Modeling and Mitigating Transient Errors in Logic Circuits

Ilia Polian, *Senior Member*, *IEEE*, John P. Hayes, *Fellow*, *IEEE*,
Sudhakar M. Reddy, *Fellow*, *IEEE*, and Bernd Becker, *Fellow*, *IEEE*

Abstract—Transient or soft errors caused by various environmental effects are a growing concern in micro and nanoelectronics. We present a general framework for modeling and mitigating the logical effects of such errors in digital circuits. We observe that some errors have time-bounded effects; the system's output is corrupted for a few clock cycles, after which it recovers automatically. Since such erroneous behavior can be tolerated by some applications, i.e., it is noncritical at the system level, we define the critical soft error rate (CSER) as a more realistic alternative to the conventional SER measure. A simplified technology-independent fault model, the single transient fault (STF), is proposed for efficiently estimating the error probabilities associated with individual nodes in both combinational and sequential logic. STFs can be used to compute various other useful metrics for the faults and errors of interest, and the required computations can leverage the large body of existing methods and tools designed for (permanent) stuck-at faults. As an application of the proposed methodology, we introduce a systematic strategy for hardening logic circuits against transient faults. The goal is to achieve a desired level of CSER at minimum cost by selecting a subset of nodes for hardening against STFs. Exact and approximate algorithms to solve the node selection problem are presented. The effectiveness of this approach is demonstrated by experiments with the ISCAS-85 and -89 benchmark suites, as well as some large (multimillion-gate) industrial circuits.

Index Terms—Soft errors, error tolerance, selective hardening, transient faults.

1 Introduction

Transient or *soft errors* are temporary deviations of a circuit's behavior from its correct or reference behavior. They are caused by single-event upsets (SEUs) due to particle strikes, electrical noise, or other environmental effects, and are a major concern in advanced digital ICs [25], [3]. They occur at unpredictable times, and so require probabilistic methods to analyze their effects or to synthesize circuits that mitigate their impact. Most approaches to these issues tend to be heuristic, and employ models that are technology or application-dependent and computationally complex. Existing methods of guarding against soft errors rely on large amounts of redundancy and incur significant overhead costs. This is particularly true for logic circuits, where techniques like ECC that are effective for memories are not applicable.

In many applications, transient errors are acceptable as long as the correct behavior is restored quickly. Suppose, for example, a system relies on input data from unreliable sensors. It must work properly even if a sensor occasionally fails to deliver its data. Such a transient fault can deteriorate the system's output for a few clock cycles, after which it returns to error-free operations without any recovery efforts. Further examples are video systems that tolerate a few missing pixels [18], network applications that handle errors by a retransmission [21], and data processing units implementing commit rollback recovery [9]. In embedded systems with a human end-user, brief deviations of the output data from their correct values may not be perceptible, and therefore, are easily tolerated [5].

In this work, we explore the modeling of transient faults and errors, and the computation of their occurrence probabilities. This is a challenge because soft errors occur at random times, and their impact is highly dependent on circuit's state. To capture the faults of interest, we introduce the *single transient fault* (STF) model, which makes it possible to efficiently estimate error probabilities in logic circuits. STFs can also be used to compute other useful metrics, and they can be evaluated by means of existing methods and software tools designed for (permanent) stuck-at faults.

We also explore the issue of *transient-error tolerance* (TET), which is based on the observation that not all errors at a circuit's output are equally critical. A transient error is considered *noncritical* or *tolerable* if it disappears within a specified time, the *noncritical error period* (NEP), with some specified probability. Noncritical soft errors can be excluded from the SER resulting in a new and more realistic metric called the *critical SER* (CSER). No protection is needed against noncritical errors, thus potentially reducing design costs. To illustrate the proposed methodology, we present a technique for selective hardening of nodes to maximize the probability of error-tolerant operation, measured by a metric called *derating factor*. We study the trade-offs between hardening cost and CSER reduction, using the STF model as a vehicle.

- I. Polian is with the University of Passau, Innstr. 43, D-94032, Passau, Germany. E-mail: polian@informatik.uni-freiburg.de.
- J.P. Hayes is with the Department of Electrical Engineering and Computer Science, University of Michigan, 2260 Hayward Street, Ann Arbor, MI 48109-2122. E-mail: jhayes@eecs.umich.edu.
- S.M. Reddy is with the ECE Department, The University of Iowa, 5324 Seamans Center for the Engineering Arts and Sciences, Iowa City, IA 52242. E-mail: sudhakar-reddy@uiowa.edu.
- B. Becker is with the University of Freiburg, Georges-Koehler-Allee 51, D-79110, Freiburg, Germany. E-mail: becker@informatik.uni-freiburg.de.

Manuscript received 12 Dec. 2008; revised 10 Dec. 2009; accepted 22 Jan. 2010; published online 2 Aug. 2010.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2008-12-0193.
Digital Object Identifier no. 10.1109/2010.26.

1545-5971/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

The rest of the paper is organized as follows: the STF model is introduced in Section 2. The concept of transient-error tolerance is defined in Section 3, and its application to combinational and sequential circuits is discussed in Sections 4 and 5, respectively. The notion of CSER is introduced in Section 6. Exact and approximate algorithms for selective hardening of circuits are described in Section 7. Experimental results are reported in Section 8. Section 9 concludes the paper.

2 Single Transient Fault Model

Early research on soft errors addressed various behavioral and statistical aspects of intermittent and transient faults without defining explicit fault models for them [6], [32]. Soft error modeling has recently received significant attention. There are two broad areas in which advances have been made: low-level modeling and high-level modeling. Low-level approaches [7], [12], [39] aim at exact modeling of physical processes which take place when an energetic particle hits a device in a CMOS digital circuit. One issue of particular interest is the creation of a voltage glitch following a particle strike at a pn junction within a memory cell, a flip-flop, or a logic gate. Another aspect is the propagation of the glitch through the combinational logic to an observable point of the system, e.g., a flip-flop. The glitch may not result in any visible effect due to three *masking* mechanisms: logical masking, latching-window masking, and electrical masking [35].

High-level approaches are concerned with the soft errors that actually affect the system behavior (i.e., are not masked) [8], [23], [19]. One key question is whether the protection mechanisms that the system provides are sufficient to keep the consequences of a soft error in check. High-level methods typically model a soft error as a bit-flip in a memory location, in a state element (flip-flop), or at a logic line of the circuit, with a duration of one clock cycle.

While different low-level models have varying degrees of accuracy, they usually agree on two issues. First, particle strikes of sufficient energy to cause a bit-flip are still relatively rare in ground-level applications, and this is expected to stay so for the near future. Hence, the probability of multiple particle strikes within a short period or even within the same clock cycle is assumed to be negligible. Second, the duration of most particle-induced glitches is less than the cycle time even for high-speed circuits. Although diffusion processes may still be transporting charge, i.e., electrons or holes ionized by particle impact, to a pn junction for a relatively long time, the amount of charge carriers is not sufficient to cause bit-flips. Thus, the immediate effect of one particle strike is confined to the clock cycle in which it occurs. Multicycle effects require that an erroneous logical value be stored in a flip-flop and the state of the circuit is corrupted.

The STF model used in this paper belongs to the class of high-level models. Its focus is on the effects of a particle strike on system behavior. It is not intended to accurately represent low-level details of particle-strike physics or glitch propagation; other (electrical) models exist for this purpose. In this respect, the STF model is fully consistent with standard testing philosophy where relatively simple fault models such as stuck-at faults, four-way bridging faults, and transition faults are used almost universally for test pattern generation, fault simulation, and related tasks.

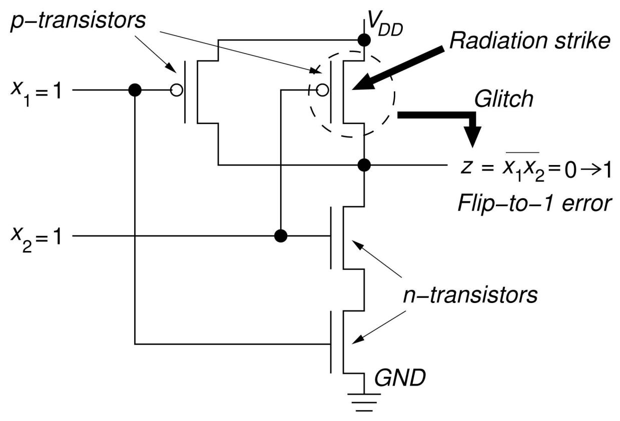


Fig. 1. A NAND2 gate affected by a transient flip-to-1 error.

