Transient Errors Resiliency Analysis Technique for Automotive Safety Critical Applications

Sujan Pandey and Bart Vermeulen
NXP Semiconductors/Research Lab, High Tech Campus 46, Eindhoven, The Netherlands

***Abstract*— When a single bit is flipped as a result of a transient error in an electronic circuit, its effect can have a severe impact if the circuit is deployed in safety critical domains such as automotive, aeronautics, and industrial automation. In the design phase it is therefore essential to evaluate, and where necessary improve, the resilience of a circuit to all possible transient errors. In this paper, we present a method to analyze the transient error resiliency of a digital circuit. This method is based on an analytical model. It models a transient error as a random function and finds the vulnerable number of bits for each node. We perform a case study on a circuit implementation of a well-known adaptive filter algorithm. The results from the analytical and simulation models show that the analytical model is accurate enough to estimate the effects of transient errors on the performance of a digital circuit. Our analytical method also reduces the analysis time significantly in a design phase.**

I. Introduction

These days a modern car consists of about 100 electronic control units (ECUs), which are connected by in-vehicle networks (IVNs), such as a local interconnect network (LIN), a controller area network (CAN), and a FlexRay network. As the number of ECUs grows for new automotive applications, data has to be transferred from one node to another node at higher data rates. The existing LIN, CAN, and FlexRay networks however do not provide sufficient bandwidth. Although the media oriented systems transport (MOST) technology exists for the automotive industry to achieve this higher data rate, the car manufacturers are currently preparing a large step in in-vehicle networking by deploying Ethernet point-to-point communication between ECUs [1]. They believe the flexibility and scalability of Ethernet will replace the MOST network in the near future by dramatically improving in-vehicle safety, comfort, and infotainment, while significantly reducing network complexity and cabling costs [1]. Furthermore, these Ethernet networks will be suited for safety critical applications, such as drive-by-wire, provided they pass a set of automotive-defined qualification tests with respect to resilience and robustness [2]. Among several fault tolerant aspects, a transient error resilient analysis therefore has to be applied and optimized from an automotive electronic design perspective, where the design trade-offs differ from those of consumer grade products.

There are many sources of transient errors in hardware. For instance voltage scaling is one of the most effective techniques to obtain a low power implementation of a digital circuit. It is still considered for smaller CMOS based technology nodes [3] due to the quadratic dependency of power consumption on the voltage. However, it also increases the susceptibility of the circuit implementation to transient errors. Although the circuit can be tuned for performance, there can still be a slow path in a design, which can results in a path delay fault and that leads to a transient error. In [4] an architecture level resilient circuit design technique (RAZOR) for low power is proposed. This approach performs dynamic voltage scaling based on the dynamic detection and correction of circuit timing errors. A process variation aware low power synthesis technique of FIR filters is proposed in [5]. This technique is based on the identification of the important coefficients by conducting a sensitivity analysis and later logic gates are synthesized by providing a set of constraints. A recent survey of on-chip wires shows that the inter wire spacing decreases rapidly [6] with shrinking geometry, while the height and width of the wires do not scale at the same rate. This increases the chance of crosstalk induced transient error. Classical crosstalk reduction techniques such as spacing and shielding are probably the most commonly used in practice, however, with a cost in additional silicon area. Another powerful method to reduce noise induced by crosstalk in interconnects is to avoid worst case patterns by means of signal encoding schemes [7]. Another source of transient errors in hardware is soft-errors [8]. There exist several works on soft-error rates (SERs) analysis [9], [10] and soft-errors aware circuit design techniques [11], [12], which use space and time redundancy approaches to address the soft-errors.

978-3-9815370-2-4/DATE14/©2014 EDAA

As discussed, there are many sources of transient errors and there are existing mitigation techniques. However, the main research challenge addressed in this paper is the identification of the vulnerable nodes and bits in a digital circuit within a reasonable time, such that different mitigation options can be evaluated early in the architecture exploration phase without affecting the time-to-market of the associated product(s). Thus the main contributions of this work are as follow:

1) We propose an analytical model that captures the effects of transient errors in the hardware implementation of a digital circuit and analyzes the error resiliency for each bit position of a data word at different locations in the circuit.
2) This model allows a full coverage of transient errors in terms of identifying vulnerable nodes and bit positions.
3) This model has a lower run-time than simulation based approaches.

