critical. Such systems consist of an analog front-end to collect data, an ADC to convert the collected data to digital form and a digital unit to process it. The reliability of these systems is affected by the ability of its constituent blocks to tolerate faults. Therefore, it is necessary to increase the reliability of ADCs to ensure a highly reliable critical system. This paper illustrates the steps involved in the reliability enhancement of ADCs by first proposing a methodology for fault sensitivity analysis and then illustrating redesign techniques to improve the reliability of the highly sensitive(to faults) blocks.

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 the redesign to gauge the benefits of the redesign. Fault sensitivity analysis involves injection of faults either in the actual hardware or through software 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 efforts.

The reliability of a system is determined by the fault tolerance of its constituent blocks. Most of the 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: (1) Grading blocks of the circuit based on their sensitivities to transients and identifying *critical blocks*. (2) Increasing the fault tolerance of the identified *critical blocks*. This 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% [3] or more of computer system failures are known to be caused by transient faults 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

* Supported in part by NSF under contract MIP-9710130 and by JPL under contract 961294.

of radioactive elements present in the packaging material or solder [4]. Extraterrestrial cosmic rays which bombard earth continously are another source of α-particle radiation. Thus the applicability of this work is not restricted to systems in outer space but also to ground-based critical systems.

This paper is organized as follows. Section 2 provides a functional description of the ADCs addressed in this work. Section 3 describes the sensitivity analysis methodology used. In Section 4, the process of increasing reliability by opting for robust implementations and by introducing redundancy is discussed. Finally, Section 5 summarizes the findings of this 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. 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 [5]. High-accuracy architectures include *successive approximation*, *delta-sigma* and *integrating* ADCs [5]. 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.

2.1: Flash ADC

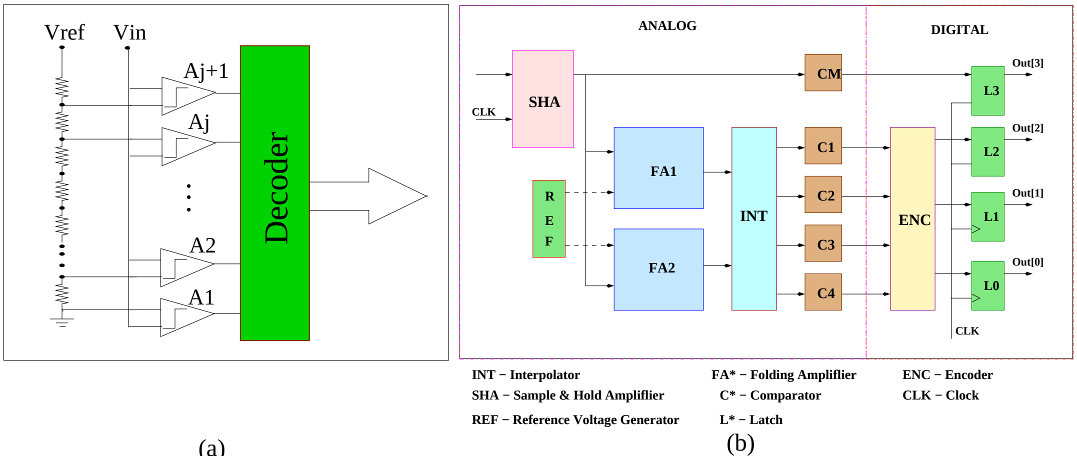


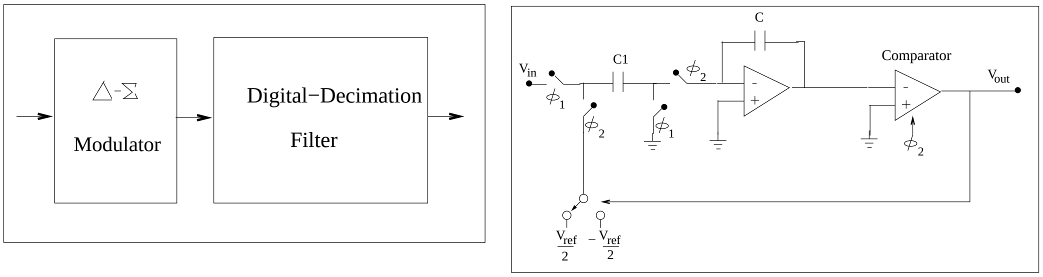
Figure 1. (a) Flash ADC (b) 4-bit Folding & Interpolating ADC.