The target circuits are assumed to be synchronous logic circuits composed of gates, flip-flops, RTL elements, etc. Let $C = (I, O, S, \delta, \lambda)$ denote a sequential circuit (finite-state machine) with k logic lines. Here, I is the input alphabet (the set of input values), O is the output alphabet, S is the set of internal states, δ is the next-state function, and λ is the output function. A *single transient fault* in C, denoted by $f(l/p, x, s)$, is defined by the following properties: 1) it causes the line l to be stuck-at p, where p is 0 or 1, for one clock cycle, and 2) the associated total state of C is x, s, where $x \in I$ and $s \in S$. The number of distinct STFs in C is $2k|I||S|$, where $|I|$ and $|S|$ are the cardinalities of I and S, respectively. While this number is large, it is not intractable if implicit techniques are used (see below). Often, we can restrict attention to small or easily computed classes of STFs.

Observe that there is no cause-effect relationship between an STF and its associated state; $f(l/p, x, s)$ is an STF that occurs when C is in state x, s. The STF model is clearly related to the standard stuck-at fault (SAF) model. Unlike an STF, an SAF l/p persists indefinitely once it occurs and is not associated with specific states. Many simulation and ATPG tools for SAFs can readily be applied to STFs.

More complex transient faults can be modeled by probability transfer matrices (PTMs) [17] which, however, require linear algebra rather than Boolean algebra for their analysis. PTMs also tend to be memory bound. Moreover, it is generally possible to enrich the STF model by low-level data, for example, to assign each gate and each input combination a specific susceptibility. Consider the two-input CMOS gate NAND2 in Fig. 1. A radiation strike can upset one or more of its transistors, causing the output z to undergo a transient flip-to-0 or flip-to-1 error. The specific error depends in part on the input pattern $x_1 x_2$ when the strike occurs.

Input $x_1 x_2 = 11$ flips z from 0 to 1 if one of the gate's p transistors is upset, as in Fig. 1. Under input patterns 10 and 01, only one n-transistor is susceptible to the strike. With input 00, both n-transistors must be upset to produce an output bit-flip. The probability that a transistor or a combination of transistors is upset by a particle can be calculated using an SER analysis tool such as TMC-DASIE [31] which is based on accurate modeling of nuclear reactions. Based on these data, soft error susceptibility or the upset probability of the NAND2 gate under inputs 00, 01, 10, and 11 can be set to some values p_{00}, p_{01}, p_{10}, and p_{11}, respectively.

3 Transient-Error Tolerance

An error that finds its way into the internal state may be eliminated by suitable design methods, or it may simply be flushed out automatically by normal inputs that happen to take the circuit to a correct state. The probability of such *self-recovery* is of interest. A circuit C is *transient-error-tolerant* for the STF set F with *noncritical error period* k and self-recovery probability p_{sr}, denoted by (F, k, p_{sr})-TET, if the internal states of the erroneous and error-free circuits are the same after k cycles with the probability of at least p_{sr}, assuming equiprobable inputs. C is (k, p_{sr})-TET if the conditional probability that its state is error-free k cycles after an arbitrary STF occurs is at least p_{sr}.

For $p_{sr} = 1.0$, the circuit C is $(F, k, 1.0)$-TET, if the state of C affected by any member of F and that of the error-free circuit are the same after k clock cycles for all possible input sequences. Note that the initial state is implicitly included in each fault f of F. A combinational circuit is thus always $(1, 1.0)$-TET since it recovers from an STF after one clock cycle. A sequential circuit is $(1, 1.0)$-TET for all faults that influence only its primary outputs, but not its next state (memory) part. A feedback-free pipelined circuit of depth m is $(m, 1.0)$-TET for all STFs. It has been shown for a motion estimation circuit that it is TET with $p_{sr} = 1.0$ for over 70 percent of its faults with period 96 [28]. This means that if one of these faults occurs, the encoding of the image being processed may be suboptimal. However, the encoding of subsequent images, starting with the next image completely transmitted, will be performed as if the fault has never occurred.

Further error-tolerance concepts of interest include error significance and error rate. *Error significance* denotes the impact of an error from an application point of view [5], [15]. For instance, peak signal-to-noise-ratio, structural similarity, and psychovisual deviation are used as error significance metrics for a JPEG encoder in [26]. The notion of *error rate* defines errors with a sufficiently low probability of occurrence as noncritical [34]. A metric called SBER, which combines error rate and error significance, has been proposed in [27]. Error significance and error rate have mostly been studied for permanent faults although they can be readily applied to soft errors.

Another class of potentially noncritical errors affects performance-enhancing units such as branch predictors in microprocessors [2]. These errors do not prevent the calculation of the correct results but delay (time-shift) its completion. Similar effects can also be observed for errors handled by a commit-rollback-recovery scheme [9]. Noncritical error effects have also been studied in the context of real-time systems under the heading of imprecise computation [20].

4 Errors in Combinational Logic

Since there is no internal state s, an STF for a combinational circuit C reduces to the form $f(l/p, x)$. An STF error then corresponds to SAF l/p and a test x for l/p, since by definition, a test propagates the fault effect (error) to a primary output. Let C have k lines, n primary inputs, and a single primary output z, and assume that all STFs are equiprobable. The STF error probability $p_{err}(z)$ is the total number of possible errors produced at z by STFs divided by the total number of possible STFs as follows:

$$p_{err}(z) = \frac{\sum_l \text{No. of tests for the faulty line } l}{k2^{n+1}}. \qquad (1)$$

Suppose C is an n-input gate of the (N)AND or (N)OR type. Equation (1) implies that

$$p_{err}(z) = (n + 2^{n-1})/(n+1)2^n. \qquad (2)$$

Here, $p_{err}(z)$ approaches $1/(2(n+1)) = 1/(2k)$ as n increases, which means that gates with greater fan-in are more likely to mask or tolerate STF errors. In the case of a gate of the X(N)OR type, $p_{err}(z) = 1/2$. In general, the STF $f(l/p, x)$ is only detectable by input vectors that make $l = \bar{p}$; it is undetectable if $l = p$. Gates of the (N)AND/(N)OR type and gates of the X(N)OR type are therefore the best and worst cases, respectively, in terms of error masking among all n-input logic functions. We conclude that for any k-line single-output combinational circuit

$$1/(2k) \le p_{err}(z) \le 1/2. \qquad (3)$$

Hence, STFs capture our intuitive notions of transient error propagation and logical masking quite well. Note that electrical and latching window masking [36] are technology-dependent and not included in our model. As will be discussed in Section 7.2, recent experimental data [40] suggest that these masking mechanisms have little influence when selecting circuit nodes for hardening.

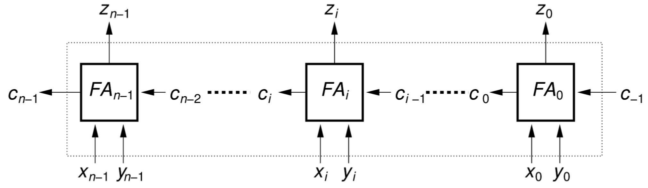


Fig. 2. An n-bit ripple-carry adder.

4.1 Ripple-Carry Adder

Consider the n-bit ripple-carry (RC) adder of Fig. 2. It is constructed from a full adder FA_i, an RTL element realizing two functions, the sum z_i, and the carry-out c_i. It has n FA_i stages, $2n+1$ inputs, and $n+1$ outputs. There are $4n+1$ lines that can be faulty, so the total number of STFs is $(4n+1)2^{2n+2}$.