The remainder of this paper is organized as follow. Section II gives a motivational example for transient errors resiliency analysis and design techniques for logic circuits that is used in an automotive safety critical application. Section III details the problem of modeling transient errors and its equivalent noise power at the output of logic circuits and Section IV presents a case study and simulation results for a high speed serial link with an adaptive echo cancellation block. Finally in Section V we discuss the results and give our conclusion.

II. Motivational Example

Fig. 1(a) illustrates a three taps fixed-point adaptive finite impulse response filter, where the term z^{-1} is a delay element, $x(n)$ is an input to the filter, $e(n)$ is a residual error after the adaptive filtering, and $y(n)$ is the filter output. The notation n is for the discrete time. The error $e(n)$ can be expressed as Eq. 1, where $r(n)$ is the reference signal.

$$e(n) = r(n) - y(n) \tag{1}$$

The filter coefficients c_i, c_{i+1}, and c_{i+2} are updated based on the least mean square algorithm (LMS) [13], which can be expressed as

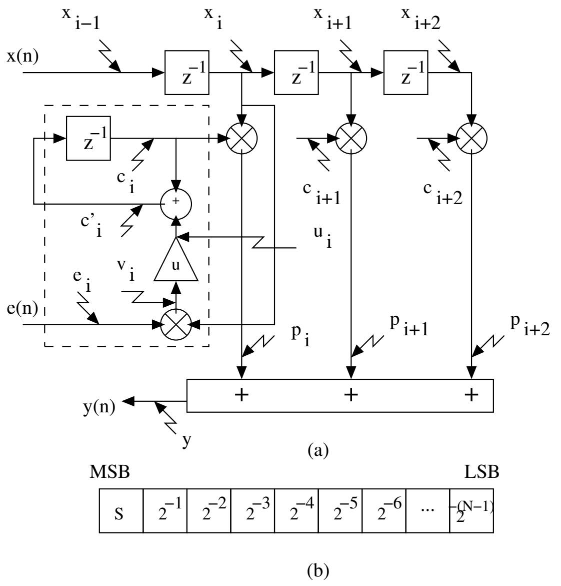

Fig. 1. Adaptive finite impulse response filter. (a) Possible fault locations due to transient errors. (b) A signed N-bits data word

Eq. 2, where the term μ is the gain factor.

$$c_i(n+1) = c_i(n) + \mu \cdot e(n) \cdot x(n-i) \tag{2}$$

The LMS computation blocks for the coefficients c_{i+1} and c_{i+2} are not shown in Fig 1(a) because of space restrictions. The algorithm computes the next value of filter coefficient $c_i(n+1); i = \{1, 2, \ldots, N_{taps}\}$ for tap i using the current value of filter coefficient $c_i(n)$, error $e(n)$, gain μ, and input $x(n)$. The notation N_{taps} stands for the number of taps of a filter. Fig. 1(a) depicts several possible fault locations at the output of each block that can potentially be vulnerable to a transient fault when the adaptive filter is operating normally after convergence. We assume that all transient faults will manifest themselves as errors and may result in a system failure, such as the loss of a communication link. A transient fault at any location in Fig. 1(a) is equivalent to one bit flip out of the N bits (including the sign bit) in a data word, as shown in Fig. 1(b). This results in the replacement of the original value of a data word with a value that ranges from $-\sum_{i=1}^{N-1} 2^{-i}$ to $+\sum_{i=1}^{N-1} 2^{-i}$. If there is a transient fault at one of the locations shown in Fig. 1(a), this can cause an erroneous output of the adaptive filter $y(n)$. For instance, if there is a fault at location c_i', then this incorrect value of this coefficient will be latched by the delay element of the LMS. Later that value will be used to compute the next value of this coefficient as well as to calculate the product of coefficient c_i and input data x_i. The value of the coefficient c_i will consequently be incorrect, until its value converges to its original value by the adaptive algorithm. Similarly, a transient fault at locations c_{i+1}, u_{i+1}, v_{i+1}, and e can have an effect on the output of an adaptive filter. The resulting bit error rate (BER) can however differ for faults at different locations. Fig. 2 shows a simulation result of the coefficient weights c_i, c_{i+1}, and c_{i+2} with i=1 for the adaptive filter shown in Fig. 1(a). It can be observed that when a transient fault is applied at location c_i', the weight of coefficient c_1 changes from its nominal zero to 0.12 and after sometime it returns back to its original weight as shown in Fig. 2. The weights of the other coefficients however do not change as a result of a transient fault at location c_i'.