This architecture is conceptually the simplest and potentially the fastest. It employs parallelism and distributed sampling to achieve high conversion speeds. Figure 1(a) 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 a m-bit binary output by the decoder.

2.2: Folding and Interpolating ADC

The problem of enormous input capacitance posed by the comparators at the input of flash ADCs led to the advent of folding and interpolating (FI) ADCs [5]. FI ADCs have folding amplifiers (FA1 and FA2 in Figure 1(b)) that fold the information represented by the reference voltages which characterize the quantization levels. Figure 1(b) shows the block diagram of a 4-bit folding and interpolating ADC. The sample and hold amplifier (SHA) samples the input and the sampled input is fed to two folding amplifiers (FA1 and FA2) which compares the input with the folded references and a comparator (CM) which generates the most significant bit. The interpolating block (INT) interpolates between the folding amplifier outputs. The INT block output is fed to the encoder (ENC) which generates the three least significant bits of the final digital output.

2.3: Δ-Σ ADC



(a) (b)

Figure 2. (a) Δ-Σ converter (b) First-order Δ-Σ Modulator.

Figure 2(a) 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 (shown in Figure 2(b)) 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.

3: Fault Sensitivity Analysis

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 an ideal point to address fault sensitivities. However, the complexity of the layout level database and the design effort needed to create the layout emphasize the need to move the analysis to an earlier stage. As we go up in the design cycle we should expect to pay a penalty in terms of the accuracy of the results, but time consuming design iterations can be avoided.

Traditionally, fault conditions in the sensitivity studies have been varied along three dimensions: space, time and injection level. It is important to also consider varying the inputs to the circuit, since this can have a bearing on selecting *critical blocks* for redesign. This is due to the fact that a block identified as a *critical block* for one input may not be as sensitive for another input. Hence, *critical blocks* should be identified based on the distribution of the input values. The circuit should be optimized for input values which are the most probable. The parameters that were varied for the fault injection experiments in this work include the fault injection time, the injection level, the node at which the α-particle strikes and the input voltage.

Our recent work [6] shows that the fault sensitivity analysis for an α-particle induced transient can be performed at an early stage in the design cycle of analog and mixed-signal circuits. This method provides an accurate analysis at an early stage in the design cycle. The double exponential α-particle transient model proposed in [7] has been used for our fault simulations. The injection current, I_{inj}, due to an α-particle strike is given by

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

where I_0 is the maximum current, τ_1 is the collection time constant for a junction and τ_2 is the ion track establishment time constant. The time constants depend on several process related factors, and in this work, the time constants given in [8] are used: $\tau_1 = 1.64 \times 10^{-10}$ sec and $\tau_2 = 0.5 \times 10^{-10}$ sec. An α-particle strike may result in an erroneous output. We define the relative error, denoted by E_{rel}, as

$$E_{rel} = \triangle V / V_{exp} \ , \qquad \triangle V = |V_{err} - V_{exp}| \tag{2}$$

where V_{exp} is the expected correct output and V_{err} is the erroneous output. The Maximum Relative Error (MRE) can also be used to get an idea about the worst case relative error. Since E_{rel} varies from one circuit node (hit by an α-particle) to the other, we define the Average Relative Error (ARE), as

$$ARE = \sum_{i=1}^{m} w_i E_{rel,i} \ \ , \ w_i = \frac{A_{s,i}}{\sum_{i=1}^{n} A_{s,i}} \tag{3}$$

where $A_{s,i}$ is the area of the fault-sensitive portion [6] of node i, w_i is the weight associated with node i [6] and m, n are the number of nodes in the block and the ADC, respectively. The sizes of the transistors in the schematics can serve as a good estimate for $A_{s,i}$. $E_{rel,i}$ is the relative error averaged over all injections (at different time instances and different levels of injection) at node i where the relative error due to each injection is calculated using (2). The metric says that an α-particle strike on a particular (ADC/block) will lead to an average relative error of ARE units. Therefore, the steps involved in the fault sensitivity analysis of an ADC are:

1. Calculate *weights* of the nodes (w_i).
2. Perform transient fault simulations on all nodes.
3. Use equation (3) to calculate the sensitivity i.e., the ARE, of the constituent blocks.