We can compute the output error probabilities $p_{err}(z_i)$ by counting the errors produced at z_i by STFs associated with a representative element FA_i. We can also subdivide the errors on z_i and c_i into two groups as follows: local errors due to faults in FA_i itself, and remote errors that originate in preceding stages, and enter FA_i via its carry-in line c_{i-1}. The local error count at z_i is 2^{2n+3} and the corresponding remote error count is $e(c_i) = 2^{2n+2} + 2e(c_{i-1})$, leading to the following formula for the STF error probability on z_i:

TABLE 1
STF Error Probabilities in RC Adders

Size n	Carry c_{n-1}	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
1	0.250	0.400							
2	0.181	0.222	0.306						
4	0.112	0.118	0.282	0.185	0.195				
6	0.079	0.080	0.110	0.125	0.132	0.136	0.138		
8	0.060	0.060	0.083	0.095	0.100	0.103	0.105	0.105	0.106

$$p_{err}(z_i) = \frac{2^{2n+3} + 2^{2(n-i)}(e(c_{i-1}) - 2^{2i+1})}{(4n+1)2^{2n+2}}. \tag{4}$$

Table 1 shows some p_{err} values derived from this analysis. Such data provide useful information about a circuit's error propagation or masking properties. For example, the $p_{err}(z_i)$s of the RC adder increase slowly with i, eventually leveling off. The error probability $p_{err}(c_{n-1})$ at the final carry-out is always less than that of the z_i (sum) outputs.

4.2 Exact Calculation of p_{err}

Consider a general, n-input combinational circuit C with m output functions $Z = z_1, z_2, \ldots, z_m$. The calculation of the STF error probability $p_{err}(z_i)$ of the output z_i according to (1) can be expressed in terms of the detection probabilities of stuck-at faults as follows:

$$p_{err}(z_i) = \frac{\sum_{f \in SAF} DP(f, z_i)}{|SAF|}, \tag{5}$$

where SAF is the set of all the stuck-at faults, $|SAF|$ is the size of this set, and

$$DP(f, z_i) = \left|z_i \oplus z_i^f\right| / 2^n \tag{6}$$

is the detection probability of stuck-at fault f at the output z_i. On the right-hand side of (6), z_i is the function of the circuit C's ith output, and z_i^f is the same function with fault f present. The function $z_i \oplus z_i^f$ maps all input vectors for which the fault-free and the faulty circuits calculate different values to 1 and all other input vectors to 0. The term $|z_i \oplus z_i^f|$ denotes the cardinality of this function's on-set, i.e., the number of input vectors mapped to 1, and thus, the number of all input vectors for which fault f is detected at the output z_i. Each stuck-at fault in (5) subsumes 2^n STFs: an STF $f(l/p, x)$ is represented by the l stuck-at-p fault. Note that the stuck-at faults are only used as a means for efficient computation of $p_{err}(z_i)$; the physical disturbances modeled by STFs (transient errors) and stuck-at faults (permanent defects) are entirely different.

The circuit C's error probability $p_{err}(C)$ considering all m outputs is expressed by

$$p_{err}(C) = \frac{\sum_{f \in SAF} DP(f)}{|SAF|}, \tag{7}$$

with

$$DP(f) = \left| \bigvee_{i=1}^{m} \left(z_i \oplus z_i^f\right) \right| \bigg/ 2^n. \tag{8}$$

Equations (7) and (8) can be evaluated efficiently using symbolic simulation with binary decision diagrams (BDDs) representing functions z_i and z_i^f. They can also be approximated using random-pattern simulation techniques from [4] or the methods described in the next section.

4.3 Approximate Calculation of p_{err}

While exact calculation of detection probabilities (8) is an NP-complete problem, it is possible to compute the values of $DP(f)$ by a fast (linear in the size of the circuit) heuristic. These approximate detection probability values can be used in (7) for circuits for which BDDs cannot be constructed. Obviously, the resulting p_{err} values are approximations of the exact values.

In this work, approximate calculation of detection probabilities is performed in a way similar to STAFAN [14] and PROTEST [37]. Two passes through the circuit are required. In the first pass of the algorithm, the circuit is traversed in topological order (from inputs to outputs). The signal probabilities of all lines in the circuit are determined by assigning a signal probability of 0.5 to each primary input and deriving the signal probability on the output of a gate from those on the gate's inputs. In the second pass, the circuit is traversed in reverse topological order (from outputs to inputs). The detection probabilities of faults on the circuit's outputs are computed from their signal probabilities. Then, detection probabilities of all other lines (inputs of some gate g) are determined from the detection probability of g's output and the signal probabilities on g's side-inputs. Details of the algorithm can be found in [30].

5 Errors in Sequential Circuits

We illustrate the foregoing TET concepts and formalisms using a small serial adder (SA) circuit with just two states for which a Markov model is easily constructed. We then propose a general method to calculate TET properties for sequential circuits. Experimental results for ISCAS-89 circuits can be found in Section 8.3.

5.1 Serial Adder

Consider the serial adder in Fig. 3. This sequential circuit adds two binary numbers X and Y serially (bit by bit) to produce the sum Z. It comprises a combinational adder FA, a D flip-flop DFF which stores the carry bit c, and, counting only the lines visible in Fig. 3, a total of 80 STFs.

Now consider the effect of an STF f on the output z and the next state c in the initial clock cycle 0 when the STF occurs. Table 2 places each STF f into one of four sets based on whether or not f produces an erroneous value of z and/or c in clock cycle 0. As implied by (3), half the possible STFs are undetectable, so SA is $(F_0, 0, 1.0)$-TET. Class F_1

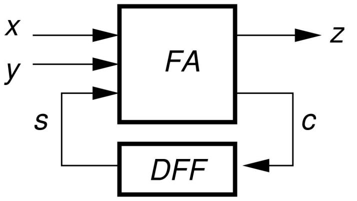


Fig. 3. RTL model of a serial adder SA.

represents the case where SA's output, but not its internal state, is erroneous; hence SA is $(F_{01}, 1, 1.0)$-TET, where $F_{01} = F_0 \cup F_1$. Thus, if an error is acceptable at z in cycles 0 and 1 only, i.e., the NEP $k = 1$, then all STFs in F_{01} are tolerated. Since these represent 75 percent of the possible STFs, we can say that SA is $(1, 0.75)$-TET. SA is also $(1, 0.875)$-TET, because although F_2 and F_3 can produce error states in cycle 1 and beyond, the probability of them doing so is 0.5, as will become clear later, and the share of F_2 and F_3 is 0.25 yielding $0.75 + 0.5 \cdot 0.25 = 0.875$.

It is easily seen that the STFs in Table 2 include a few faults that can leave an error lurking indefinitely in the circuit's internal state. Thus SA is not $(k, 1.0)$-TET for any finite k when all STFs are considered.

Self-recovery can be analyzed by Markov models [6]. We use them to compute the probability p_{good} of the circuit going from erroneous states induced by STFs to correct states within k clock cycles. We can then say that the circuit is (k, p_{good})-TET.

Considering SA again, its state transition graph is in Fig. 4a. For half the input patterns xy, the next internal state c (but not the output z) is independent of the initial state. $xy = 00$ always sets the internal state to $c = 0$, while $xy = 11$ always sets c to 1. The other two xy values leave the internal state unchanged. Hence, $xy = 00$ and 11 automatically correct an erroneous state of SA; the other two input vectors do not.

It follows that a transition from either of SA's two internal states has probability 1/2. Once returned to a good state, SA operates correctly until a new fault occurs. This leads to the Markov model shown in Fig. 4b. If all four input combinations xy are equiprobable, the probability of remaining in a bad (erroneous) state k cycles after entering a bad state is 0.5^k. The circuit is thus $(k, 1 - 0.5^k)$-TET, i.e., it is TET with NEP k and the probability of self-recovery $1 - 0.5^k$. Fig. 5 shows how the probability $p_{err}(k)$ of an error lurking in SA decreases exponentially with time. Thus, we can, in cases like this, derive an analytic formula for $p_{err}(k)$ that can be used to determine error tolerance with respect to given thresholds on $p_{err}(k)$ or k.

TABLE 2
Classification of All STFs Affecting SA of Fig. 3

STF class	Class definition	No. of STFs in class
F_0	No effect on SA	40
F_1	Erroneous output z in cycle 0; no effect on next state c in cycle 0	20
F_2	No effect on z in cycle 0; erroneous c in cycle 0	12
F_3	Erroneous z in cycle 0; erroneous c in cycle 0	8

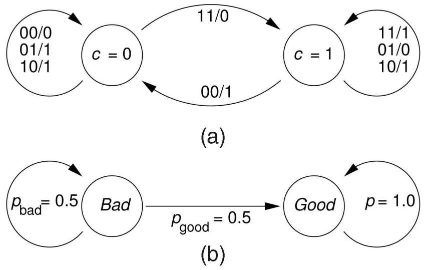


Fig. 4. (a) State transition graph, and (b) Markov model for SA.