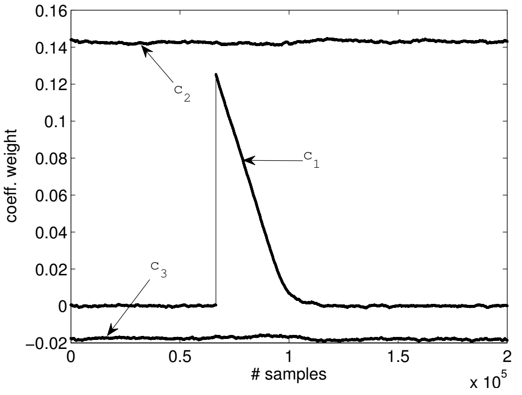

Fig. 2. Weight of coefficients c_1, c_2, and c_3 when a transient error is injected at location c_i' of the adaptive filter shown in Fig. 1(a)

III. Transient Error modeling

This section presents a technique to model the effect of a transient error for an adaptive FIR filter that is discussed in Section II. The modeling technique is however not limited to digital filters. The proposed analytical based modeling approach can also be applied to any data path that is associated with signal processing.

Let $a(n)$ be a fixed point data that is written at the output of a logic block (e.g. adder, multiplier, flip flop etc.) and b(n) be a fixed point data that is read from the same logic block. The value of read data $b(n)$ can differ from the value of the written data $a(n)$, since each bit at the output of a logic block can be flipped with an equal probability as a result of a transient error. Furthermore, we also assume that all outputs are equally vulnerable to an error. Their effect can however be different at the output of different logic blocks. In a typical digital logic circuit, the effect of an error depends on where the error occurs and how it subsequently propagates through the circuit. In a broad sense, the propagation path can be classified as purely combinational, sequential with feedback, and without any feedback. If there is an error at the output of a logic block and the error propagates only through combinational logic, then the effect of this error can only be observed for one clock cycle. In case of sequential logic with feedback, the effect of an error stays until the error is filtered out. This can take more than one clock cycle and depends on the magnitude of the error. While in the case of sequential logic without feedback, the effect of an error can be seen until the error reaches the depth of the sequential logic. This also can be more than one clock cycle. For example in Fig. 1(a), the propagation path of an error at location p_i is combinational until it reaches the output of the filter $y(n)$. For an error at location c_i', the propagation path of the error is sequential with feedback. In case of an error at location x_i, the error propagation path is sequential without any feedback, where the error will disappear after the depth of the shift register.

A bit flip as a result of a transient error can be conceptually considered as an additive noise such that

$$a(n) = b(n) + \delta(n) \tag{3}$$

where we assume that $\delta(n)$ is uniformly distributed in the range $-\sum_{i=1}^{N-1} 2^{-i} \leq \delta(n) \leq +\sum_{i=1}^{N-1} 2^{-i}$ with mean $\mu_\delta = 0$ and variance σ_δ^2. The transfer function of the LMS algorithm integrator in Fig. 1(a) for tap i can be expressed as $H_{c_i}(z) = c_i(z)/u_i(z) = z^{-1}/1 - z^{-1}$. Since we assume that in the worst case each transient fault in the circuit manifests itself as an error, a transient error at location c_i' and its effect to the filter coefficient can be modeled as

$$c_i(n+1) + \delta(n+1) = c_i(n) + u_i(n), \tag{4}$$

where the term $\delta(n+1)$ represents the erroneous data at the output of the LMS adder and variable $c_i(n+1)$ is denoted as c_i'. Eq. 4 can be rewritten to yield the total noise power $\sigma^2_{\epsilon_{c_i}}$, given in Eq. 5, that is added to coefficient c_i

$$\sigma^2_{\epsilon_{c_i}}(n+1) = \sigma^2_\delta \cdot |z \cdot H_{c_i}(z)|^2. \tag{5}$$