Note that for a comprehensive evaluation of fault susceptibility it is necessary to perform a full transient simulation of the system in the presence of transient faults.

4: Reliability Enhancement Techniques

A sensitivity analysis identifies *critical* blocks that the designer can concentrate on to improve the reliability of the system. Reliability 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) Incorporating fault tolerance into 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 an effective fault tolerance design strategy.

4.1: 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 making a choice for an implementation for the ADC in question, the sensitivity of the feasible implementations should be compared and an appropriate implementation must be chosen. A sensitivity analysis of the 4-bit FI ADC identified the sample and hold amplifier (SHA) as a *critical block*. Table 1 shows the results of sensitivity analysis of three possible implementations of the SHA. McCreary's[9] implementation shows a 19.8% reduction in ARE over the conventional [5] implementation. In addition, it also shows a reduction of 11% in the MRE. Though Lim's [10] implementation shows a higher improvement in sensitivity it consumes much more area than McCreary's implementation and hence may not be an effective replacement for the conventional implementation. Table 2 shows the results of the sensitivity analysis on alternative comparator implementations. The initial version of the 4-bit flash ADC incorporated the comparator proposed by Tabatabaei [11]. It was found that the comparator proposed by Hester [13] and the differential [5] were the least sensitive among the impelementations considered. Sensitivity gains of as much as 89% were observed. The differential impelementation also showed a reduction of 50% in the MRE.

Design	Input (V)			ARE	MRE
	1.2	1.5	1.8		
Conventional[5]	0.320	0.121	0.058	0.212	1.8
McCreary[9]	0.337	0.097	0.060	0.170	1.6
Lim[10]	0.011	0.0017	0.0019	0.0048	1.6

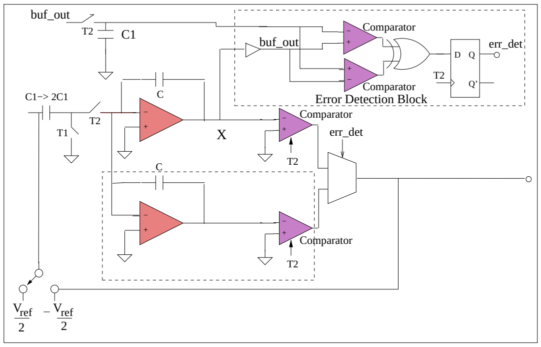
Table 1. Sensitivities of SHAs with varying inputs for 4-bit FI ADC

Design	Input (V)			ARE	MRE
	1.25	1.85	2.35		
Tabatabaei[11]	0.061	0.013	0.0053	0.0265	4
Yee[12]	0.065	0.0127	0.0045	0.027	4
Diff.[5]	0.0063	0.0014	0.00036	0.0029	2
Hester[13]	0.0066	0.0019	0.00065	0.0030	4

Table 2. Sensitivities of comparators with varying inputs for 4-bit Flash ADC

4.2: Adding Redundancy

Whereas the previous technique emphasizes *fault resilience*, this technique attempts to mask the effect of a fault. One of the ways fault tolerance can be achieved by adding redundancy is to first detect the fault and then recover from it. 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 Δ-Σ modulator in a Δ-Σ ADC. 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% [11] of the cycle time is spent in sampling, this scheme would address a sizeable number of faults. Figure 3(a) shows the modified first-order Δ-Σ modulator with the redundancy incorporated in it. The capacitor C1 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 3(a)) 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). Figure 3(b) shows the result of the sensitivity analysis run on the Non-Fault Tolerant (NFT) and the Fault Tolerant (FT) version of the Δ-Σ Modulator. The results show a 65.8% reduction in ARE and 25% reduction in MRE with approximately 75% area overhead. In our experiments, α-particles were injected in different time instances during the cycle and consequently, the calculated ARE reflects the fact that only during



(a)

Input(V)	NFT (ARE)	FT (ARE)	% imp
2	0.3	0.164	45.3
2.5	1.37	0.08	95
3	0.66	0.563	14.7
Avg.	0.79	0.27	65.8
MRE	0.667	0.5	25

(b)

Figure 3. (a) Δ-Σ modulator with redundancy (b) Sensitivity ($\times 10^{-4}$)of Δ-Σ modulator

about 50% of the cycle the added redundancy is beneficial.