5.2 Calculation of $p_{err}(k)$

For large or unstructured sequential circuits, computer simulation can be used to determine $p_{err}(k)$, the probability that the circuit's state is still erroneous after k clock cycles, numerically. The procedure, which reduces the calculation of $p_{err}(k)$ to the calculation of p_{err} in a combinational circuit (described in Sections 4.2 and 4.3), is outlined next.

First, we construct a combinational k time-frame expansion $TFE_k(C)$ of the sequential circuit C. $TFE_k(C)$ consists of k copies of C's combinational core, denoted by $C_1, \dots, C_k$. The secondary outputs of C_j are connected to the secondary inputs of C_{j+1}; the primary inputs (outputs) of C_j represent the primary inputs (outputs) of the sequential circuit C in the jth clock cycle. The secondary inputs of C_1 correspond to the initial state of C, and the secondary outputs of C_k correspond to C's state after k clock cycles. $TFE_k(C)$ is employed in automatic test pattern generation for sequential circuits as follows: the behavior of a circuit affected by a

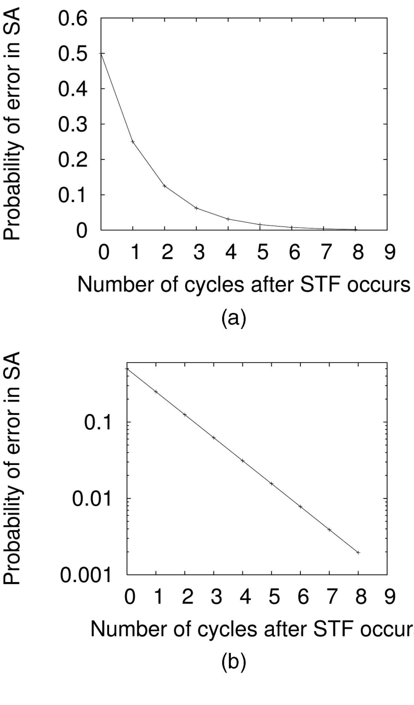


Fig. 5. Probability $p_{err}(k)$ of an error in SA's state k cycles after an STF shown (a) in normal form and (b) in logscale form.

permanent fault, e.g., a stuck-at fault, is obtained by injecting the fault into each C_j [1].

Second, $p_{err}(k)$ is calculated by determining the detection probabilities of the stuck-at faults in circuit $TFE_k(C)$ using either the exact method from (8) or the heuristic from Section 4.3 and applying (7). There are two important restrictions as follows: first, only single-stuck-at faults in the first time frame, i.e., C_1, are considered. These faults represent a disturbance which occurs in the first clock cycle and disappears later, which corresponds to the STF model. (In contrast, permanent faults considered during sequential test generation are injected into all C_js, leading to multiple faults.) Second, only the secondary outputs of C_k are taken into account (7), because they represent the circuit's state after k clock cycles.

With these restrictions, p_{err} of $TFE_k(C)$ yields $p_{err}(k)$. The probability of self-recovery within k cycles $p_{sr}(k)$ is obtained as $1 - p_{err}(k)$. Note that $p_{err}(k)$ is different from $P^c(SF|FF_i)$ studied in [4]. $P^c(SF|FF_i)$ is the probability that a soft error in flip-flop FF_i leads to an error at a primary output in at least one clock cycle within c clock cycles after the soft error has occurred. $P^c(SF|FF_i)$ grows with increasing number of cycles c, often reaching values close to 1, while $p_{err(k)}$ declines with increasing number of cycles k.

6 Critical Soft Error Rate

The soft error rate (SER) is the frequency with which soft errors occur. The CSER describes the rate of errors that can lead to serious disruptions of circuit operation, while the SER describes the frequency of bit-flips in the circuit, some of which do not propagate anywhere. Hence, the CSER appears to be the more useful indicator of the circuit's susceptibility to soft errors. Based on this assumption, we study the relationship between CSER and SER, as well as ways to minimize the CSER (rather than the SER) by selective hardening.

Let $e_1, e_2, \ldots, e_n$ be different soft errors that potentially affect circuit operation according to a soft error model. Possible error models include the STF model introduced earlier, STF enriched by electrical information, and aggregated models where one error is composed of multiple STFs. The SER of an individual error and that of the whole circuit are introduced first. Let the SER of an error e, $SER(e)$, denote the probability that the error occurs, i.e., a logical value on a circuit node flips. Note that no propagation of the error to a primary output or a memory element is required. Let the SER of the circuit be the sum of the SERs of the individual errors as follows:

$$SER = \sum_{i=1}^{n} SER(e_i). \qquad (9)$$

This assumes that soft errors are independent stochastic events (as pointed out above, this assumption is justified in current and future CMOS technologies where soft errors are few and far between). The actual SER is determined by the physical parameters of the circuit and its manufacturing technology. Calculation of SER taking physical parameters into account is beyond the scope of this paper; a large body of literature exists on this subject [7], [12], [29], [39].

Soft errors that are masked and have no effect on the outputs of the circuits are always noncritical, such as the errors in the class F_0 of the serial adder. If the masking probability of an error e_i is $p_{mask,i}$ and all errors that are not masked are assumed to be critical, then the CSER is given by

$$CSER = \sum_{i=1}^{n} SER(e_i) \cdot (1 - p_{mask,i}). \qquad (10)$$

If the value $p_{mask,i}$ is calculated taking only logical masking into consideration, then the error probability p_{err} introduced above can be expressed:

$$p_{err} = CSER/SER. \qquad (11)$$

It is possible to take into account further masking mechanisms defined by low-level error models.

Observe that the error rate actually measured on the circuit's outputs, e.g., in an accelerated testing experiment using a radiation source, is CSER rather than SER according to our definitions, because masking is already accounted for in such an experimental setup. In this paper, SER is the error rate which can be derived from the individual error rates of the components as indicated (9). This simplifies the extension of CSER into the framework of transient-error tolerance, as described below.

Suppose that an error effect is defined as noncritical if it does not affect the state of the circuit after k cycles where k is the NEP. Let $p_{mask,i}(k)$ be the probability that the effect of error e_i does not propagate to the circuit state after k cycles. Since we study values of k which are far smaller than the likely number (trillions) of clock cycles between two soft errors, we can safely assume that a second soft error will not occur within k cycles.

While the SER of the circuit is not modified by such a loosening of the specification, the CSER changes. We define

$$CSER(k) = \sum_{i=1}^{n} SER(e_i) \cdot (1 - p_{mask,i}(k)), \qquad (12)$$

and observe that for $p_{err}(k)$, the probability that the circuit is still affected by an error after k cycles, the following relationship similar to (11) holds:

$$p_{err}(k) = CSER(k)/SER. \qquad (13)$$

7 Selective Hardening

The goal of any hardening strategy is to create a circuit that meets some SER or CSER objectives. We assume that individual circuit nodes can be hardened, thus reducing their SER contribution (9) and improving the overall SER and CSER. Selective hardening has been investigated in [22], [25], [33].[1] While the actual hardening mechanism is out of this paper's scope, we mention several approaches from the literature. In a study by NXP [25], logic gates to be hardened were simply duplicated, leading to SER improvement by 60 percent at 20 percent cost. Garg et al. [10] supplemented the duplication by connecting the outputs of the gates by a diode or a transistor. There are also techniques to harden the flip-flops of the circuit, or a subset of its flip-flops [16], [24], [38].

We discuss a minimum-cost selective-hardening strategy to achieve a given CSER target. In contrast to [10], [22], [25],

[1] A similar approach in the context of error detection is found in [13].

we formulate the selection of the nodes to be hardened as a general optimization problem. We first summarize our assumptions on the selective hardening mechanism. Then, we discuss possible optimization criteria and solution approaches for both $CSER$ targets (no error effect is allowed to show up on the outputs) and relaxed $CSER(k)$ targets (an error is required to be removed from the system after k cycles). We conclude with an example using the serial adder introduced above.