Similarly, the noise power $\sigma^2_{\epsilon_{c_i}}$ that is added to coefficient c_i at time $n+1$ for an error at $u_i(n)$ (corresponding to a read "error at location u_i at time n and tap i"), $c_i(n)$, $v_i(n)$, $e(n)$, and $x_i(n)$ is estimated as

$$\sigma^2_{\epsilon_{c_i}}(n+1) = \begin{cases} \sigma^2_\delta \cdot |H_{c_i}(z)|^2, & \text{at } u_i(n) \text{ and } c_i(n) \\ \mu^2 \cdot \sigma^2_\delta \cdot |H_{c_i}(z)|^2, & \text{at } v_i(n) \\ \forall_{\{i=1,\ldots,N_{taps}\}} & \\ \mu^2 \cdot \sigma^2_{x_i}(\sigma^2_e + \sigma^2_\delta) \cdot & \\ |H_{c_i}(z)|^2, & \text{at } e(n) \\ \forall_{\{i=k,\ldots,N_{taps}\}} & \\ \mu^2 \cdot \sigma^2_e(\sigma^2_{x_i} + \sigma^2_\delta) \cdot & \\ |H_{c_i}(z)|^2, & \text{at } x_i(n) \end{cases} \tag{6}$$

and

$$k = 1, 2, \ldots, N_{taps} \tag{7}$$

where the term σ^2_e is the variance of the LMS residual error $e(n)$. When there is a transient error at locations $u_i(n)$ or $c_i(n)$, the noise power at time $n+1$ depends mainly on the magnitude of the error and the transfer function of the LMS integrator, assuming that the power of the transmitted signal is fixed. Similarly, in case of an error at location $v_i(n)$, the filter coefficient noise power for tap i is a function of the transient noise power, the LMS integrator transfer function, and the step size. For all of the above cases, when there is a transient error at tap i, its direct effect in terms of noise power appears only for the coefficient of tap i, while the other taps are noise free. In contrast, a transient error at $e(n)$ affects the filter coefficients of all taps. Thus Eq. 6 can be used to compute the filter coefficient noise power $\sigma^2_{\epsilon_{c_i}}(n+1)$ for all taps i. Similarly, in case of an error at location $x_i(n)$, the filter coefficient noise power is computed for taps that starts from tap k where the transient error originated at time n to the number of taps N_{taps}. The error propagation paths for a set of error locations are summarized in Eq. 7 and are sequential. Thus if there is an error at time n then the noise power for each filter coefficient is computed for time $n+1$. Further, assume that there is no transient error for a while after an error that occurred at time n, the filter coefficient noise power for an error that is associated with the sequential error propagation path with feedback is calculated for time $t > n+1$ as

$$\sigma^2_{\epsilon_{c_i}}(n+m)|_{\{m=2\cdots\infty\}} = \sigma^2_{\epsilon_{c_i}}(n+m-1) \cdot |H_{c_i}(z)|^2 \tag{8}$$

where at first the filter coefficient noise power $\sigma^2_{\epsilon_{c_i}}(n+m)$ for tap i and time $n+2$ is computed based on the noise power that is obtained from Eq. 6 and the LMS integrator transfer function. Later for time $t > n+2$ filter coefficient noise power is computed recursively for each tap i using Eq. 8.

So far Eqs. 5, 6, and 8 allow to compute filter coefficient noise power for each tap i due to a transient error at time n, however, the main interest of this investigation is to estimate the total noise power at the output $\hat{y}$ of a filter which is summarized in Eqs. 9 and 10.

When there is an error at time n in location $c_i(n)$, $x_i(n)$, $p_i(n)$, or $y(n)$ and its primary[1] and secondary[2] error propagation paths are combinational and sequential, respectively, the noise power at the output of the filter at time n is estimated for a tap where the error is originated. The details of the noise power estimation is as follows

$$\sigma^2_\epsilon(n) = \begin{cases} \sigma^2_\delta \cdot \sigma^2_{x_i}, & \text{at } c_i(n) \\ \sigma^2_\delta \cdot \sigma^2_{c_i}, & \text{at } x_i(n) \\ \sigma^2_\delta, & \text{at } p_i(n), y(n) \end{cases} \tag{9}$$

where the terms $\sigma^2_{x_i}$ and $\sigma^2_{c_i}$ are the variances of the input data $x(n)$ and the coefficient c_i for tap i, respectively. For instance at time n if there is a transient error at location $c_i(n)$, $x_i(n)$, $p_i(n)$, or $y(n)$, filter output is erroneous at time n since their primary error propagation path is combinational. While for time $t > n+1$, assuming that there is no more error after $t > n$, the secondary error propagation path of the above error locations is sequential, thus the filter output noise power is estimated as

$$\sigma^2_\epsilon(n+m-1) = \sum_{i=1}^{N_{taps}} \sigma^2_{\epsilon_{c_i}}(n+m-1) \cdot \sigma^2_{x_i} \tag{10}$$