4.3: Pattern Detection

In some ADCs (Flash, FI) the signal lines at the boundary between the analog and the digital blocks are limited to certain patterns. If the expected pattern is not detected either a flag can be asserted or if possible, 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 between a string of 1s or vice-versa, indicates an error. This error can be corrected by selecting the majority value from within a neighbourhood of x bits on either side of the bit to be corrected, where $x \geq 1$. For our implementation x was taken as 1. Table 3(b) shows the results of the sensitivity analysis of the flash ADC. It was observed that the analog portion of the ADC, which is primarily comprised of comparators, was more sensitive (Table 3(a)) than the digital part. An improvement of around 67.8% in sensitivity at the cost of 55% area overhead, was observed.

Block	NFT		FT	
	ARE	MRE	ARE	MRE
Analog	0.026	4	0.0057	1
Digital	0.00157	0.5	0.0036	4

(a)

Input(V)	NFT (ARE)	FT (ARE)	% imp
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.	0.028	0.009	67.8

(b)

Table 3. (a) Flash ADC block sensitivity (b) Flash ADC (NFT Vs FT) sensitivity

4.4: 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 [14]. One of the primary factors influencing the magnitude of 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. However, sizing up also increases the fault sensitive area. Therefore, sizing up need not always lead to reliability gains. This technique was implemented in a 2-bit counter used

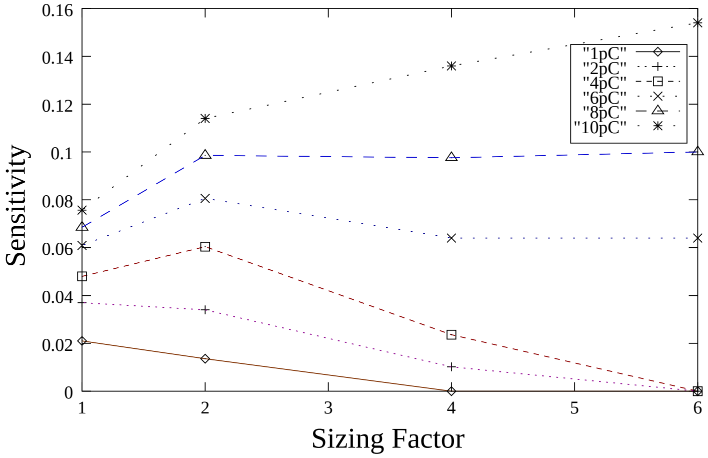


Figure 4. Sensitivity variation with sizing and injection levels

in the digital decimation filter in the Δ-Σ converter and the variation of the sensitivity with sizing and bounded maximum injection level was analyzed. Figure 4 shows that the sensitivity increases and then decreases with sizing for bounded levels of injection. Therefore, it can be concluded that beyond a certain sizing factor, sizing always leads to improvement in sensitivity. For injection levels bounded by 1pC an improvement in sensitivity of 33% is observed by sizing the circuit by twice its original size. Furthermore, the maximum sensitivity point for a higher injection bound occurs at a higher sizing ratio.

5: Conclusions

A generic methodology for the reliability enhancement of ADCs is presented. Fault sensitivity analysis followed by circuit redesign was identified as the fault tolerance strategy to be applied. A new metric, ARE, was proposed which also includes the magnitude of error as opposed to the earlier measure namely, the POF. This methodology was used to first identify *critical blocks* in the FI, Flash and Δ-Σ ADC and then increase their reliability by circuit redesign. By opting for more robust implementations, adding redundancy, pattern detection and transistor sizing, sensitivity gains of as much as 89%, 65.8%, 67.8% and 33%, respectively, were observed.

References

[1] 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.

[2] 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.

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

[4] 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.

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

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

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.