7.1 Selective Hardening Mechanism

Let the set of STFs $f(l/p, x, s)$ with identical l and p be called an error $e_{l/p}$. An error $e_{l/p}$ can be regarded as a stuck-at-p fault on the line l that persists for only one clock cycle. The number of such errors, n, equals $|SAF|$. Hardening to eliminate or mask the effect of an STF $f(l/p, x, s)$ at a circuit node l masks all the STFs included in the corresponding error $e_{l/p}$, and for brevity, we say that $e_{l/p}$ is hardened. In order to determine which fault sites should be hardened, we consider the set of all errors $e_{l/p}$, denoted by $e_1, e_2, \ldots, e_n$, and determine a subset of the errors to harden so as to meet the desired specification for SER. We assume that any subset of the errors $e_1, e_2, \ldots, e_n$ can be selected for hardening.

If error e_i is selected for hardening, cost c_i is incurred and the susceptibility of the circuit to error e_i changes from $SER(e_i)$ to $s_i \cdot SER(e_i)$, where $0 \leq s_i < 1$. The actual costs of hardening are determined by the particular technique used. For instance, gate duplication [10], [25] would increase the gate count by the number of gates hardened. Note that this increase may not translate into an area increase due to potential routing overhead. Moreover, the duplicated gates are more expensive in [10] than in [25]. There could also be some optimization potential from sharing the duplicated logic. Using BISER [38] as the hardening mechanism for a flip-flop would lead to little area but significant energy overhead. An accurate model of hardening costs c_i is not in the scope of this paper. In our experiments, we assume that the hardening costs are proportional to the number of gates to be hardened (by setting all c_is to 1).

If errors e_{i_1}, e_{i_2}, ..., e_{i_m}, $m \leq n$, have been selected for hardening, the SER of the selectively hardened circuit becomes

$$SER_{sh} = \sum_{j=1}^{m} s_{i_j} \cdot SER(e_{i_j}) + \sum_{i \notin \{i_1, i_2, \ldots, i_m\}} SER(e_i). \tag{14}$$

Equivalently, s_i can be set to 1 for all errors e_i not selected for hardening ($i \notin \{i_1, i_2, \ldots, i_m\}$). Then

$$SER_{sh} = \sum_{i=1}^{n} s_i \cdot SER(e_i). \tag{15}$$

If all errors visible at the outputs are considered critical, the CSER becomes

$$CSER_{sh} = \sum_{i=1}^{n} s_i \cdot SER(e_i) \cdot (1 - p_{mask,i}). \tag{16}$$

If errors that are flushed out of the circuit state after k cycles can be tolerated, the CSER is given by

$$CSER_{sh}(k) = \sum_{i=1}^{n} s_i \cdot SER(e_i) \cdot (1 - p_{mask,i}(k)). \tag{17}$$

The *derating factor* D is the SER of the nonhardened circuit divided by the critical soft error rate of the selectively hardened circuit as follows:

$$D = SER/CSER_{sh}, \tag{18}$$

$$D(k) = SER/CSER_{sh}(k). \tag{19}$$

The derating factor indicates the combined improvement from the selective hardening and the consideration of only the critical errors instead of all errors. When no hardening is done and the SER contributions of individual nodes are equal, $D = 1/p_{err}$. If the "raw" SER of the circuit is known, e.g., from accelerated testing of circuit elements, the derating factor can be used to calculate the CSER of the hardened design, which is the relevant characteristic of the circuit as argued earlier.

The costs c_i may be derived from the extra area or power consumption associated with hardening. We assume that these costs are additive, i.e., hardening the circuit against a subset of errors has a cost equal to the sum of the costs c_i corresponding to the individual errors.

7.2 Selecting Nodes for Hardening

One possible goal of selective hardening is to achieve a given level of CSER at lowest cost by selecting an optimal subset of errors for hardening. In other words, given a desired CSER threshold Th_{CSER}, select a subset of errors for hardening such that $CSER$, as defined (16), does not exceed the threshold, and the sum of the costs is minimal. Alternatively, a *derating threshold* Th_D may be given. Then, the derating factor D must exceed the derating threshold Th_D, while the costs are minimized. The same criteria can be set for $CSER(k)$ and $D(k)$.

Equations (7) and (11) are equivalent to

$$CSER = \frac{\sum_{i=1}^{n} SER(e_i) \cdot DP(e_i)}{|SAF|}. \tag{20}$$

Hardening a circuit against an individual error e_i corresponds to replacing $SER(e_i)$ by $s_i \cdot SER(e_i)$ in the numerator. Assuming that $s_i = 0$, i.e., hardening against a soft error eliminates the possibility of error altogether, the corresponding detection probability $DP(e_i)$ can be removed from the numerator. It is possible to sort the errors e_i according to $DP(e_i)$ adjusted by the costs of hardening c_i and to harden the circuit against the errors in the sorted list until the CSER or the derating target is achieved.

The definition of p_{err}, and thus derating, focuses on logical masking, and does not take temporal or electrical masking into account. A recent study [40] evaluated the validity of node selection based on logical masking information only (as done in this work), using an accurate electrical-level single-event transient simulator based on the novel UGC particle strike model. It turned out that the derating predicted using logical masking only and the derating measured by the accurate simulator tracks reasonably well, provided that the factors s_i are accounted for. (Note that the inverse of s_i, called the local hardening factor or LHF, is used in [40].) An advantage of a method working at the logical level is its applicability in early design steps, when low-level information necessary to quantify temporal and electrical masking is not available.

7.3 Example

Suppose that the serial adder SA must be TET with $k = 2$ and $Th_D = 15$, i.e., the CSER must be at least 15 times less than the SER. This means that the adder must self-recover after two cycles with probability $1 - 1/15 \approx 0.933$ or more. From Fig. 5, we see that $p_{err}(2) = 0.125$, i.e., $D(2) = 8$, which means that SA does not meet the specification and requires hardening. Calculation of the detection probabilities for SAFs is illustrated by the stuck-at-0 fault on line x, denoted by $x/0$. For this fault, there are four initial state/input sequences which result in an erroneous state after $k = 2$ cycles: $sx_1y_1x_2y_2 = 01110$, 01101, 11010, and 11001. Consequently, the test set size in (8) is four and the detection probability $DP(x/0)$ is $4/2^5 = 0.125$. Similarly, the detection probability also equals 0.125 for faults $x/1$, $y/0$, $y/1$, $s/0$, and $s/1$; it is 0.25 for faults $c_{in}/0$ and $c_{in}/1$; and it is zero for faults $z/0$ and $z/1$. For simplicity, we assume that all s_is are 0 and all c_is are 1, i.e., the hardening against any of the errors is associated with identical costs and eliminates any possibility that the error occurs.

A fault with the largest detection probability is selected first, e.g., $c_{in}/0$. This reduces $p_{err}(2)$ by $DP(c_{in}/0)/|SAF| = 0.025$, i.e., from 0.125 to 0.1. The derating is $1/0.1 = 10$, which is still less than 15. Fault $c_{in}/1$, which is selected next, results in $p_{err}(2) = 0.075$ and $D(2) \approx 13.3 < 15$. Selecting a third fault such as $x/0$ results in $p_{err}(2) = 0.0625$ and $D(2) = 16$, which is above Th_D. The specification has been met by hardening node c against both flip-to-0 and flip-to-1 errors and hardening node x against flip-to-0 errors only. The hardening cost is three or 30 percent of the cost of hardening all nodes (which is 10). Note that by selecting only the most critical nodes to harden, the achieved reduction of 50 percent for $p_{err}(2)$ exceeds the proportion of hardened nodes (30 percent).

In cases where a Markov model can be constructed, such as SA, it is not necessary to consider all k cycles explicitly. Since we know that for SA, $p_{err}(2) = p_{err} \cdot (0.5)^2$ (or, equivalently, $D(2) = D/(0.5)^2$) holds, it suffices to select faults considering only the probability that the circuit enters state Bad (in Fig. 4b) in the beginning. To meet the specification, this probability must be below $1/(Th_D \cdot 0.5^2) \approx 0.266$, while the actual probability is 0.5. It is easy to see that by selecting faults $c_{in}/0$, $c_{in}/1$, and $x/0$, the probability becomes 0.25. A significant reduction in computational complexity is thus achieved with no loss of accuracy. Hence, it is preferable to construct Markov models of the target circuits where feasible.