where the variable m, $m \in \{2, \ldots, \infty\}$, is a discrete time index. The filter output noise power is calculated by summing the product of filter coefficient noise power and variance of input data for each tap i, $i \in \{1, \ldots, N_{taps}\}$. After obtaining the filter output noise power from Eqs. (9) and (10), Algorithm 1 finds the bit position as well as the number of bits that are vulnerable to transient errors. The algorithm takes four inputs: the number of bits (N) of a data word, the signal to noise ratio (SNR) that is required to receive bits at the receiver with a certain bit error rate, the signal to noise ratio margin (ΔM), and the estimated noise power σ^2_ϵ. In the algorithm at line 1, the variables vulnerable bit position q_p and the number of bits q are initialized to zero. At line 2, the for loop runs from the LSB to the MSB of the data word. Later at line 3 it computes the noise power $\sigma^2_\epsilon(i)$ at the output, assuming a transient error at bit position i and compares it with the SNR+ΔM. If the computed noise power is greater than the SNR plus the margin, then the variable q is incremented and the bit position is stored in variable q_p at line 4 and 5, respectively. Otherwise, this variable is kept unchanged at line 7. At line 10 the number of vulnerable bits is returned.

IV. Case study

Fig. 3 depicts an echo canceller for a serial communication link with a local transmitter (Tx-L) and a remote transmitter (Tx-R), which communicate via a channel with a transfer function H(t). The complexity of this echo canceller depends on the near-end echo factor. The output of the hybrid $h_o(t)$ goes through an amplifier with gain g and results in an amplified signal $g \cdot h_o(t)$. The continuous time signal $g \cdot h_o(t)$ passes through an eight-bit (one sign bit and seven fraction bits) analog-to-digital converter (ADC). The resulting signal

[1] When an error occurs at time n and at that moment the path that error follows is called the primary path

[2] When an error occurred at time n and path it follows at time $n+1$ is called the secondary path

Transient	Bit position (2^{-1})			Bit position (2^{-2})			Bit position (2^{-3})			Bit position (2^{-4})			Run
error	#bit	Noise (σ_ϵ^2) in dB		#bit	Noise (σ_ϵ^2) in dB		#bit	Noise (σ_ϵ^2) in dB		#bit	Noise (σ_ϵ^2) in dB		time
location	errors	Sim.	Ana.	errors	Sim.	Ana.	errors	Sim.	Ana.	errors	Sim.	Ana.	
$c_i(n+1)$	$\sim 10^8$	-7	-6	3659	-11	-12	14	-18	-18	0	-24	-24	$\sim$3 hrs
$c_i(n)$	$\sim 10^8$	-7	-6	3627	-11	-12	16	-18	-18	0	-24	-24	$\sim$3 hrs
$u_i(n)$	$\sim 10^8$	-7	-6	3584	-11	-12	16	-18	-18	0	-24	-24	$\sim$3 hrs
$v_i(n)$	0	-30	-26	0	-36	-32	0	-42	-38	0	-48	-44	$\sim$1 hr
$e(n)$	0	-26	-23	0	-32	-29	0	-38	-35	0	-44	-41	$\sim$1 hr
$x_i(n)$	0	-27	-25	0	-33	-31	0	-39	-37	0	-45	-44	$\sim$1 hr
$p_i(n)$	1	-8	-6	0	-14	-12	0	-21	-18	0	-27	-24	$\sim$0.5 hr
$y(n)$	1	-8	-6	0	-14	-12	0	-21	-18	0	-27	-24	$\sim$0.5 hr

Table I: Transient errors resiliency analysis based on the analytical and simulation models

```
NO.-OF-VULNERABLE-BITS($N, SNR, \Delta M, \sigma_\epsilon^2$)
  $q_p, q \leftarrow 0$;
  for $i = LSB : MSB$,
    if ($\sigma_\epsilon^2(i) > SNR + \Delta M$) {
      $q++$;
      $q_p(i) = i$; }
    else
      $q = q$;
    endif;
  endfor;
  return $q$;
```

Algorithm 1: Identify the number of vulnerable bits to be protected for transient errors