7.4 Node Selection Using Approximate Information

The method outlined in Section 7.2 can employ approximate detection probabilities calculated using the method from Section 4.3. In this case, the value of CSER or derating achieved so far may be erroneously qualified as satisfying the target, i.e., exceeding the threshold, while the actual CSER or derating (calculated using exact information) is below the threshold.

To compensate for this possible overestimation of the solution quality, we set an optimistic derating or CSER target for the approximate method by multiplying the threshold with a *safety margin* $SM \geq 1$. For instance, for a derating threshold of 10 and $SM = 2$, we terminate node selection when the derating (calculated using approximate information) exceeds $20 = 10 \cdot 2$. The influence of SM on the robustness of the results is discussed in Section 8.2.

TABLE 3
p_{err} and Cost of Selective Hardening for the ISCAS-85 Combinational Benchmarks

Cct.	p_{err}	$Th_D=10$		$Th_D=100$		$Th_D=1000$		$Th_D=10000$	
		Cost	%	Cost	%	Cost	%	Cost	%
c17	0.299	16	47.06	34	100	34	100	34	100
c432	0.105	5	0.58	557	64.47	752	87.04	827	95.72
c499	0.198	158	15.83	563	56.41	847	84.87	965	96.69
c880	0.198	317	18.01	1130	64.20	1498	85.11	1648	93.64
c1355	0.152	246	9.08	1622	59.85	2197	81.07	2564	94.61
c1908	0.185	672	17.61	1963	51.44	2685	70.36	3430	89.88
c2670	0.167	675	12.64	2899	54.29	3883	72.72	4314	80.79
c3540	0.127	332	4.69	3300	46.61	5372	75.88	6388	90.23
c5315	0.135	676	6.36	6750	63.50	9537	89.72	10241	96.34

8 Experimental Results

Next, we describe an application of the foregoing methodology to the mitigation of soft errors in combinational and sequential circuits.

8.1 Combinational Circuits

Using symbolic simulation, we calculated the effect of selective hardening on p_{err} for nine combinational ISCAS-85 benchmark circuits for which BDD-based simulation was feasible. Table 3 gives p_{err} (with $k = 0$) for a circuit with no hardening and, for four values of the derating threshold Th_D, the cost of selective hardening to achieve $(1/p_{err}) \geq Th_D$, assuming $s_i = 0$ and $c_i = SER(e_i) = 1$ for all errors. (The overall cost of the hardening corresponds to the number of stuck-at faults excluded from (20) and the percentage of these faults among all faults.) The detection probability in the numerator of (20) has been obtained using (7). For this purpose, we constructed, for each stuck-at fault, BDDs for the outputs of the fault-free and faulty circuits using the CUDD package and applied BDD operations provided by that package to calculate the test set and its size. We employed arbitrary precision arithmetic to represent large numbers. Recall that every stuck-at fault represents a class of STFs in this analysis.

The probability p_{err} of a soft error showing up on an output is between 0.1 and 0.2 for all circuits except c17, for which it is 0.3. This means that only approximately every fifth to tenth soft error is actually visible on a circuit output and the other faults are masked by the circuit itself. These findings are consistent with the data in [22], [25], even though the modeling in these works took low-level information into account. The cost of selective hardening with a derating factor of 10 is generally quite low. In contrast, higher derating thresholds require overheads which are probably unacceptable. Recall that no high derating factors may be required for combinational circuits as the faulty effect will definitely last only for one clock cycle. Hence, selective hardening is useful for combinational circuits if the derating threshold is not much larger than the actual derating figure, which is likely to be the case.

8.2 Approximate Method

We applied the node selection algorithm based on approximate detection probabilities to the combinational parts of the ISCAS-89 circuits. We used a derating threshold of 10 and three different safety margins. Fig. 6 shows the actual derating (computed using exact information) of the solutions found. One can see that the approximate algorithm indeed

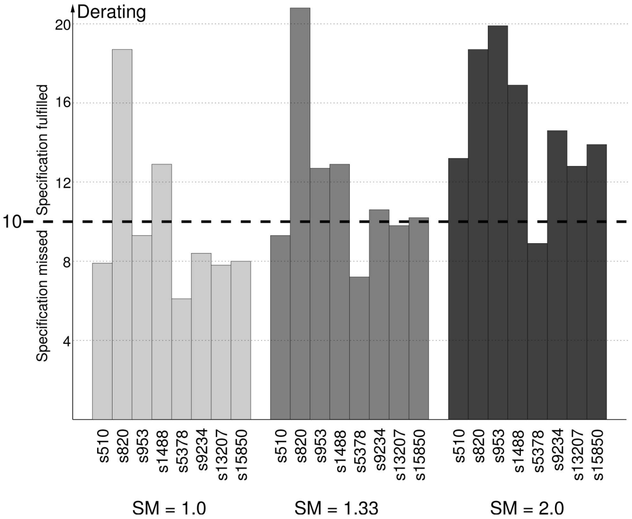


Fig. 6. Exact derating achieved by the approximate method with derating target 10 and safety margins 1, 1.33, and 2.

tends to overestimate the derating, resulting in an early termination of node selection, and thus, a solution of insufficient quality. Employing a safety margin of two, the target is met for almost all cases (the method is heuristic and cannot be expected to perform as well as the exact algorithm). It is advisable to use a safety margin of this order of magnitude for large circuits for which the BDDs cannot be generated, and thus, no exact derating figures are available.

To evaluate the scalability of the approximate method, we applied it to the combinational cores of some large industrial circuits provided by NXP. The results for node selection with derating target $D^{target} = 10$ and $D^{target} = 20$ are reported in Table 4. Column % contains the percentage of the selected soft errors (given in the column H) among all soft errors. The exact method could not be run for these circuits because of the memory requirements of the BDDs. The total number of soft errors again equals twice the number of gates in the circuit, roughly indicated by the circuit names. The runtimes are in CPU seconds and do not include the time to load the circuit (which took around 15 s for the largest circuit p2927k with approximately 2.5 million gates).

The results suggest that high derating values can be obtained for large circuits with a reasonable overhead and that the method is scalable.

TABLE 4
Selective Hardening of Industrial Circuits with Derating Target

Circuit	Soft errors	$Th_D = 10$			$Th_D = 20$		
		H	%	Time	H	%	Time
p35k	93456	6965	7.45	0.05	16605	17.77	0.05
p45k	87442	10713	12.25	0.05	20301	23.22	0.06
p77k	143266	7706	5.38	0.10	17394	12.14	0.09
p78k	154782	8218	5.31	0.10	28270	18.26	0.10
p81k	185674	22545	12.14	0.13	51191	27.57	0.13
p100k	193338	27683	14.32	0.12	53254	27.54	0.13
p267k	559650	116334	20.79	0.41	188020	33.6	0.41
p330k	696048	54346	7.81	0.53	132966	19.1	0.54
p378k	773894	41082	5.31	0.56	141335	18.26	0.57
p2927k	4887944	502711	10.28	4.20	1105949	22.63	4.31

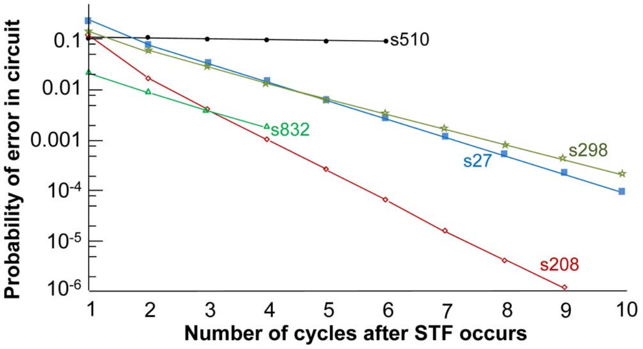


Fig. 7. $p_{err}(k)$ for ISCAS-89 circuits in graph form (logscale).

8.3 Sequential Circuits

Fig. 7 shows the values of $p_{err}(k)$ for the sequential ISCAS-89 circuits and various values of k in graph form. Note that $p_{err}(k)$ is the probability that the error still affects the circuit state after k cycles and that it decreases with k. It can be seen that for $k > 1$, the graphs closely approximate straight lines. This means that the ratio $p_{err}(k+1)/p_{err}(k)$ is nearly constant for a given circuit, although it does vary significantly from circuit to circuit as demonstrated by the slope of the curves.

The data from Fig. 7 suggest that knowing $p_{err}(k)$ for $k = 1$, 2, and 3 is sufficient for an accurate analysis in most cases, and it is possible to approximate $p_{err}(k)$ for larger values of k as follows:

$$p_{err}(k) \approx p_{err}(2) \cdot (p_{err}(3)/p_{err}(2))^{k-2} \quad (k > 2). \tag{21}$$

It is also possible to construct a Markov model similar to that in Fig. 4 based on the probabilities $p_{err}(1)$, $p_{err}(2)$, and $p_{err}(3)$. Note that this approximation is orthogonal to the detection-probability-based approximated node selection from Section 7.4.

TET and selective hardening are evaluated in Table 5 for the ISCAS-89 benchmark circuit s298. The table shows the number of gates to be hardened and their percentage to meet four alternative derating targets. It can be seen that selective hardening is indeed a low-cost way to reach a given derating threshold if nonreference behavior is acceptable for a few clock cycles. For example, improvement by three orders of magnitude is possible by hardening just seven percent of nodes if the noncritical error period is set to seven.

TABLE 5
Selective Hardening of Circuit s298

k	Th_D = 10		Th_D = 100		Th_D = 1000		Th_D = 10000	
	Cost	%	Cost	%	Cost	%	Cost	%
1	50	8.39	351	58.89	508	85.23	553	92.79
2	0	0	186	31.21	403	67.62	492	82.55
3	0	0	99	16.61	293	49.16	457	76.68
4	0	0	23	3.86	188	31.54	368	61.74
5	0	0	0	0	146	24.50	252	42.28
6	0	0	0	0	102	17.11	208	34.90
7	0	0	0	0	42	7.05	182	30.54
8	0	0	0	0	0	0	155	26.01
9	0	0	0	0	0	0	116	19.46
10	0	0	0	0	0	0	67	11.24
11	0	0	0	0	0	0	5	0.84

9 Conclusions

We introduced a general framework to model transient errors taking their impact on circuit functionality into account, in particular, their ability to disturb the circuit state for specified periods of time. Using the STF model, we defined various practical and technology-independent metrics for the errors of interest. These metrics include the error probability p_{err} for an individual circuit node, and the probability of self-recovery p_{sr} for the entire circuit. We also introduced the critical soft error rate CSER and the derating factor D, which can serve as measures of soft error susceptibility/tolerance of a design during logic synthesis. All these metrics can be computed quite efficiently using conventional simulation and ATPG techniques and, in the case of very large circuits, can be approximated quickly. We demonstrated the successful application of the proposed methodology in the case of the ISCAS-89 benchmarks achieving significant CSER improvement at limited cost.

Acknowledgments

This work was supported in part by the Alexander Von Humboldt Foundation, the DFG Project RealTest under Grants BE 1176/15-2 and PO 1220/2-2, and the US Air Force Research Laboratory under Agreement No. FA8750-05-1-0282. We are thankful to J. Schlöffel of NXP for providing the industrial benchmarks. Parts of this paper are based on "An Analysis Framework for Transient-Error Tolerance" which was presented at the VLSI Test Symposium, May 2007 [11] and on "Scalable Calculation of Logical Masking Effects for Selective Hardening against Soft Errors" which was presented at the International Symposium on VLSI, April 2008 [30].

References

[1] M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design.* Computer Science Press, 1990.

[2] S. Almukhaizim, T. Verdel, and Y. Makris, "Cost-Effective Graceful Degradation in Speculative Processor Subsystems: The Branch Prediction Case," *Proc. IEEE Int'l Conf. Computer Design,* pp. 194-197, 2003.

[3] H. Ando, R. Kan, Y. Tosaka, K. Takahisa, and K. Hatanaka, "Validation of Hardware Error Recovery Mechanisms for the SPARC64 V Microprocessor," *Proc. Int'l Conf. Dependable Systems and Networks,* pp. 62-69, 2008.

[4] H. Asadi and M. Tahoori, "Soft Error Modeling and Protection for Sequential Elements," *Proc. IEEE Defect and Fault Tolerance Symp.,* pp. 463-471, 2005.

[5] M. Breuer and H. Zhu, "An Illustrated Methodology for Analysis of Error Tolerance," *IEEE Design and Test of Computers,* vol. 25, no. 2, pp. 168-177, Mar./Apr. 2008.

[6] M.A. Breuer, "Testing for Intermittent Faults in Digital Circuits," *IEEE Trans. Computers,* vol. 22, no. 3, pp. 241-246, Mar. 1973.

[7] P.E. Dodd and L.W. Massengill, "Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics," *IEEE Trans. Nuclear Science,* vol. 50, no. 3, pp. 583-602, June 2003.

[8] K. Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg, "Byzantine Fault Tolerance, from Theory to Reality," *Proc. Int'l Conf. Computer Safety, Reliability and Security,* pp. 235-248, 2003.

[9] E.N. Elnozahy, L. Alvisi, Y.M. Wang, and D.B. Johnson, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," *ACM Computing Surveys,* vol. 34, no. 3, pp. 375-408, 2002.

[10] R. Garg, N. Jayakumar, S.P. Khatri, and G. Choi, "A Design Approach for Radiation-Hard Digital Electronics," *Proc. IEEE Design Automation Conf.,* pp. 773-778, 2006.

[11] J.P. Hayes, I. Polian, and B. Becker, "An Analysis Framework for Transient-Error Tolerance," *Proc. Very Large-Scale Integration Test Symp.,* pp. 249-255, 2007.

[12] S. Hellebrand, C.G. Zoellin, H.-J. Wunderlich, S. Ludwig, T. Coym, and B. Straube, "A Refined Electrical Model for Particle Strikes and Its Impact on SEU Prediction," *Proc. IEEE Defect and Fault Tolerance Symp.,* 2007.

[13] E. Hill, M. Lipasti, and K. Saluja, "An Accurate Flip-Flop Selection Technique for Reducing Logic SER," *Proc. Int'l Conf. Dependable Systems and Networks,* pp. 32-41, 2008.

[14] S.K. Jain and V.D. Agrawal, "Statistical Fault Analysis," *IEEE Design and Test of Computers,* vol. 2, no. 1, pp. 38-44, Jan./Feb. 1985.

[15] Z. Jiang and S. Gupta, "Threshold Testing: Improving Yield for Nanoscale VLSI," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 28, no. 12, pp. 1993-1895, Dec. 2009.

[16] V. Joshi, R.R. Rao, D. Blaauw, and D. Sylvester, "Logic SER Reduction through Flip Flop Redesign," *Proc. Int'l Symp. Quality Electronic Design,* pp. 611-616, 2006.

[17] S. Krishnaswamy, G.F. Viamontes, I.L. Markov, and J.P. Hayes, "Probabilistic Transfer Matrices in Symbolic Reliability Analysis of Logic Circuits," *ACM Trans. Design Automation of Electronic Systems,* vol. 13, no. 1, 2008.

[18] W.Y. Kung, C.S. Kim, and C.C.J. Kuo, "Spatial and Temporal Error Concealment Techniques for Video Transmission over Noisy Channels," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 16, no. 7, pp. 789-802, July 2006.

[19] X. Li and D. Yeung, "Application-Level Correctness and Its Impact on Fault Tolerance," *Proc. Int'l Symp. High Performance Computer Architecture,* pp. 181-192, 2007.

[20] J.W.S. Liu, W.K. Shin, K.J. Lin, R. Bettati, and J.Y. Chung, "Imprecise Computations," *Proc. IEEE,* vol. 82, no. 1, pp. 83-94, Jan. 1994.

[21] M. May, M. Alles, and N. Wehn, "A Case Study in Reliability-Aware Design: A Resilient LDPC Code Decoder," *Proc. Conf. Design, Automation and Test in Europe,* 2008.

[22] K. Mohanram and N.A. Touba, "Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits," *Proc. IEEE Int'l Test Conf.,* pp. 893-901, 2003.

[23] H.T. Nguyen and Y. Yagil, "A Systematic Approach to SER Estimation and Solutions," *Proc. Int'l Reliability Physics Symp.,* pp. 60-70, 2003.

[24] M. Nicolaidis, "GRAAL: A Fault-Tolerant Architecture for Enabling Nanometric Technologies," *Proc. Int'l On-Line Test Symp.,* p. 255, 2007.