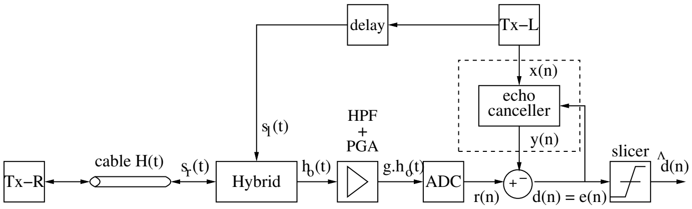

Fig. 3. Echo cancellation for a high speed serial communication link

is transformed into the discrete time signal $r(n)$. In this work, an ADC based echo cancellation system is chosen in order to see the effect of a transient error on the performance of the echo cancellation. Each cycle the echo canceller estimates the echo that is present in the received signal $r(n)$ and subtract the signal $y(n)$, which results in an echo free signal $d(n)$ (is equivalent to signal $e(n)$ in Fig. 1 after echo cancellation). Finally, the slicer maps the amplitude of the signal $d(n)$ to its corresponding symbol $\hat{d}(n)$. Table I summarizes the results of our transient errors resiliency analysis, when we apply it to the LMS filtering algorithm that is employed in the echo canceller for a high speed serial link as shown in Fig. 3. The simulation is carried out in Matlab Simulink and is started from the beginning for each transient error. The column entitled *Error locations* gives a list of locations where a transient error is injected. The names of the error locations in the LMS adaptive algorithm within the echo canceller are the same as in Fig. 1, in order to validate the results from the analytical and simulation models. The columns *Bit position* show which bit of a word is erroneous as a result of an injected transient error. For instance bit position 2^{-1} is the MSB of a word and 2^{-4} is the bit that is the 3^{rd} bit away from this MSB bit. In the table, for each bit position where a transient error is injected, the total number of bit errors and the noise power in dB at the output of the echo canceller are presented for all error locations. The columns *Noise (σ_ϵ^2) in dB* provide the filter output noise power for both simulation as well as analytical models, which are named *Sim.* and *Ana.*, respectively. The columns (*#bit errors*) and *Run time* give the number of bit errors and run time for the Matlab Simulink based simulation model.

V. Discussion and Conclusion

Our analysis correctly shows that the MSB is the most vulnerable and the LSB is the least vulnerable bit to a transient error for most of the data path. However, it is not so obvious for the other bits. All nodes within the hardware are also not equally affected by a transient error. The results show that the effect of an error depends on the bit position, where the error is injected and the architecture i.e., error propagation paths that are combinational and sequential. There are simulation tools, which can also be used to try to find the vulnerable nodes and bits based on simulation in design phase. This is very time consuming when the system complexity grows, preventing to cover all nodes and bits. Our analytical model can however be used to identify a set of vulnerable nodes and bits of hardware early in the design phase and several mitigation techniques can be explored in order to make a design resilient. The results show that the maximum deviation of the analytical noise power from the simulation based noise power is 4dB, which is equivalent to less than one bit [14] according to the quantization theory. This corresponds to a +/-1 bit worst case estimation error in terms of finding the number of vulnerable bits of a word.

References

[1] OPEN Alliance Special Interest Group. http://www.opensig.org.

[2] ISO 26262. http://www.iso.org.

[3] J. M. Rabaey. *Low power design essentials*. Springer, 2009.

[4] D. Ernst et al. RAZOR: A low-power pipeline based on circuit-level timing speculation. In *Proc. Int. Symp. Microarchitecture*, 2003.

[5] N. Banerjee et al. A process variations aware low power synthesis methodology for a fixed-point FIR filters. In *Proc. Int. Symp. Low Power Elec. Des.*, 2007.

[6] R. Ho et al. The future of wires. *Proc. of the IEEE*, 89(4), 2001.

[7] K. Hirose and H. Yasuura. A bus delay reduction technique considering crosstalk. In *Proc. Design Autom. and Test in Europe*, 2000.

[8] R. C. Baumann. Radiation-induced soft errors in advanced semiconductors technologies. *IEEE Trans. on Device and Materials Reliability*, 5(3), 2005.

[9] G. Asadi et al. An analytical approach for soft error rate estimation in digital circuits. In *IEEE Int. Symp. on Circuit and Systems*, 2005.

[10] J. P. Hayes et al. An analysis framework for transient-error tolerance. In *VLSI Test Symp.*, 2007.

[11] D. G. Mavis and P. H. Eaton. Soft error rate mitigation techniques for modern microcircuits. In *Proc. Int. Reliability Physics Symp.*, 2002.

[12] S. Pandey et al. Soft-errors resilient logic optimization for low power. In *Proc. Int. On-Line Testing Symp.*, 2012.

[13] B. Widrow and M. E. Hoff. Adaptive switching circuits. In *WESCON Conv. Rec.*, 1960.

[14] M. J. M. Pelgrom. *Analog-to-digital conversion*. Springer, 2010.