[25] A.K. Nieuwland, S. Jasarevic, and G. Jerin, "Combinational Logic Soft Error Analysis and Protection," *Proc. Int'l On-Line Test Symp.,* 2006.

[26] D. Nowroth, I. Polian, and B. Becker, "A Study of Cognitive Resilience in a JPEG Compressor," *Proc. Int'l Conf. Dependable Systems and Networks,* pp. 32-41, 2008.

[27] Z. Pan and M.A. Breuer, "Basing Acceptable Error-Tolerant Performance on Significance-Based Error-Rate (SBER)," *Proc. Very Large-Scale Integration. Test Symp.,* 2008.

[28] I. Polian, B. Becker, M. Nakasato, S. Ohtake, and H. Fujiwara, "Low-Cost Hardening of Image Processing Applications against Soft Errors," *Proc. Int'l Symp. Defect and Fault Tolerance,* pp. 274-279, 2006.

[29] I. Polian, J.P. Hayes, S. Kundu, and B. Becker, "Transient Fault Characterization in Dynamic Noisy Environments," *Proc. IEEE Int'l Test Conf.,* pp. 1039-1048, 2005.

[30] I. Polian, S.M. Reddy, and B. Becker, "Scalable Calculation of Logical Masking Effects for Selective Hardening against Soft Errors," *Proc. IEEE Int'l Symp. Very Large-Scale Integration,* pp. 257-262, 2008.

[31] C. Rusu, A. Bougerol, L. Anghel, C. Weulerse, N. Buard, S. Benhammadi, N. Renaud, G. Hubert, F. Wrobel, T. Carriere, and R. Gaillard, "Multiple Event Transient Induced by Nuclear Reactions in CMOS Logic Cells," *Proc. Int'l On-Line Test Symp.,* pp. 137-145, 2007.

[32] J. Savir, "Testing for Single Intermittent Failures in Combinational Circuits by Maximizing the Probability of Fault Detection," *IEEE Trans. Computers,* vol. 29, no. 5, pp. 410-416, May 1980.

[33] S.A. Seshia, W. Li, and S. Mitra, "Verification-Guided Soft Error Resilience," *Proc. Conf. Design, Automation and Test in Europe,* 2007.

[34] S. Shahidi and S.K. Gupta, "ERTG: A Test Generator for Error-Rate Testing," *Proc. IEEE Int'l Test Conf.,* 2007.

[35] P. Shivakumar, M. Kistler, W. Keckler, D. Burger, and L. Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," *Proc. Int'l Conf. Dependable Systems and Networks,* pp. 389-398, 2002.

[36] F. Wang and V. Agrawal, "Soft Error Rate Determination for Nanometer CMOS VLSI logic," *Proc. Southeastern Symp. System Theory,* pp. 324-328, 2008.

[37] H.J. Wunderlich, "PROTEST: A Tool for Probabilistic Testability Analysis," *Proc. IEEE Design Automation Conf.,* 1985.

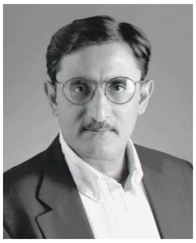
[38] M. Zhang, S. Mitra, T.M. Mak, N. Seifert, N.J. Wang, Q. Shi, K.S. Kim, N.R. Shanbhag, and S.J. Patel, "Sequential Element Design with Built-In Soft Error Resilience," *IEEE Trans. Very Large-Scale Integration Systems,* vol. 14, no. 12, pp. 1368-1378, Dec. 2006.

[39] M. Zhang and N.R. Shanbhag, "Soft Error-Rate Analysis (SERA) Methodology," *IEEE Trans. Computer-Aided Design,* vol. 25, no. 10, pp. 2140-2155, Oct. 2006.

[40] C.G. Zoellin, H.-J. Wunderlich, I. Polian, and B. Becker, "Selective Hardening in Early Design Steps," *Proc. European Test Symp.,* pp. 185-190, 2008.

Ilia Polian received the diploma (master's) and PhD degrees in computer science from the University of Freiburg, Germany, in 1999 and 2003, respectively. He was with Micronas, Freiburg, IBM R&D, Böblingen, Germany, and the Nara Institute of Science and Technology (NAIST), Japan. He served as a senior member of the scientific staff at the Chair of Computer Architecture at the University of Freiburg and recently joined the University of Passau, Germany, as a full professor of computer engineering. His research interests include defect modeling, design for testability, and formal verification of hybrid and real-time systems. He was a European Champion and Vice World Champion at the 1999 ACM International Collegiate Programming Contest, VDE Award Laureate 1999, and Wolfgang-Gentner Award Laureate 2004. He served as the finance chair of the IEEE European Test Symposium (ETS) 2007, the vice-program chair of the Reliability-Aware System Design and Test Workshop (RASDAT) 2010 and 2011, the General Chair of the GI/ITG/GMM Test Methods and Reliability Workshop 2011, and was on a number of Program Committees. He is a senior member of the IEEE.

John P. Hayes received the BE degree from the University College Dublin, and the MS and PhD degrees from the University of Illinois, Urbana-Champaign, all in electrical engineering. While at the University of Illinois, he participated in the design of the ILLIAC III computer. He is currently a professor in the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, where he holds the Claude E. Shannon endowed chair in engineering science. He was previously a faculty member of the Departments of Electrical Engineering Systems and Computer Science at the University of Southern California, Los Angeles. He spent two years with the Operations Research Group at the Shell Benelux Computing Center in The Hague, where he was involved in mathematical programming. He has also held visiting positions at several organizations including Stanford University, LogicVision Inc., and the University of Freiburg. His current teaching and research interests include computer-aided design, verification, and testing; VLSI circuits; computer architecture, especially embedded computer networks; fault-tolerant and safety-critical systems; and quantum computing. He was the founding director of the University of Michigan's Advanced Computer Architecture Laboratory (ACAL). He has authored more than 250 technical papers, several patents, and six books, including *Computer Architecture and Organization* (third ed., McGraw-Hill, 1998) and *Quantum Circuit Simulation* (Springer, 2009). He has served as an editor of the *Communications of the ACM*, the *IEEE Transactions on Parallel and Distributed Systems*, and the *Journal of Electronic Testing*. He is a fellow of the IEEE and the ACM, and a member of the Sigma Xi. He received the University of Michigan's Distinguished Faculty Achievement Award in 1999, and the Alexander von Humboldt Foundation's Research Award in 2004.



Sudhakar M. Reddy received the BE degree in electronics and communication engineering from Osmania University, Hyderabad, India, the ME degree from the Indian Institute of Science, Bengaluru, India, and the PhD degree in electrical engineering from the University of Iowa. Since 1968, he has been a faculty member in the Department of Electrical and Computer Engineering, University of Iowa, where he is currently a University of Iowa Foundation distinguished professor. He served as the chair of the department from 1981 to 2000. He has published more than 500 papers in the areas of test and design for test of digital VLSI circuits, coding theory, and fault-tolerant computing. He is a life fellow of the IEEE. He received a Von Humboldt Senior Research Fellowship in 1995, and a Life Time Achievement Award from VLSI Design Conference. He has served twice as a guest editor and as an associate editor of the *IEEE Transactions on Computers*, and as an associate editor of the *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. He was the technical program chair of the 1989 International Fault-Tolerant Computing Symposium.

Bernd Becker is a full professor in the Faculty of Engineering, University of Freiburg, Germany. Prior to joining University of Freiburg in 1995, he was with J. W. Goethe-University Frankfurt as an associate professor for complexity theory and efficient algorithms. His research interests include computer-aided design, test, and verification of (digital) circuits and systems (VLSI CAD). A focus of his research is the development and analysis of efficient data structures and algorithms in VLSI CAD. The development of symbolic methods for test and verification of digital circuits and their integration in the industrial flow is one of the major achievements of his work. More recently, he has been working on verification methods for embedded systems and test techniques for nanoelectronic circuitry. He has published more than 200 papers in peer-reviewed conferences and journals and has been on the program and organizing committees of numerous major international conferences. He has been the holder of several research grants from DFG, BMBF, and industry as well. He currently serves as the co-speaker of the DFG transregional collaborative research center "Automatic Analysis and Verification of Complex Systems (AVACS)" with project partners from the University of Freiburg, the University of Saarland, the University of Oldenburg, and the Max Planck Institute of Computer Science. He is a fellow of the IEEE.